1. Integration Documentation . . . 2
1.1 Integration Points . . . 4
1.1.1 Application Links (AppLinks) . . . 4
1.1.2 Bamboo Builds in Eclipse . . . 6
1.1.3 Bamboo Builds in IDEA . . . 7
1.1.4 Crucible Reviews in IDEA . . . 9
1.1.5 JIRA Bamboo Integration . . . 11
1.1.6 JIRA Issues in IDEA . . . 13
1.1.7 JIRA Issues on a Confluence Page . . . 14
1.1.8 JIRA Portlets on a Confluence Page . . . 16
1.1.9 JIRA Studio Bamboo Integration . . . 16
1.1.10 Plugin Development . . . 17
1.1.11 Single Sign-On (SSO) across Web Applications . . . 19
1.1.12 User and Group Management across Directories and Web Applications . . . 21
1.2 Links and Resources . . . 22
1.2.1 Atlassian Application Configuration - Documentation Links . . . 23
1.3 Guide to Installing an Atlassian Integrated Suite . . . 23
1.3.1 Here Be Dragons . . . 23
1.3.1.1 Dragons Stage 1 - Install Java, PostgreSQL and Crowd . . . 25
1.3.1.2 Dragons Stage 2 - Install JIRA . . . 30
1.3.1.3 Dragons Stage 3 - Install GreenHopper into JIRA . . . 37
1.3.1.4 Dragons Stage 4 - Install Confluence . . . 40
1.3.1.5 Dragons Stage 5 - Install FishEye . . . 47
1.3.1.6 Dragons Stage 6 - Get JIRA and FishEye Talking . . . 53
1.3.1.7 Dragons Stage 7 - Install Bamboo . . . 56
1.3.1.8 Dragons Stage 8 - Bamboo Gadgets and JIRA Victory . . . 65
1.3.2 Dragon Slayers with JIRA Already Installed . . . 67
1.3.2.1 Dragons with JIRA Stage 1 - Install Java, PostgreSQL and Crowd . . . 68
1.3.2.2 Dragons with JIRA Stage 2 - Set Up JIRA . . . 72

Integration Documentation

Watch this space

This documentation is under development. We will keep adding information about how your Atlassian applications work together.

Links and Resources

Want to know more? Take a look at our Links and Resources, or follow the links in the table of contents on the left.

Atlassian Integration Points

Here are some of the ways your Atlassian applications can work together. Click an item to see more details of the integration point and how you can make it happen.

Application Links (AppLinks)

Use Application Links (AppLinks) to link your JIRA, Confluence, FishEye, Crucible and Subversion applications. You can even choose to associate individual projects, spaces, reviews and repositories with each other. You can then use shortcut links within each application. The application will 'know' automatically which space, project or repository the link is referring to.

Bamboo Builds in Eclipse

The Atlassian Connector for Eclipse displays Bamboo build information in your Eclipse development environment.

Bamboo Builds in IDEA

The Atlassian Connector for IntelliJ IDEA displays Bamboo build information in your IntelliJ IDEA development environment.

Crucible Reviews in IDEA

The Atlassian Connector for IntelliJ IDEA lets you work with your Crucible code reviews within your IntelliJ IDEA development environment.

JIRA Bamboo Integration

Integrate JIRA with Bamboo to link JIRA issues to Bamboo builds. Once you have linked JIRA issues to your builds, you can view the linked issues in your Bamboo build results and build plans. You can also view build information in JIRA via tabs on related issues, versions and projects or on portlets.

JIRA Issues in IDEA

The Atlassian Connector for IntelliJ IDEA integrates JIRA issues into your IntelliJ IDEA development environment.

JIRA Issues on a Confluence Page

Use the JIRA Issues macro to display a list of JIRA issues on a Confluence page. The list is shown in tabular format, and is dynamically updated from your JIRA site. People viewing the Confluence page will see only the JIRA issues which they are authorised to see.

JIRA Portlets on a Confluence Page

Use the JIRA Portlet macro to display a portlet from your JIRA site on a Confluence page. The information is dynamically updated from your JIRA site. People viewing the Confluence page will see only the JIRA issues which they are authorised to see.

JIRA Studio Bamboo Integration

Integrate your hosted Bamboo instance with your JIRA Studio Subversion repository to enable all of the continuous integration benefits of Bamboo for your development suite.

Plugin Development

Atlassian is moving towards wider and more intimate integration between the different applications we develop. Providing a common plugin framework is a key part of this initiative, because the plugin framework will help developers (both Atlassians and others) to write plugins that work on more than one of the products. It will also allow plugins on each product to have a consistent look and behaviour from a user's point of view, even when the plugin works on only one application.

Single Sign-On (SSO) across Web Applications

Using Crowd for single sign-on (SSO), each person needs only one username and password to access all web applications. You only need to log in once, to Crowd or a Crowd-connected application. When you start another Crowd-connected application, you will be logged in automatically. Similarly, when you log out, you will be logged out of Crowd and the other applications at the same time.

User and Group Management across Directories and Web Applications

Use Crowd to manage users and groups simply and centrally. You can use Crowd's internal user directories or hook up your existing LDAP user bases, then link them to your web applications. Crowd provides an Administration Console for administrators as well as a Self-Service Console, allowing users to view their login permissions and to change their own passwords and user profiles.

Integration Points

- Application Links (AppLinks)
- Bamboo Builds in Eclipse
- Bamboo Builds in IDEA
- Crucible Reviews in IDEA
- JIRA Bamboo Integration
- JIRA Issues in IDEA
- JIRA Issues on a Confluence Page
- JIRA Portlets on a Confluence Page
- JIRA Studio Bamboo Integration
- Plugin Development
- Single Sign-On (SSO) across Web Applications
- User and Group Management across Directories and Web Applications

Application Links (AppLinks)

This page summarises the Application Links (AppLinks) plugin and provides links to more information. Unless otherwise stated, the information here refers to the latest version of the plugin. Please follow the links to find out about previous versions.

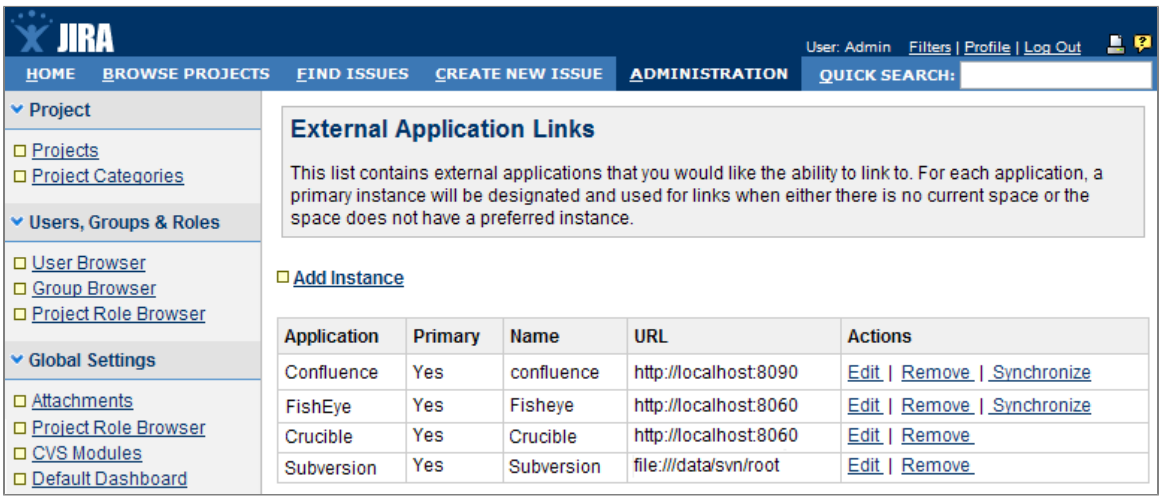
On this page:

- Short Description
- Full Description
- Components
- Version Matrix
- Documentation

Short Description

Use Application Links (AppLinks) to link your JIRA, Confluence, FishEye, Crucible and Subversion applications. You can even choose to associate individual projects, spaces, reviews and repositories with each other. You can then use shortcut links within each application. The application will 'know' automatically which space, project or repository the link is referring to.

Screenshot: Linking Applications to JIRA



Full Description

What is Application Links?

Application Links (AppLinks) is a plugin that allows you to link your JIRA, Confluence, FishEye, Crucible and Subversion applications. You can even choose to associate individual projects, spaces, reviews and repositories with each other. Then, when you add text into one of the applications, you can refer to a page, issue or source file using just a simple textual reference. You do not need to enter the full URL. The text is automatically linked to the correct project, space, review or repository.

An Example

- Let's say that your JIRA installation has a project called myProject.
- Let's also say that your Confluence installation has a space called mySpace, which contains all the documentation for myProject on JIRA.
- Using AppLinks, you have linked myProject to mySpace.
- Now, you can add the following markup to a JIRA issue in myProject: [Test Page].
- JIRA will automatically display the text with a link to the Confluence page called 'Test Page' in mySpace.

Components

Type of Integration

Application Links is a plugin, which you will install and configure in the relevant products.

List of Components

You will need the following products to make use of Application Links:

- Two or more of the following: Confluence, JIRA, FishEye, Crucible and/or Subversion.
- The AppLinks plugin.

Version Matrix

The matrix below shows the applications that support AppLinks. The applications are listed horizontally across the top and the AppLinks versions are listed vertically on the left.

Version numbers in brackets, e.g. (3.0.1), show a **future** application release expected to support the relevant AppLinks version.

	Confluence	Crucible	FishEye	JIRA	Subversion
AppLinks 1.1	2.8.x	1.5.x	1.5.x	3.12.x	1.4
AppLinks 1.3	2.8.x 2.9.x 2.10.x Confluence 3.0.0 does not support Application Links.	1.5.x Crucible 1.6.x does not support Application Links.	1.5.x FishEye 1.6.x does not support Application Links.	3.12.x 3.13.0 JIRA 3.13.1 to 3.13.4 do not support Application Links.	1.4
AppLinks 2.0.1	3.0.1	2.0.0	2.0.0	3.13.5	1.4
AppLinks 2.1	3.0.1	2.0.0	2.0.0	4.0	1.4

Other useful links:

- AppLinks full version compatibility on the Atlassian Plugin Exchange
- Version matrix for the Atlassian Plugin Framework

Documentation

Latest Version

Please refer to the Application Links documentation:

- Application Links Installation Guide
- Application Links Administration Guide
- Application Links User Guide
- Application Links Development Hub
- Application Links Release Notes
- Application Links FAQ

All Versions

AppLinks 2.1 Documentation
AppLinks 2.0 Documentation
AppLinks 1.x Documentation

Bamboo Builds in Eclipse

This page introduces the Atlassian Connector for Eclipse and provides links to more information.

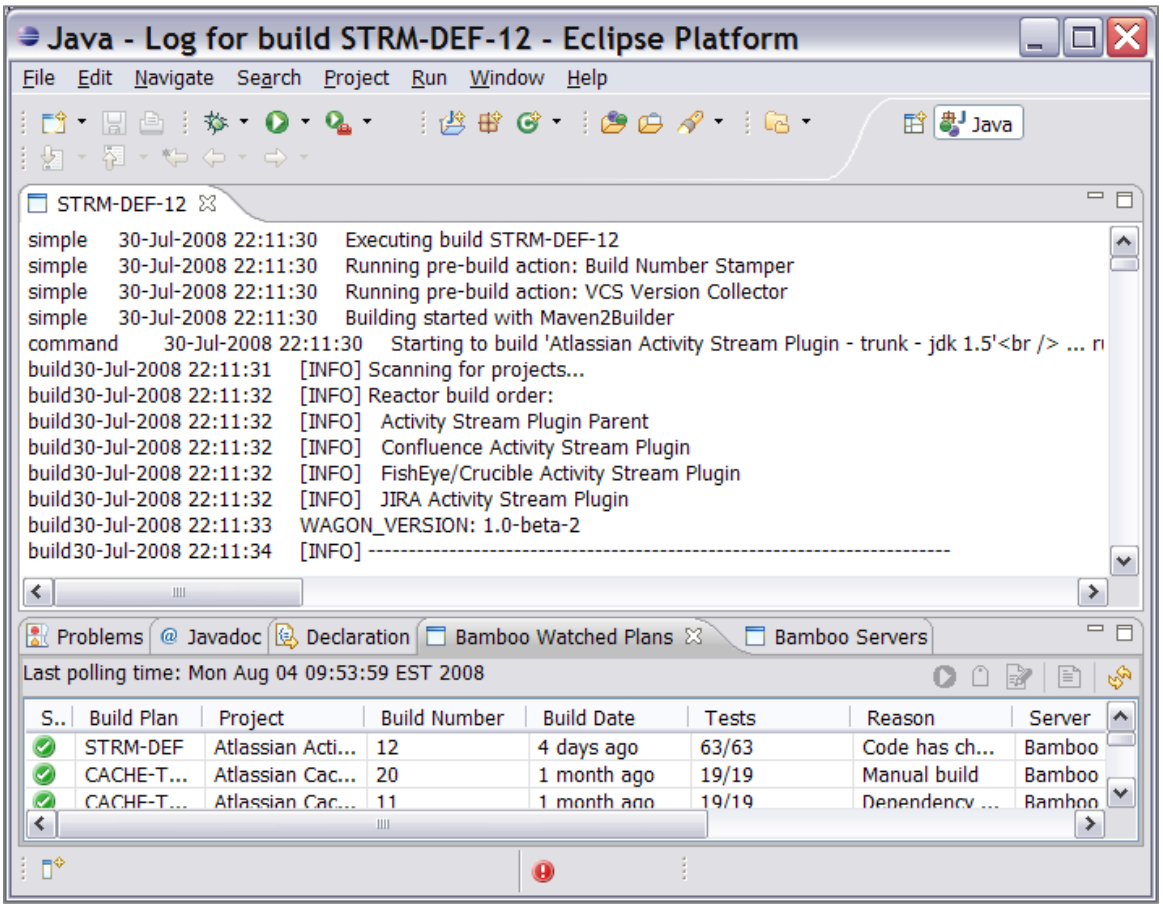
On this page:

- Short Description
- Full Description
- Components
- Version Matrix
- Documentation

Short Description

The Atlassian Connector for Eclipse displays Bamboo build information in your Eclipse development environment.

Screenshot: Bamboo Watched Plans and Build Log in Eclipse



Full Description

The Atlassian Connector for Eclipse is an Eclipse plugin. It allows you to work with the Atlassian products within your Eclipse IDE. Now you don't have to switch between websites, email messages and news feeds to see what's happening to your project and your code. Instead, you can see the relevant JIRA issues, Crucible reviews and Bamboo build information right there in your development environment.

Components

The Atlassian Connector for Eclipse is a plugin that you will install and configure in Eclipse.

Version Matrix

The Atlassian Connector for Eclipse is compatible with the following software versions:

	Atlassian Connector for Eclipse — version 1.1.x
Bamboo	1.x 2.x recommended
Crucible	1.6 *or later* We recommend that you upgrade to the latest version of Crucible 1.6.x or 2.x for best results. However, please note that the new features of Crucible 2.0 are not reflected in the connector, such as iterative reviews, read/unread support, etc. Existing Crucible 1.6.x functionality will work when you connect to a Crucible 2.0 server.
Eclipse	3.3 (Europa) 3.4 (Ganymede) 3.5 (Galileo)
FishEye	1.6 *or later* We recommend that you upgrade to the latest version of FishEye 1.6.x or 2.x for best results.
JIRA	3.4 *or later* 3.7 *or later* recommended

Documentation

Please refer to the documentation for the Atlassian Connector for Eclipse:

- Installation and Upgrade Guide for the Eclipse Connector
- Configuring the Eclipse Connector
- Using Bamboo in the Eclipse Connector
- Using Crucible in the Eclipse Connector
- Using FishEye in the Eclipse Connector
- Using JIRA in the Eclipse Connector
- Eclipse Connector FAQ

Bamboo Builds in IDEA

This page introduces the Atlassian Connector for IntelliJ IDEA and provides links to more information.

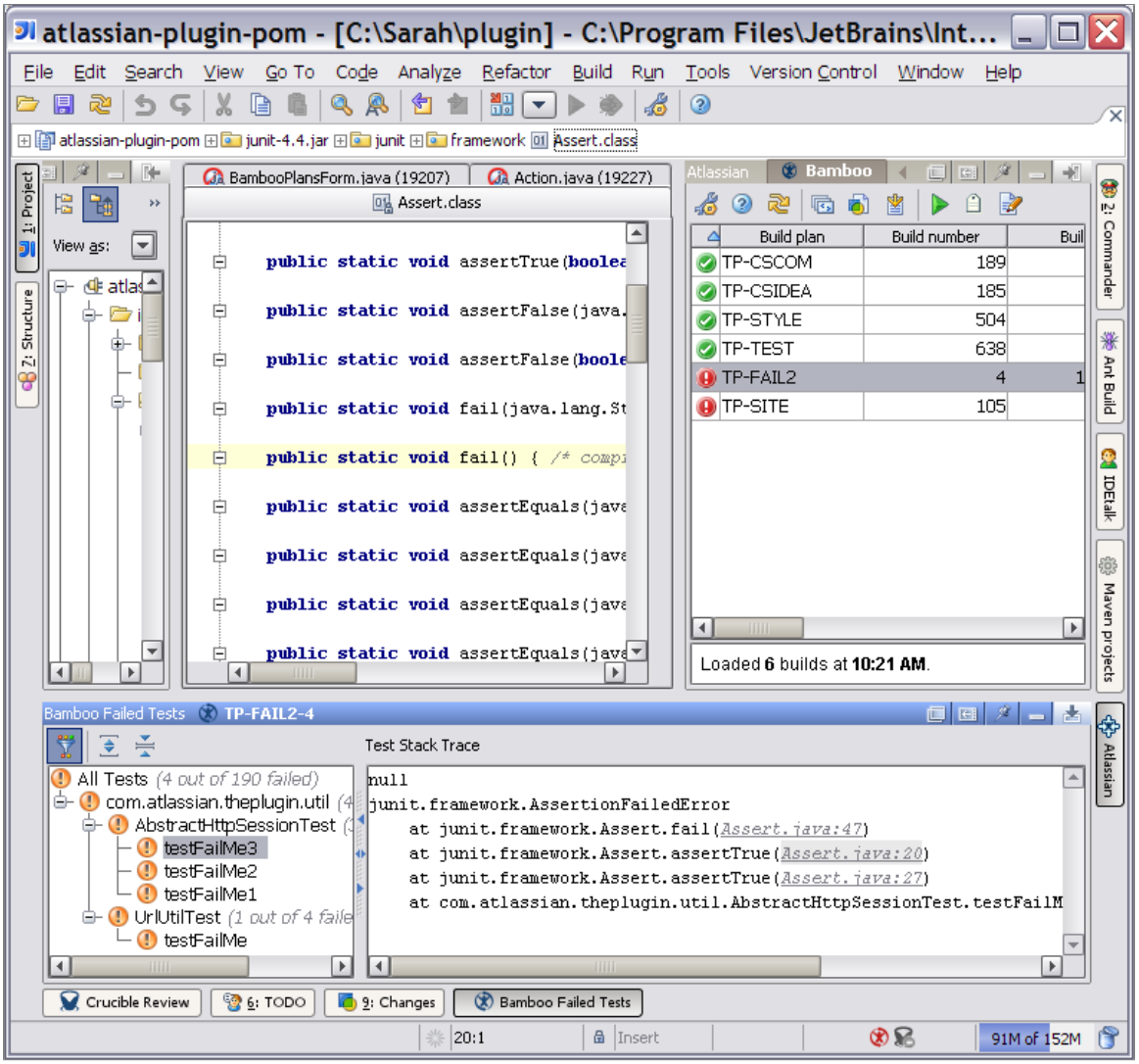
On this page:

- Short Description
- Full Description
- Components
- Version Matrix
- Documentation

Short Description

The Atlassian Connector for IntelliJ IDEA displays Bamboo build information in your IntelliJ IDEA development environment.

Screenshot: Bamboo Builds in an IDEA Tool Window



Full Description

The Atlassian Connector for IntelliJ IDEA is an IntelliJ IDEA plugin. It allows you to work with the Atlassian products within your IDE. Now you don't have to switch between websites, email messages and news feeds to see what's happening to your project and your code. Instead, you can see the relevant JIRA issues, Crucible reviews and Bamboo build information right there in your development environment.

Components

The Atlassian Connector for IntelliJ IDEA is a plugin, which you will install and configure in IntelliJ IDEA.

Version Matrix

The Atlassian Connector for IntelliJ IDEA works with the application versions listed below.

Bamboo	1.x 2.x recommended
Crucible	1.6 *or later* ⚠ Earlier versions of Crucible are not supported. We recommend that you upgrade to the latest version of Crucible 1.6.x or 2.x for best results. However, please note that the new features of Crucible 2.0 are not reflected in the connector, such as iterative reviews, read/unread support, etc. Existing Crucible 1.6.x functionality will work when you connect to a Crucible 2.0 server.
FishEye	1.6 *or later* ⚠ Earlier versions of FishEye are not supported.

JIRA	3.7 *or later* 3.12 *or later* recommended
IntelliJ IDEA	7.0.2 *or later, including IDEA 8.0*

Documentation

Please refer to the documentation for the Atlassian Connector for IntelliJ IDEA:

- Installation and Upgrade Guide for the IntelliJ Connector
- Configuring the IntelliJ Connector
- Keyboard Shortcuts in the IntelliJ Connector
- Using Bamboo in the IntelliJ Connector
- Using Crucible in the IntelliJ Connector
- Using FishEye in the IntelliJ Connector
- Using JIRA in the IntelliJ Connector
- IntelliJ Connector FAQ

Crucible Reviews in IDEA

This page introduces the Atlassian Connector for IntelliJ IDEA and provides links to more information.

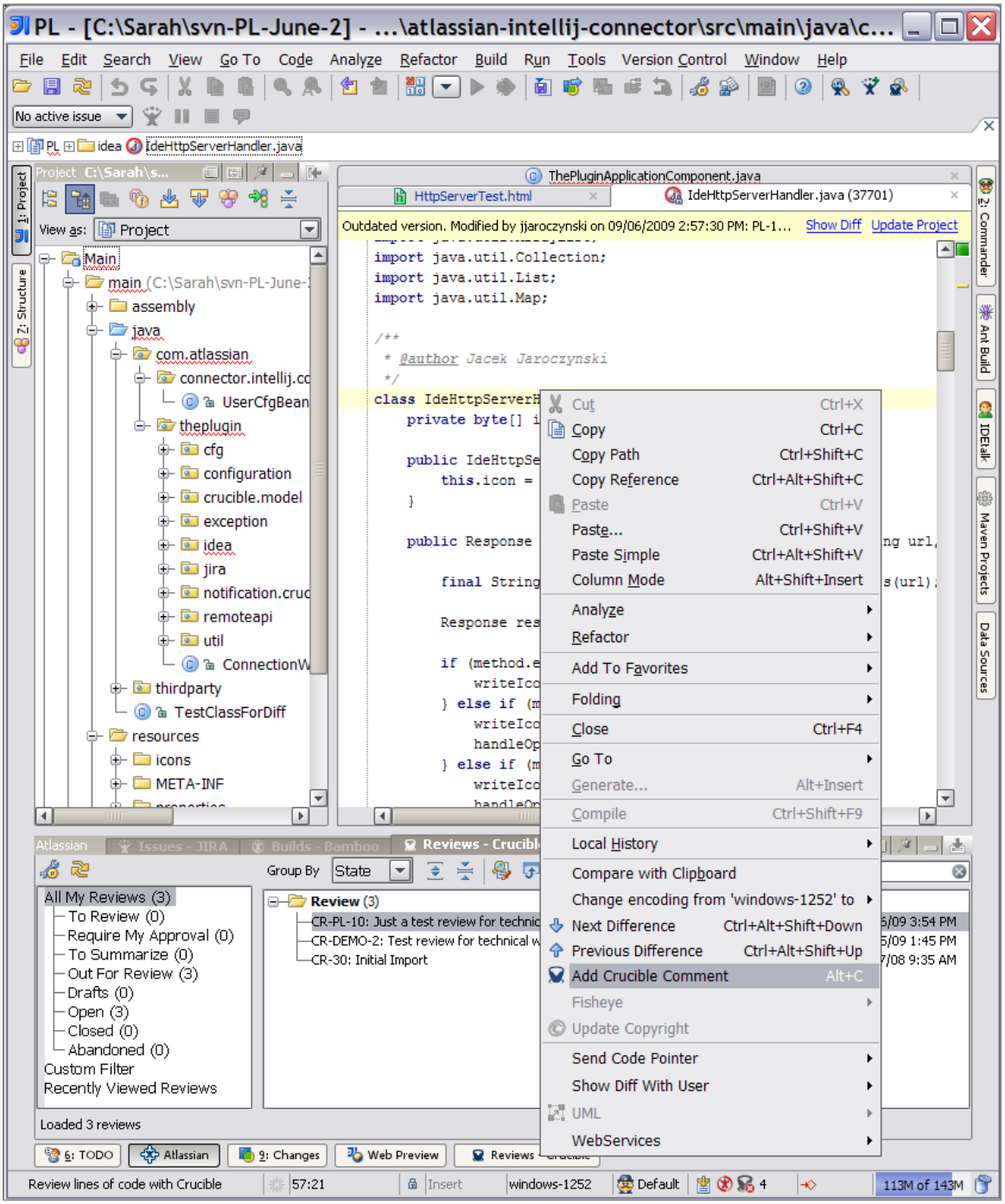
On this page:

- Short Description
- Full Description
- Components
- Version Matrix
- Documentation

Short Description

The Atlassian Connector for IntelliJ IDEA lets you work with your Crucible code reviews within your IntelliJ IDEA development environment.

Screenshot: Crucible Reviews in an IDEA Tool Window



Full Description

The Atlassian Connector for IntelliJ IDEA is an IntelliJ IDEA plugin. It allows you to work with the Atlassian products within your IDE. Now you don't have to switch between websites, email messages and news feeds to see what's happening to your project and your code. Instead, you can see the relevant JIRA issues, Crucible reviews and Bamboo build information right there in your development environment.

Components

The Atlassian Connector for IntelliJ IDEA is a plugin, which you will install and configure in IntelliJ IDEA.

Version Matrix

The Atlassian Connector for IntelliJ IDEA works with the application versions listed below.

Bamboo	1.x 2.x recommended
Crucible	1.6 *or later* ⚠ Earlier versions of Crucible are not supported. We recommend that you upgrade to the latest version of Crucible 1.6.x or 2.x for best results. However, please note that the new features of Crucible 2.0 are not reflected in the connector, such as iterative reviews, read/unread support, etc. Existing Crucible 1.6.x functionality will work when you connect to a Crucible 2.0 server.
FishEye	1.6 *or later* ⚠ Earlier versions of FishEye are not supported.
JIRA	3.7 *or later* 3.12 *or later* recommended
IntelliJ IDEA	7.0.2 *or later, including IDEA 8.0*

Documentation

Please refer to the documentation for the Atlassian Connector for IntelliJ IDEA:

- Installation and Upgrade Guide for the IntelliJ Connector
- Configuring the IntelliJ Connector
- Keyboard Shortcuts in the IntelliJ Connector
- Using Bamboo in the IntelliJ Connector
- Using Crucible in the IntelliJ Connector
- Using FishEye in the IntelliJ Connector
- Using JIRA in the IntelliJ Connector
- IntelliJ Connector FAQ

JIRA Bamboo Integration

This page gives an overview of JIRA Bamboo integration and provides links to more information. Unless otherwise stated, the information here refers to the latest version of JIRA, Bamboo and the JIRA Bamboo plugin. Please follow the links to find out about previous versions.

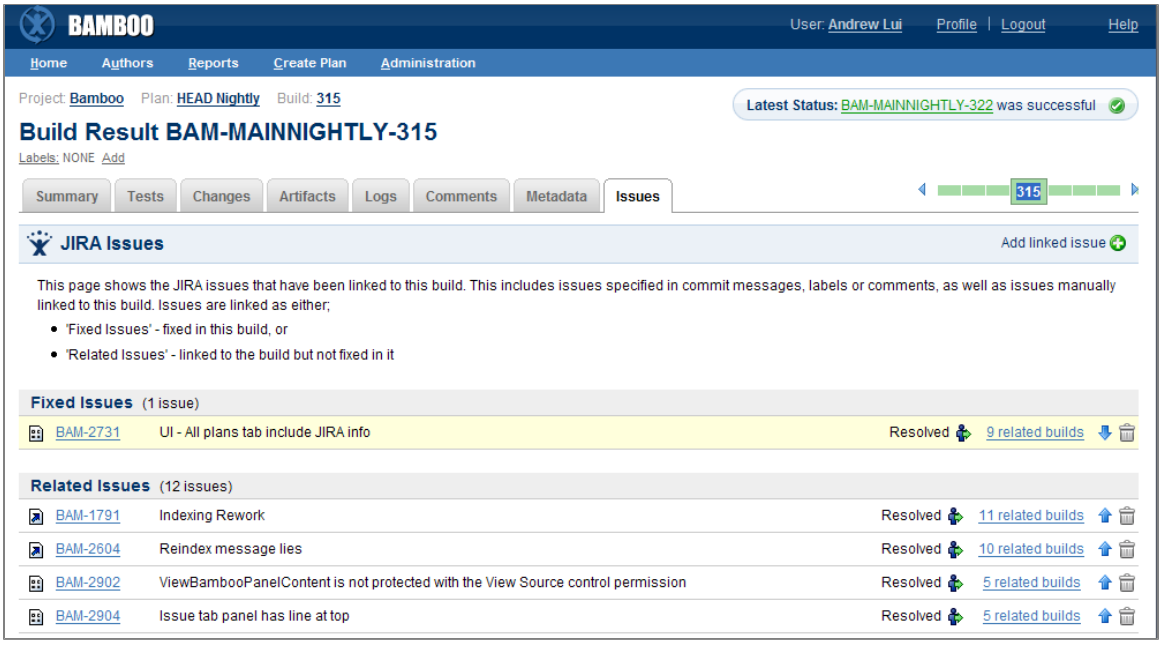
On this page:

- Short Description
- Full Description
- Components
- Version Matrix
- Documentation

Short Description

Integrate JIRA with Bamboo to link JIRA issues to Bamboo builds. Once you have linked JIRA issues to your builds, you can view the linked issues in your Bamboo build results and build plans. You can also view build information in JIRA via tabs on related issues, versions and projects or on portlets.

Screenshot: Viewing the JIRA Issues for a Build Result



Full Description

Integrating Bamboo with Atlassian's JIRA combines Bamboo's continuous integration capabilities with your issue tracker to give you a unified view of your software development project. Using JIRA and Bamboo together, you can see which issues are being actively coded, which builds have run for an issue, find the build that fixed the issue, download your distribution and much more.

Components

Type of Integration

JIRA Bamboo integration is enabled by two separate plugins, the **JIRA plugin for Bamboo** and the **JIRA Bamboo plugin** (i.e. Bamboo Plugin for JIRA). These plugins need to be installed and configured in both Bamboo and JIRA, as described in the JIRA Bamboo integration documentation.

List of Components

You will need the following products to integrate JIRA and Bamboo:

- JIRA
- Bamboo
- The JIRA Bamboo plugin — refer to the JIRA Bamboo integration documentation for details on how to set this up.

Please note, the JIRA Plugin for Bamboo is shipped with Bamboo.

Version Matrix

The following matrix provides version compatibility information for the JIRA Bamboo plugin. As the JIRA plugin for Bamboo is bundled with Bamboo, the following matrix should provide you with all the information you need regarding component versions:

Version	Summary	Product Version Compatibility	JAR
1.0.1	View Bamboo builds related to a JIRA issue	JIRA 3.7 or higher & Bamboo 1.2.4 or higher	http://downloads.atlassian.com/software/bamboo/jira-bamboo-plugin-1.0.1.jar

1.0.2	Portlet functionality added	JIRA 3.7 or higher & Bamboo 2.0.x or higher	http://repository.atlassian.com/jira-bamboo-plugin/jars/jira-bamboo-plugin-1.0.2.jar
2.1	Enhanced build information in JIRA and enhanced issue information in Bamboo	JIRA 3.12 or higher & Bamboo 2.1 or higher	https://maven.atlassian.com/public/com/atlassian/jira/plugins/jira-bamboo-plugin/2.1/jira-bamboo-plugin-

Source:

- **v1.x.x:** http://svn.atlassian.com/svn/public/contrib/jira/jira-bamboo-plugin-v1/
- **v2.x.x:** http://svn.atlassian.com/svn/public/contrib/jira/jira-bamboo-plugin-v2/

Documentation

Please refer to the JIRA Bamboo integration documentation, as well as the JIRA Bamboo Plugin documentation.

JIRA Issues in IDEA

This page introduces the Atlassian Connector for IntelliJ IDEA and provides links to more information.

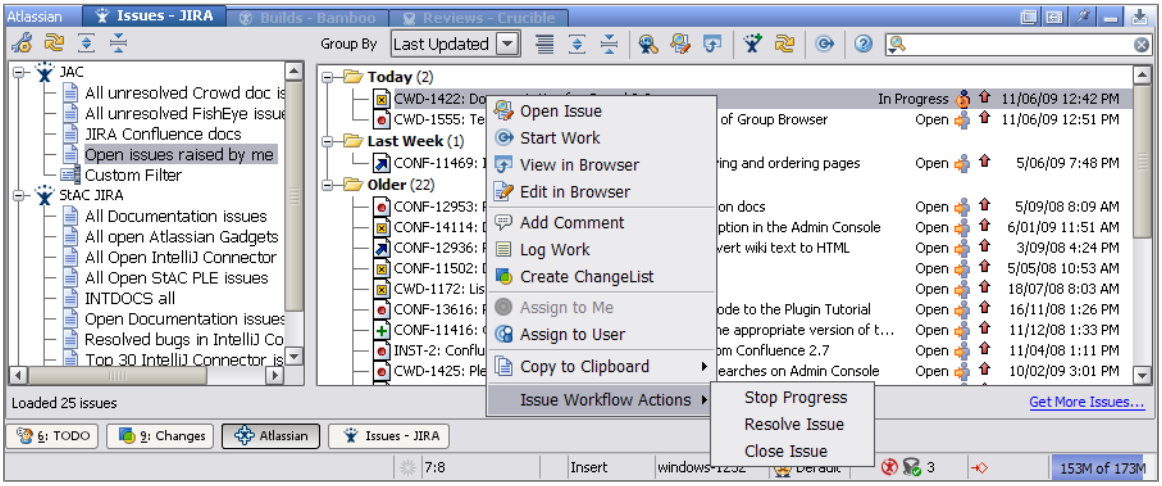
On this page:

- Short Description
- Full Description
- Components
- Version Matrix
- Documentation

Short Description

The Atlassian Connector for IntelliJ IDEA integrates JIRA issues into your IntelliJ IDEA development environment.

Screenshot: JIRA Issues in an IDEA Tool Window



Full Description

The Atlassian Connector for IntelliJ IDEA is an IntelliJ IDEA plugin. It allows you to work with the Atlassian products within your IDE. Now you don't have to switch between websites, email messages and news feeds to see what's happening to your project and your code. Instead, you can see the relevant JIRA issues, Crucible reviews and Bamboo build information right there in your development environment.

Components

The Atlassian Connector for IntelliJ IDEA is a plugin, which you will install and configure in IntelliJ IDEA.

Version Matrix

The Atlassian Connector for IntelliJ IDEA works with the application versions listed below.

Bamboo	1.x 2.x recommended
Crucible	1.6 *or later* Earlier versions of Crucible are not supported. We recommend that you upgrade to the latest version of Crucible 1.6.x or 2.x for best results. However, please note that the new features of Crucible 2.0 are not reflected in the connector, such as iterative reviews, read/unread support, etc. Existing Crucible 1.6.x functionality will work when you connect to a Crucible 2.0 server.
FishEye	1.6 *or later* Earlier versions of FishEye are not supported.
JIRA	3.7 *or later* 3.12 *or later* recommended
IntelliJ IDEA	7.0.2 *or later, including IDEA 8.0*

Documentation

Please refer to the documentation for the Atlassian Connector for IntelliJ IDEA:

- Installation and Upgrade Guide for the IntelliJ Connector
- Configuring the IntelliJ Connector
- Keyboard Shortcuts in the IntelliJ Connector
- Using Bamboo in the IntelliJ Connector
- Using Crucible in the IntelliJ Connector
- Using FishEye in the IntelliJ Connector
- Using JIRA in the IntelliJ Connector
- IntelliJ Connector FAQ

JIRA Issues on a Confluence Page

This page summarises the JIRA Issues macro and provides links to more information. Unless otherwise stated, the information here refers to the latest version of Confluence and the macro. Please follow the links to find out about previous versions.

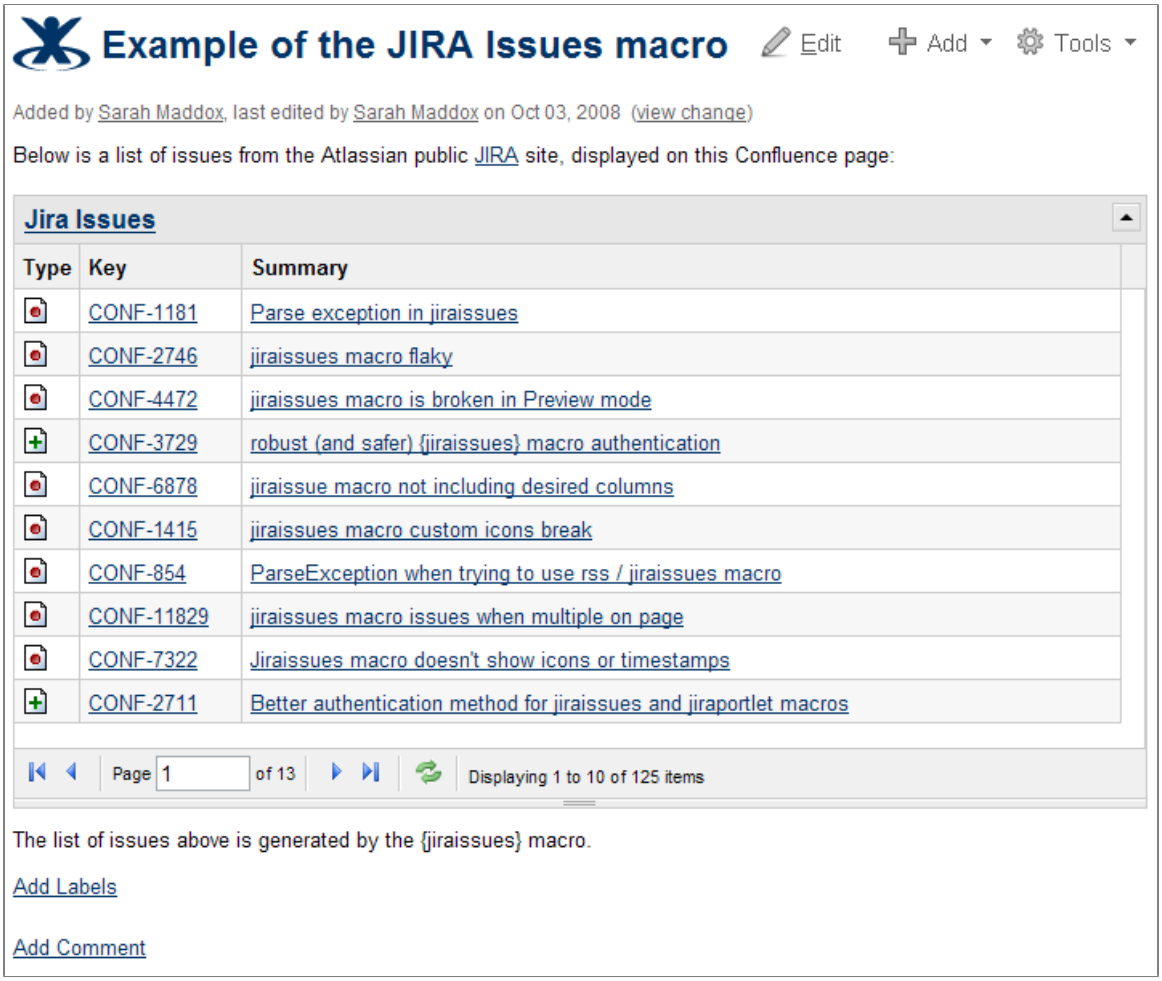
On this page:

- Short Description
- Full Description
- Components
- Documentation

Short Description

Use the JIRA Issues macro to display a list of JIRA issues on a Confluence page. The list is shown in tabular format, and is dynamically updated from your JIRA site. People viewing the Confluence page will see only the JIRA issues which they are authorised to see.

Screenshot: JIRA issues displayed on a Confluence page



Full Description

JIRA is the issue tracking and project management system supplied by Atlassian. The JIRA Issues macro allows you to display a list of issues from a JIRA site within a page in Confluence.

In other words, if you have your own JIRA site, your Confluence page can show a dynamically-updated list of issues from your JIRA project. You can also show a list of issues from any JIRA site to which you and your readers have access.

Components

Type of Integration

The JIRA Issues macro is provided by a plugin, which is shipped as part of the Confluence installation. Provided that the Confluence administrator has not disabled the macro, you will be able to use it without any additional installation.

Documentation

Latest Version

Please refer to the Confluence JIRA Issues macro.

All Versions

Confluence 3.0
Confluence 2.10.x
Confluence 2.9.x
Confluence 2.8.x
Confluence 2.7.x
Confluence 2.6.x
Confluence 2.5.4 to 2.5.8
Confluence 2.0 to 2.5.3

JIRA Portlets on a Confluence Page

This page summarises the JIRA Portlet macro and provides links to more information. Unless otherwise stated, the information here refers to the latest version of Confluence and the macro. Please follow the links to find out about previous versions.

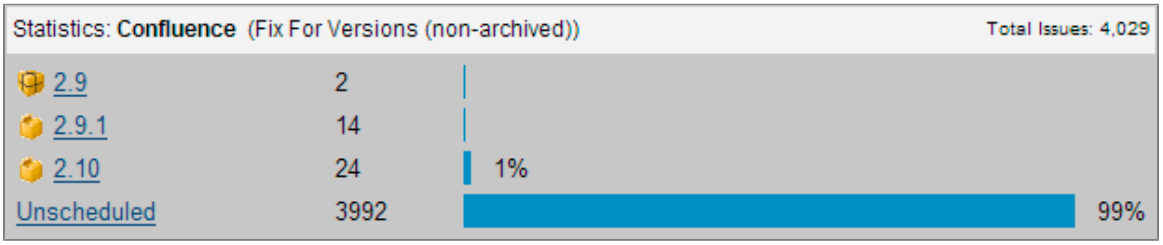
On this page:

- Short Description
- Full Description
- Components
- Documentation

Short Description

Use the JIRA Portlet macro to display a portlet from your JIRA site on a Confluence page. The information is dynamically updated from your JIRA site. People viewing the Confluence page will see only the JIRA issues which they are authorised to see.

Screenshot: JIRA Portlet displayed on a Confluence page



Full Description

JIRA is the issue tracking and project management system supplied by Atlassian. The JIRA Portlet macro allows you to display a JIRA dashboard portlet on a Confluence page.

Components

Type of Integration

The JIRA Portlet macro is provided by a plugin, which is shipped as part of the Confluence installation. Provided that the Confluence administrator has not disabled the macro, you will be able to use it without any additional installation.

Documentation

Latest Version

Please refer to the Confluence JIRA Portlet macro.

All Versions

Confluence 3.0
Confluence 2.10.x
Confluence 2.9.x
Confluence 2.8.x
Confluence 2.7.x
Confluence 2.6.x
Confluence 2.5.4 to 2.5.8
Confluence 2.0 to 2.5.3

JIRA Studio Bamboo Integration

This page gives an overview of JIRA Studio integration with Bamboo and provides links to more information. Unless otherwise stated, the information here refers to the latest version of JIRA Studio integration with Bamboo. Please follow the links to find out about previous versions.

On this page:

- Short Description
- Full Description
- Documentation

Short Description

Integrate your hosted Bamboo instance with your JIRA Studio Subversion repository to enable all of the continuous integration benefits of Bamboo for your development suite.

Full Description

Integrating your hosted Bamboo instance with JIRA Studio allows you to run builds on the same source that you are monitoring and reviewing in JIRA Studio. While Atlassian currently does not offer a hosted Bamboo service, you are able to purchase a license for our installed edition and have it independently hosted. Any Bamboo instance can be configured for access to your Studio Subversion repository. To setup Bamboo builds of your JIRA Studio projects:

1. Sign up for Studio
2. Purchase a Bamboo installed license
3. Contact Contegix or another hosting provider to arrange for hosting of your Bamboo instance.
4. Configure Bamboo access to your Studio repository. Your hosting provider may be able to do this for you, otherwise simply open a support ticket in the Studio project for assistance.

Documentation

To set up a hosted instance of Bamboo, you will require a valid Bamboo license and a hosting provider. Contegix are a licensed reseller of Atlassian licenses and are able to offer dedicated hosting. Contact your hosting provider directly to set up your hosted Bamboo instance.

To use your hosted Bamboo instance with your JIRA Studio Subversion repository, please refer to the following documentation:

- Bamboo — Setting up Subversion as the repository for build plan
- JIRA Studio — Subversion repository details

Plugin Development

This page summarises the components of the Atlassian plugin development platform and provides links to more information. Unless otherwise stated, the information here refers to the latest version of the plugin framework. Please follow the links to find out about previous versions.

On this page:

- Short Description
- Full Description
- Components
- Version Matrix
- Documentation

Short Description

Atlassian is moving towards wider and more intimate integration between the different applications we develop. Providing a common plugin framework is a key part of this initiative, because the plugin framework will help developers (both Atlassians and others) to write plugins that work on more than one of the products. It will also allow plugins on each product to have a consistent look and behaviour from a user's point of view, even when the plugin works on only one application.

Full Description

The Atlassian Plugin Framework 2 replaces the original Atlassian Plugins framework, described in the Developer Network. The new framework is based on OSGi, a dynamic module system for Java.

A plugin is a bundle of code, resources and configuration files that can be dropped into an Atlassian product to add new functionality or change the behaviour of existing features.

Every plugin is made up of one or more plugin modules. A single plugin may do many things, while a plugin module represents a single function of the plugin.

There are two versions of plugins in the Atlassian Plugin Framework 2:

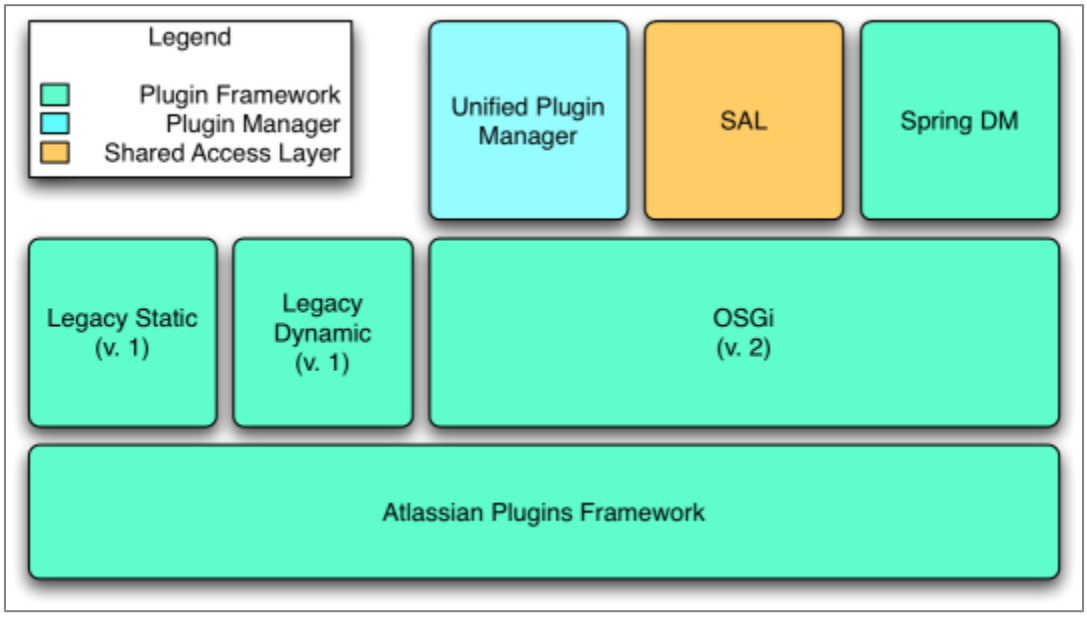
- **Version 1** — These may be static (deployed in `WEB-INF/lib`) or dynamic (via the web UI, only in Confluence) and should work the

same as they did in version 1 of the Atlassian Plugin Framework. The capabilities and features available to version 1 plugins vary significantly across products.
- **Version 2** — These plugins are dynamically deployed on an internal OSGi container to provide a consistent set of features and behaviours, regardless of the application the plugin is running on. Version 2 plugins have to be specifically declared as such, using the `plugins-version="2"` attribute in `atlassian-plugin.xml`.

Components

There are three major components in the Atlassian plugin development platform:

- **Plugin Framework** - The framework that executes the plugins and manages the available plugin modules.
- **Shared Access Layer (SAL)** - The API for accessing common services, regardless of the underlying Atlassian application interfaces.
- **Unified Plugin Manager** - The UI for installing, managing, upgrading, and sharing Atlassian plugins (Not yet available.)



Version Matrix

Almost every Atlassian application has a version of the Atlassian Plugin Framework, however, they may not all have the same version. Eventually, each Atlassian application (Confluence, FishEye, Crucible, Crowd, etc) will be upgraded to version 2.x, but many are running version 1 today. If you are developing a plugin, you need to know what your version is capable of, and how it will interact with other versions.

> **The Atlassian Plugin Framework 2 is available in some, but not all, Atlassian applications**

The matrix below shows the applications which support version 2.x of the Plugin Framework. The applications are listed horizontally across the top and the Plugin Framework 2.x versions are listed vertically on the left.

- Version numbers next to a tick show the **earliest** release of the application which supports the relevant framework version.
- Version numbers in brackets show a future release expected to support the relevant framework version.

Plugin Framework	Bamboo	Confluence	Crowd	Crucible	FishEye	JIRA
2.0			Crowd 1.5	Crucible 1.6	FishEye 1.6	
2.1		Confluence 2.10	Crowd 1.6			
2.2	Bamboo 2.3	Confluence 3.0	Crowd 2.0	Crucible 2.0	FishEye 2.0	
2.3						(JIRA 4.0)

Documentation

Plugin Development Platform

Please refer to the Developer Network:

- Overview of the Plugin Development Platform
- Compatibility of Older and Newer Plugins

Plugin Framework — Latest Version

Please refer to the Plugin Framework documentation:

- Writing Atlassian Plugins
  - Creating your Plugin Descriptor
  - Plugin Module Types
  - Adding Plugin and Module Resources
  - Supporting Minification of JavaScript and CSS Resources
  - Adding a Configuration UI for your Plugin
  - Ensuring Standard Page Decoration in your Plugin UI
  - Using Packages and Components Exposed by an Application
  - Running your Plugin in the Reference Implementation
  - OSGi, Spring and the Plugin Framework
- Embedding the Plugin Framework
  - Quick Start Guide to Embedding
  - Exposing Host Components via Spring
  - Using the Built-In Plugin Modules
  - Writing a Plugin Module Descriptor
  - License and Copyright for the Atlassian Plugin Framework
- Plugin Framework Release Notes
  - Plugin Framework 2.3.1 Release Notes
  - Plugin Framework 2.3 Release Notes
  - Plugin Framework 2.2.4 Release Notes
  - Plugin Framework 2.2.3 Release Notes
  - Plugin Framework 2.2.2 Release Notes
  - Plugin Framework 2.2.1 Release Notes
  - Plugin Framework 2.2 Release Notes
  - Plugin Framework 2.1.4 Release Notes
  - Plugin Framework 2.1.2 Release Notes
  - Plugin Framework 2.1.1 Release Notes
  - Plugin Framework 2.1 Release Notes
  - Plugin Framework 2.0.5 Release Notes
  - Plugin Framework 2.0.4 Release Notes
  - Plugin Framework 2.0.3 Release Notes
- Plugin Framework Version Matrix
- Plugin Framework Glossary
  - Bundle (Glossary Entry)
  - Complex Plugin (Glossary Entry)
  - Cross-Product Plugin (Glossary Entry)
  - Dynamic Plugin (Glossary Entry)
  - Lifecycle (Glossary Entry)
  - Plugin (Glossary Entry)
  - Service (Glossary Entry)
  - Service Registry (Glossary Entry)
  - Simple Plugin (Glossary Entry)
  - Static Plugin (Glossary Entry)
  - Version 1 or Version 2 Plugin (Glossary Entry)
- Plugin Framework FAQ
  - Development FAQ
  - Platform FAQ
  - Troubleshooting FAQ

Plugin Framework — All Versions

Plugin Framework 2.3 Documentation
Plugin Framework 2.2 Documentation
Plugin Framework 2.1 Documentation
Plugin Framework 2.0 Documentation

Single Sign-On (SSO) across Web Applications

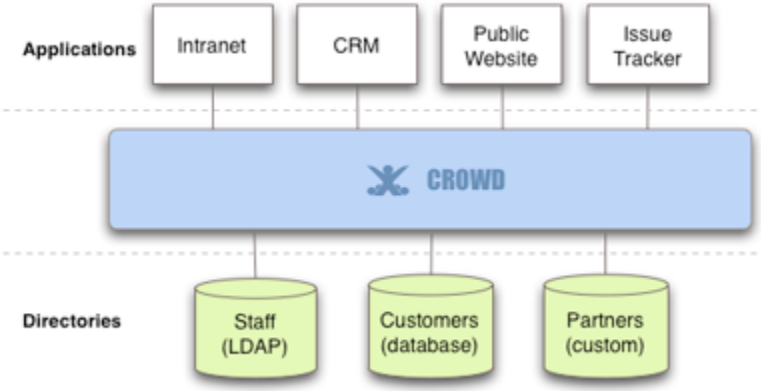
This page summarises the single sign-on (SSO) feature provided by Atlassian Crowd and provides links to more information.

On this page:

- Short Description
- Full Description
- Components
- Documentation

Short Description

Using Crowd for single sign-on (SSO), each person needs only one username and password to access all web applications. You only need to log in once, to Crowd or a Crowd-connected application. When you start another Crowd-connected application, you will be logged in automatically. Similarly, when you log out, you will be logged out of Crowd and the other applications at the same time.



Full Description

Atlassian's Crowd is a software application installed by the system administrator. The administrator will also connect one or more of your organisation's applications to Crowd. When you log in to a Crowd-connected application, Crowd will verify your password and login permissions.

Using Crowd for single sign-on (SSO), each person needs only one username and password to access all web applications. You can host your own OpenID provider to include external applications.

- You only need to log in once, to Crowd or a Crowd-connected application. When you start another Crowd-connected application, you will be logged in automatically.
- When you log out of Crowd or one of the Crowd-connected applications, you will be logged out of Crowd and the other application(s) at the same time.

Crowd also manages the information held about you as a user of other software applications:

- Your login permissions to various applications.
- The password you use to log in to those applications.
- The groups and roles you belong to, which are used by the applications to decide which functions you can perform within the applications.
- The user directories which hold your information.

Components

Type of Integration

Crowd is an application security framework that handles authentication and authorisation for your web-based applications. You will need to download and install Crowd, and then configure it to hook up your applications and user directories.

- More about Crowd
- Download Crowd

Documentation

Latest Version

Please refer to the Crowd documentation:

- Crowd 101
- Crowd Administration Guide
- Crowd User Guide

- Crowd Installation & Upgrade Guide
- Crowd Development Hub
- CrowdID Administration Guide
- CrowdID User Guide
- Crowd FAQ
- Tips of the Trade

All Versions

Crowd 2.0 Documentation
Crowd 1.6 Documentation
Crowd 1.5 Documentation
Crowd 1.4 Documentation
Crowd 1.3 Documentation
Crowd 1.2 Documentation
Crowd 1.1 Documentation
Crowd 1.0 Documentation

User and Group Management across Directories and Web Applications

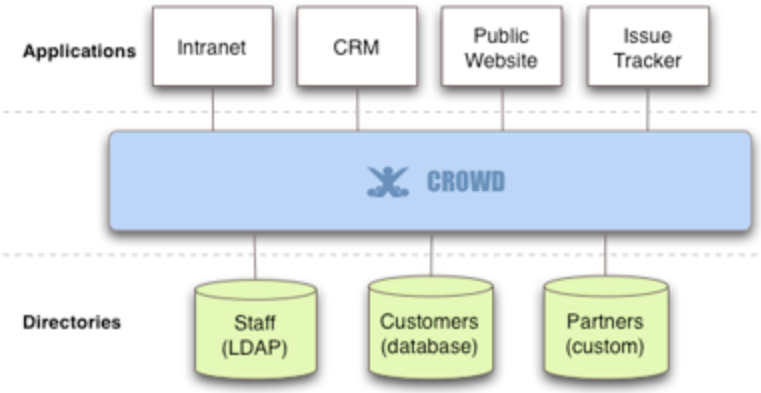
This page summarises the user and group management features offered by Atlassian Crowd and provides links to more information.

On this page:

- Short Description
- Full Description
- Components
- Documentation

Short Description

Use Crowd to manage users and groups simply and centrally. You can use Crowd's internal user directories or hook up your existing LDAP user bases, then link them to your web applications. Crowd provides an Administration Console for administrators as well as a Self-Service Console, allowing users to view their login permissions and to change their own passwords and user profiles.



Full Description

Atlassian's Crowd is a software application installed by the system administrator. The administrator will also connect one or more of your organisation's applications to Crowd. When you log in to a Crowd-connected application, Crowd will verify your password and login permissions.

Using Crowd for single sign-on (SSO), each person needs only one username and password to access all web applications. You can host your own OpenID provider to include external applications.

- You only need to log in once, to Crowd or a Crowd-connected application. When you start another Crowd-connected application, you will be logged in automatically.
- When you log out of Crowd or one of the Crowd-connected applications, you will be logged out of Crowd and the other application(s) at the same time.

Crowd also manages the information held about you as a user of other software applications:

- Your login permissions to various applications.
- The password you use to log in to those applications.

- The groups and roles you belong to, which are used by the applications to decide which functions you can perform within the applications.
- The user directories which hold your information.

Components

Type of Integration

Crowd is an application security framework that handles authentication and authorisation for your web-based applications. You will need to download and install Crowd, and then configure it to hook up your applications and user directories.

- More about Crowd
- Download Crowd

Documentation

Latest Version

Please refer to the Crowd documentation:

- Crowd 101
- Crowd Administration Guide
- Crowd User Guide
- Crowd Installation & Upgrade Guide
- Crowd Development Hub
- CrowdID Administration Guide
- CrowdID User Guide
- Crowd FAQ
- Tips of the Trade

All Versions

Crowd 2.0 Documentation
Crowd 1.6 Documentation
Crowd 1.5 Documentation
Crowd 1.4 Documentation
Crowd 1.3 Documentation
Crowd 1.2 Documentation
Crowd 1.1 Documentation
Crowd 1.0 Documentation

Links and Resources

Links
If you would rather not install everything yourself, try one of our hosted offerings: • JIRA Studio • Enterprise Hosting Looking for more information? Try these links: • Overview of Atlassian Applications • Beyond JIRA • Atlassian Developer Network

Support
Do you have a question, a problem or a suggestion? • If you have a problem and need help from our support team, please raise a ticket on our Support System. • Try our discussion forum for hints and tips from other Atlassian customers. • Check out the training we offer.

Watch this space
This documentation is under development. We will keep adding more information about how you can make your Atlassian applications work together.

About Integrating your Atlassian Applications

Using our open API and scores of plugins, your Atlassian applications will soon put the information you need at your fingertips when you need it.

This documentation is your first port of call for information on integrating your Atlassian applications. The integration points show the many different ways the applications work together and how you can make it happen.

Integration Points

Here are a few of the ways our applications work together:

- Application Links (AppLinks)
- Bamboo Builds in Eclipse
- Bamboo Builds in IDEA
- Crucible Reviews in IDEA
- JIRA Bamboo Integration
- JIRA Issues in IDEA
- JIRA Issues on a Confluence Page
- JIRA Portlets on a Confluence Page
- JIRA Studio Bamboo Integration

Want to see more? Take a look at the full list of integration points.

Atlassian Application Configuration - Documentation Links

This page is a compilation of links to key configuration documents for each of the Atlassian applications. Please note, this is not a supportability matrix. It is intended to be a handy reference to key application documentation only.

*This page is **under development** and may be changed or moved without notice.*

On this page:

- Database Documentation

Database Documentation

PostgreSQL	JIRA 3.13.x	Confluence 3.0	Bamboo 2.3	Crowd 2.0	FishEye 2.0	Crucible 2.0
MySQL	JIRA 3.13.x	Confluence 3.0	Bamboo 2.3	Crowd 2.0	FishEye 2.0	Crucible 2.0

Guide to Installing an Atlassian Integrated Suite

We have put together a guide (see below) to integrating a suite of Atlassian applications. The guide consists of detailed step-by-step instructions for setting up a specific configuration. There are also links to the installation and configuration guides for each component.

Setting up the integrated suite will give you awesome results, but we know that the setup procedure can be long and difficult. So we invite you to join the **Atlassian Dragon Quest**.

- **Starting from scratch?** If you do not have any Atlassian applications installed, please start at Here Be Dragons.
- **Got JIRA?** If you are already using JIRA, please start at Dragon Slayers with JIRA Already Installed.

Here Be Dragons

Beware, all ye who enter, for here be dragons! This is the starting point for the Atlassian Dragon Quest.

By the time you reach the end of this set of instructions, you will have an awesome Atlassian integrated development suite (details below). There's a good chance that the Atlassian Integration Dragon will scorch the clothes off your back somewhere along the way, so we'll also send you a free, limited-edition Atlassian DragonSlayer T-shirt when you have completed all the steps.

Got JIRA? If you are already using JIRA, please start at Dragon Slayers with JIRA Already Installed.

Getting help
If you run into problems at any stage of the integration procedure, please go immediately to the Dragon Slayers' Forum. Please don't try to battle on alone. Instead, ask for help immediately. We'll be monitoring the forum 24 hours a day, and you're sure to meet other battle-weary dragon slayers there too.

Rushing into the Dragon's Lair

Don your armour and alert your serfs
If you like, you can tweet your status.

- Please read the introduction below.

Now you're ready to start stage 1. Meet the dragon if you dare!

- Dragons Stage 1 - Install Java, PostgreSQL and Crowd
- Dragons Stage 2 - Install JIRA
- Dragons Stage 3 - Install GreenHopper into JIRA
- Dragons Stage 4 - Install Confluence
- Dragons Stage 5 - Install FishEye
- Dragons Stage 6 - Get JIRA and FishEye Talking
- Dragons Stage 7 - Install Bamboo
- Dragons Stage 8 - Bamboo Gadgets and JIRA Victory

Follow yon Brave Dragon Slayers
On the Atlassian Dragons Twitter stream.

What's This All About?

We have put together these instructions for integrating a suite of Atlassian applications. Setting up the integrated suite will give you awesome results, but we know that the setup procedure can be long and difficult.

Why would we ask you to undertake this exercise?

- We're issuing a **challenge** to our boldest and most skilled customers and evaluators.
- We'd like to **learn** from your experiences, so that we can improve the setup and integration procedures.
- It's another excuse to give away some **T-shirts**.

The Wins

When you have completed the final stage, you will have set up these Atlassian applications and features to work with each other:

- JIRA for bug tracking.
- GreenHopper for agile project management.
- Confluence, the enterprise wiki.
- FishEye to open up your source repository.
- Bamboo for continuous integration.
- Crowd for user management and single sign-on.

- Atlassian Gadgets.

If you slay the dragon, you win a free, limited-edition Atlassian DragonSlayer T-shirt too.

How Long Will It Take to Slay the Dragon?

We estimate that it will take **5 hours** to complete all the stages.

Getting Kitted Out

Before you start, please note the points below.

Assumptions

- This guide is written for a technical audience. You will need to install a database, install the Atlassian applications and adjust the configuration files.
- This guide assumes that you are starting from scratch, with no Atlassian applications installed or with only JIRA installed.
    - If you can start with a clean slate, with no Atlassian applications at all, please do continue with the integration procedure described on this page and its child pages.
    - If you have JIRA but nothing else, please start at Dragon Slayers with JIRA Already Installed.
    - If you already have Confluence, Crowd, FishEye or Bamboo, please consult our Support team.
- This guide assumes that you will be using a specific database and specific versions of the applications and plugins, as described in each stage of this guide. If you need to use other drivers or application versions, please consult our Support team.

Hardware Requirements

We recommend the following:

- 2GB RAM
- No other applications running — just the operating system, JAVA, PostgreSQL and the Atlassian applications.
- 500MB disk space for application files.

Software Requirements

- **Program for extracting our downloaded archive files:** Please check your unzip/unpack program before extracting any of the Atlassian downloaded zip or archive files. Some unzip/unpack programs cause errors.
    - **Linux** or **Unix** users can use any unpack program.
    - **Solaris** users must use GNU Tar instead of Solaris Tar.
    - **Windows** users should use a third-party unzip program like 7Zip or Winzip. If you do not have one, please download and install one before continuing:
        - 7Zip — Recommended. If in doubt, download the '32-bit .exe' version
        - Winzip
- **Operating System:** The instructions are for Windows, UNIX and Linux. We do not offer instructions for Mac OS X as it is not a platform preferred by our customers. If you have specific questions please seek assistance on the Dragon Slayers' Forum.
- **Application server:** By following our instructions, you will set up a standalone version of each Atlassian product, using the default Tomcat or Jetty server provided with each application.
- **Database:** By following our instructions, you will set up a PostgreSQL database server in stage 1 and use the database server in all subsequent stages.
- **Source repository:** For the purposes of this integration exercise, we have provided a read-only Subversion repository that you can connect to your FishEye and Bamboo installations. We recommend this repository because:
    - We have committed a code change with a JIRA issue key in the commit message. This will allow you to see the JIRA and FishEye integration immediately, without having to do your own commit.
    - The sample repository is small, so that FishEye's initial repository indexing process will be fast.
- **Build tool:** For the Bamboo integration stages you will need a build tool, also called a builder. For this integration exercise, we assume that you are using Maven 2. You can use any of the build tools supported by Bamboo, such as Maven 1, Maven 2, Ant, PHPUnit and others. See the Bamboo documentation.
- **Java Development Kit:** You will need Sun JDK 1.5 or higher. Note that the JRE alone is not enough. Stage 1 of these instructions will guide you through the installation process.

Other Notes

- **Virus checkers:** If you have a virus checker running, there may be a delay in the availability of JAR files after you have placed a required JAR into a directory, while the virus checker scans the file. This may happen with the PostgreSQL database driver files, for example. If you receive an error saying that a driver or other such file is not available, wait a few minutes and try again.
- **Passwords:** At several points in this integration procedure you will need to enter a password. The password will be used to secure your data. The password you choose is up to you, but it is important you pick something that is hard to guess. Take a moment now to think of a password. Here are some guidelines from AusCERT on choosing a good password. This will save you time later.

Rush into the dragon's lair.

Dragons Stage 1 - Install Java, PostgreSQL and Crowd

Beware, all ye who enter, for here be dragons. You are embarking on stage 1 of the Atlassian Dragon Quest.

In this stage, you will install Java and a database (PostgreSQL) to hold the data for your Atlassian applications. Then you will set up Atlassian Crowd for centralised user management and single sign-on (SSO).

Time estimate: This stage will take approximately **60 minutes**.

On this page:

- Step 1. Install Java
- Step 2. Install your PostgreSQL Database Server
- Step 3. Create your Crowd Database in PostgreSQL
- Step 4. Install Crowd
- Step 5. Set Up Crowd
- Victory!

Step 1. Install Java

Requirements: **Sun JDK 1.5 or higher**. Note that the JRE alone is not enough.

If you do not have the right version of the Java Development Kit (JDK) already installed, follow the steps below to get it.

1. Download the Sun Java SE Development Kit – Get the latest JDK 6.
2. Follow the Sun installation instructions.
3. Make sure you have a `JAVA_HOME` environment variable pointing to the root directory of the JDK. Some JDK installers set this automatically.
    - Check by typing one of the following into a command window, depending on your operating system.
        - On Windows: `echo %JAVA_HOME%`
        - On Linux or UNIX: `echo $JAVA_HOME`
    - If the above command does not show you the path to your JDK, please refer to the Crowd instructions on setting JAVA_HOME.

Step 2. Install your PostgreSQL Database Server

Requirements: **PostgreSQL version 8.3.x**.

1. Download PostgreSQL – Get the latest 8.3.x. For the simplest installation, choose one of the one-click installers.
2. Install PostgreSQL. If you chose one of the PostgreSQL one-click installers, this is simple: Run the executable that you downloaded and follow the prompts. If necessary, you can refer to the PostgreSQL installation instructions.
3. Enter a password for the super user ('postgres').
4. Accept the default port 5432.
5. Accept all the other default settings.
6. Download the PostgreSQL JDBC driver from http://jdbc.postgresql.org/download.html and save it locally for later use.
    - If you have installed JDK 6.x, get JDBC4 Postgresql Driver, Version 8.4-701.
    - If you have JDK 5.x, get JDBC3 Postgresql Driver, Version 8.4-701.

Step 3. Create your Crowd Database in PostgreSQL

Now you will create a database where the Atlassian Crowd application will store its data, and the user that Crowd will use to connect to the database.

We're using pgAdmin III, the administration user interface supplied with PostgreSQL. If you used the one-click installer, pgAdmin III will be already installed on your computer.

1. Start pgAdmin III.
2. Add a new login role called 'crowduser':
    - Right-click '**Login Roles**' and select '**New Login Role**'.
    - Enter the role '**Role name**': `crowduser`.
    - Enter a '**Password**' and enter it again to confirm it.
    - Select '**Can create database objects**'.
    - Select '**Can create roles**'.
    - Click '**OK**' to create the user.
3. Add a new database called 'crowd':
    - Right-click '**Databases**' and select '**New Database**'.
    - Enter the database '**Name**': `crowd`.
    - Select the '**Owner**': `crowduser`.
    - Click '**OK**' to create the database.

Alternatively, If you are on UNIX and do not have pgAdmin III, you can use the command line interface instead. Assuming that you are using the default installation directory of `/opt/PostgreSQL/8.3/bin/`, enter the following commands:

```
sudo -s -H -u postgres
# Create the Crowd user:
/opt/PostgreSQL/8.3/bin/createuser -S -d -r -P -E crowduser
# Create the Crowd database:
/opt/PostgreSQL/8.3/bin/createdb -O crowduser crowd
exit
```

Screenshot 1 (click to enlarge): Crowd database and user in PostgreSQL

Step 4. Install Crowd

Requirements: **Crowd 2.0.2**.

For Windows: (click to expand)

1. Go to the Atlassian download centre.
2. Download the '**Standalone (ZIP Archive)**' file for **Crowd 2.0.2**.
3. Unpack the zip archive into a directory of your choice, avoiding spaces in the directory name.
4. Tell Crowd where to find its Crowd Home directory:
    - Edit the properties file at `{CROWD_INSTALL}\crowd-webapp\WEB-INF\classes\crowd-init.properties`.
    - Complete the following line and remove the # at the beginning of the line:
      `crowd.home=`
      For example:
      `crowd.home=c:/data/crowd-home`
      (Note the forward slashes.)
5. Add the PostgreSQL JDBC driver JAR to your `{CROWD_INSTALL}\apache-tomcat\lib` directory.
6. Start your Crowd server by running `start_crowd.bat` in the directory where you unpacked Crowd.

For UNIX or Linux: (click to expand)

1. Go to the Atlassian download centre.
2. Click the 'Linux' tab and download the '**Standalone (TAR.GZ Archive)**' file for **Crowd 2.0.2**.
3. Unpack the archive into a directory of your choice, avoiding spaces in the directory name.
4. Tell Crowd where to find its Crowd Home directory:
    - Edit the properties file at `{CROWD_INSTALL}/crowd-webapp/WEB-INF/classes/crowd-init.properties`.
    - Complete the following line and remove the # at the beginning of the line:
      `crowd.home=`
      For example:
      `crowd.home=/var/crowd-home`
5. Create the above Crowd Home directory if it does not already exist, because in some cases Crowd may not create it for you.
6. Add the PostgreSQL JDBC driver JAR to your `{CROWD_INSTALL}/apache-tomcat/lib` directory.
7. Start your Crowd server by executing `start_crowd.sh` in the directory where you unpacked Crowd.

Full details are in the Crowd installation guide.

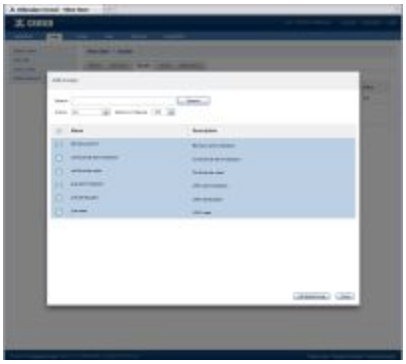
Problems? Please go immediately to the Dragon Slayers' Forum.
Victory? Please continue.

Step 5. Set Up Crowd

Now you can run Crowd's Setup Wizard, then add **Charlie of Atlassian** and the groups needed for JIRA, Confluence and the other applications.

1. To access Crowd, go to your web browser and type this address: http://localhost:8095/crowd.
2. The Crowd Setup Wizard will start up, to guide you through the process of setting up your Crowd server and creating an administration user. Detailed instructions are in the Crowd documentation. Here are the things you need to know for our Dragon Quest:
    - License – If you do not already have a Crowd license, follow the prompts on the Setup Wizard screen to get an evaluation license key.
    - Installation type – Select '**New Installation**'.
    - Database configuration – Select '**JDBC Connection**' then enter the following information to connect to your Crowd database (created above):
        - Database: `PostgreSQL`.
        - Driver Class Name – Leave this at the default value, i.e. `org.postgresql.Driver`.
        - JDBC URL – Leave this at the default value, i.e. `jdbc:postgresql://localhost:5432/crowd`.
        - Username: `crowduser`.
        - Password – The password you specified when creating your Crowd database above.
        - Hibernate Dialect – Leave this at the default value, i.e. `org.hibernate.dialect.PostgreSQLDialect`.
    - Deployment title – Enter a short, descriptive name. If you will only have one Crowd installation, then 'Crowd' is good enough.
    - Base URL – Enter the full website address at which Crowd is running, not just 'localhost'. For example, if your computer name is 'coopers' then the base URL should be: `http://coopers:8095/crowd`. Or specify a website address, such as `http://www.foobar.com:8095/crowd`
    - Email details – Enter the details of your administrator email account. We recommend that you give your own email account details here.
    - Internal directory – This is the Crowd directory that will hold your users and groups. Enter the following information, and leave the other fields at the default values:
        - Name: `Crowd`.
        - Description: `Crowd User Directory`.
    - Default administrator – This is the Crowd super user. Enter the following information:
        - Email address – Enter the address of your administrator email account. We recommend that you give your own email address here.
        - Username – Enter the administrator's login name: **`charlie`**.
        - Password – Enter a password for the administrator account and enter it again to confirm it.
        - Enter a first name for your administrator: **`Charlie`**.
        - Enter a last name for your administrator: **`of Atlassian`**.
    - Integrated applications – Leave both selected, as is the default.
3. Log in to Crowd with username **`charlie`**.
4. Add the group that will hold all your JIRA users:
    - Click '**Groups**' the top navigation bar and then click '**Add Group**'.
    - Enter the following information:
        - Group name: `jira-users`.
        - Description: `JIRA users`.
        - Directory: `Crowd`.
        - Active – Leave this checkbox selected.
    - Click '**Create**' to add the group.
5. Add the following groups too, all in the same 'Crowd' directory. These groups are needed for JIRA, Confluence and Bamboo:
    - `jira-developers` — `JIRA developers`
    - `jira-administrators` — `JIRA administrators`
    - `confluence-users` — `Confluence users`
    - `confluence-administrators` — `Confluence administrators`
    - `bamboo-admin` — `Bamboo administrators`
6. Make **Charlie of Atlassian** an administrator in JIRA, Confluence and Bamboo by adding him to the relevant groups:
    - Click '**Users**' in the the top navigation bar and find '**Charlie of Atlassian**'.
    - Click the name to view Charlie's user information.
    - Click the '**Groups**' tab, then click '**Add Groups**'.
    - The 'Add Groups' screen will appear. Click '**Search**' to see all the groups in the directory.
    - Select the checkbox at top left, next to the 'Name' column, to select all groups.
    - Click '**Add Selected Groups**' to add Charlie to the groups.

Screenshot 2 (click to enlarge): Adding Charlie to groups in Crowd



Problems? Please go immediately to the Dragon Slayers' Forum.
Victory? Please continue.

Victory!

Charlie of Atlassian can now log into Crowd. If he checks his profile (using the 'My Profile' link at top right of the Crowd screen), he will see the groups he belongs to.

Screenshot 3 (click to enlarge): Charlie's profile showing the groups he belongs to

Problems? Please go immediately to the Dragon Slayers' Forum.
Victory? Please continue.

Don your Belt and Boots, and Move to the Next Stage

- **Tweet?** Tweet.
- Go to Dragons Stage 2 - Install JIRA.

Dragons Stage 2 - Install JIRA

Beware of fiends and dragons on the gargoyled eaves. You are embarking on stage 2 of the Atlassian Dragon Quest.

In this stage, you will install Atlassian JIRA for bug tracking and issue management. You will also hook JIRA up to Crowd, for SSO and centralised user management.

Time estimate: This stage will take approximately **60 minutes**.

On this page:

- Step 1. Create your JIRA Database in PostgreSQL
- Step 2. Install JIRA
- Step 3. Set Up JIRA
- Step 4. Hook JIRA up to Crowd
- Step 5. Set up a Project and Create your JIRA Dashboard
- Victory!

Step 1. Create your JIRA Database in PostgreSQL

Now you will create a database where the Atlassian JIRA application will store its data, and the user that JIRA will use to connect to the database. We are assuming that you have already created your PostgreSQL database server in **Dragons Stage 1**.

We are using pgAdmin III, the administration user interface supplied with PostgreSQL. If you used the one-click installer when installing PostgreSQL, pgAdmin III will be already installed on your computer.

1. Start **pgAdmin III**.
2. Add a new login role called 'jirauser':
   - Right-click '**Login Roles**' and select '**New Login Role**'.
   - Enter the role '**Role name**': `jirauser`.
   - Enter a '**Password**' and enter it again to confirm it.
   - Select '**Can create database objects**'.
   - Select '**Can create roles**'.
   - Click '**OK**' to create the user.
3. Add a new database called 'jira':
   - Right-click '**Databases**' and select '**New Database**'.
   - Enter the database '**Name**': `jira`.
   - Select the '**Owner**': `jirauser`.
   - Click '**OK**' to create the database.

Alternatively, If you are on UNIX and do not have pgAdmin III, you can use the command line interface instead. Assuming that you are using the default installation directory of `/opt/PostgreSQL/8.3/bin/`, enter the following commands:

```
sudo -s -H -u postgres
# Create the JIRA user:
/opt/PostgreSQL/8.3/bin/createuser -S -d -r -P -E jirauser
# Create the JIRA database:
/opt/PostgreSQL/8.3/bin/createdb -O jirauser jira
exit
```

Screenshot 1 (click to enlarge): JIRA database and user in PostgreSQL

Step 2. Install JIRA

Requirements: **JIRA 4.0.0**.

For Windows: (click to expand)

1. Go to the Atlassian download centre.
2. Click the '**Show all**' link above the download buttons, to see all the download file types.
3. Download the '**Standalone (ZIP Archive)**' file for **JIRA 4.0.0**.

   ⚠ Do *not* use the 'Windows Installer' for this integration exercise, because the workflow for configuring an external database is simpler when installing from the zip archive.
4. Unpack the zip archive into a directory of your choice, avoiding spaces in the directory name.
5. Tell JIRA where to put its JIRA Home directory:
   - Edit the properties file at {`JIRA_INSTALL}\atlassian-jira\WEB-INF\classes\jira-application.properties`.
   - Complete the following line:
     `jira.home =`
     For example:
     `jira.home = C:/data/jira-home`
     (Note the forward slashes.)
   - Save the file.
6. Connect JIRA to your PostgreSQL database:
   - Add the PostgreSQL JDBC driver JAR (downloaded in Dragons Stage 1) to your `{JIRA_INSTALL}\common\lib` directory.
   - Edit the configuration file at `{JIRA_INSTALL}\conf\server.xml` and update the following attributes in the JDBC `Resource` element:
     - `username="jirauser"` – This is the user you created in step 1 above.
     - `password="secret"` – Enter the password you chose in step 1 above.
     - `driverClassName="org.postgresql.Driver"`
     - `url="jdbc:postgresql://localhost:5432/jira"`
   - Delete the following two attributes:
     - `minEvictableIdleTimeMillis="4000"`
     - `timeBetweenEvictionRunsMillis="5000"`
   - The `Resource` element now looks like this:

```
<Resource name="jdbc/JiraDS" auth="Container" type="javax.sql.DataSource"
          username="jirauser"
          password="secret"
          driverClassName="org.postgresql.Driver"
          url="jdbc:postgresql://localhost:5432/jira"
          maxActive="20" />
```

   - Save the configuration file.
   - Edit the configuration file at `{JIRA_INSTALL}\atlassian-jira\WEB-INF\classes\entityengine.xml` and update the following attributes in the `datasource` element:
     - `field-type-name="postgres72"`
     - `schema-name="public"` – Note the change from upper case "PUBLIC" to lower case "public".
   - The `datasource` element now looks like this:

```
<datasource name="defaultDS" field-type-name="postgres72"
          schema-name="public"
          helper-class="org.ofbiz.core.entity.GenericHelperDAO"
          check-on-start="true"
          use-foreign-keys="false"
          use-foreign-key-indices="false"
          check-fks-on-start="false"
          check-fk-indices-on-start="false"
          add-missing-on-start="true"
          check-indices-on-start="true">
   <jndi-jdbc jndi-server-name="default" jndi-name=
     "java:comp/env/jdbc/JiraDS"/>
</datasource>
```

   - Save the configuration file.
7. Start your JIRA server by running `{JIRA_INSTALL}\bin\startup.bat`.

For UNIX or Linux: (click to expand)

1. Go to the Atlassian download centre.
2. Click the 'Linux' tab and download the '**Standalone (TAR.GZ archive)**' file for **JIRA 4.0.0**.
3. Unpack the archive into a directory of your choice, avoiding spaces in the directory name.
4. Tell JIRA where to put its JIRA Home directory:
    - Edit the properties file at {`JIRA_INSTALL}/atlassian-jira/WEB-INF/classes/jira-application.properties`.
    - Complete the following line:
      `jira.home =`
      For example:
      `jira.home = /var/jira-home`
    - Save the file.
5. Connect JIRA to your PostgreSQL database:
    - Add the PostgreSQL JDBC driver JAR (downloaded in Dragons Stage 1) to your `{JIRA_INSTALL}/common/lib` directory.
    - Edit the configuration file at `{JIRA_INSTALL}/conf/server.xml` and update the following attributes in the JDBC `Resource` element:
        - `username="jirauser"` – This is the user you created in step 1 above.
        - `password="secret"` – Enter the password you chose in step 1 above.
        - `driverClassName="org.postgresql.Driver"`
        - `url="jdbc:postgresql://localhost:5432/jira"`
    - Delete the following two attributes:
        - `minEvictableIdleTimeMillis="4000"`
        - `timeBetweenEvictionRunsMillis="5000"`
    - The `Resource` element now looks like this:

```
<Resource name="jdbc/JiraDS" auth="Container" type="javax.sql.DataSource"
          username="jirauser"
          password="secret"
          driverClassName="org.postgresql.Driver"
          url="jdbc:postgresql://localhost:5432/jira"
          maxActive="20" />
```

    - Save the configuration file.
    - Edit the configuration file at `{JIRA_INSTALL}/atlassian-jira/WEB-INF/classes/entityengine.xml` and update the following attributes in the `datasource` element:
        - `field-type-name="postgres72"`
        - `schema-name="public"` – Note the change from upper case "PUBLIC" to lower case "public".
    - The `datasource` element now looks like this:

```
<datasource name="defaultDS" field-type-name="postgres72"
          schema-name="public"
          helper-class="org.ofbiz.core.entity.GenericHelperDAO"
          check-on-start="true"
          use-foreign-keys="false"
          use-foreign-key-indices="false"
          check-fks-on-start="false"
          check-fk-indices-on-start="false"
          add-missing-on-start="true"
          check-indices-on-start="true">
   <jndi-jdbc jndi-server-name="default" jndi-name=
     "java:comp/env/jdbc/JiraDS"/>
</datasource>
```

    - Save the configuration file.
6. Start your JIRA server by running `{JIRA_INSTALL}/bin/startup.sh`.

Full details are in the JIRA installation guide.

Problems? Please go immediately to the Dragon Slayers' Forum.
Victory? Please continue.

Step 3. Set Up JIRA

Now you can run JIRA's Setup Wizard and then enable some JIRA features that are required for the later stages in this integration procedure.

1. To access JIRA, go to your web browser and type this address: http://localhost:8080.
2. The JIRA Setup Wizard will start up, to guide you through the process of setting up your JIRA server and creating an administration user. Detailed instructions are in the JIRA documentation. Here are the things you need to know for our Dragon Quest:
    - Base URL – Enter the full website address at which JIRA is running, not just 'localhost'. For example, if your computer name is 'coopers' then the base URL should be: `http://coopers:8080`. Or specify a website address, such as `http://www.foobar.com:8080`.
    - Leave all the default directories selected.
    - License – If you do not already have a JIRA license, follow the prompts on the Setup Wizard screen to get an evaluation license key.

      Make sure you have a **JIRA 4** license. Existing 3.x licenses will not work.
    - Administrator account – This is the JIRA super user, and should be the same as the Crowd super user entered in Dragons Stage 1. Enter the following information:
        - Username: **`charlie`**.
        - Password – Enter a password for the administrator account and enter it again to confirm it.
        - Full name: **`Charlie of Atlassian`**.
        - Email address – We recommend that you give your own email address here.
    - Email notifications – For the purposes of the Atlassian Dragon Quest, we recommend that you **disable email notifications**.
3. Log in to JIRA with username **`charlie`** and perform the following configuration steps:
    a. Configure JIRA to use the wiki renderer for comments and descriptions:
        - Click '**Administration**' in the top navigation bar.
        - Click '**Field Configurations**' in the left-hand panel (in the 'Issue Fields' section).
        - The 'View Field Configurations' screen will appear. Click '**Configure**' next to 'Default Field Configuration'.
        - The 'View Field Configuration' screen will appear.
        - To change the renderer for comments, click '**Renderers**' next to '**Comment**' and then select '**`Wiki Style Renderer`**' as the 'Active Renderer'. Click '**Update**' and then click it again to confirm the change.
        - To change the renderer for issue descriptions, click '**Renderers**' next to '**Description**' and then select '**`Wiki Style Renderer`**' as the 'Active Renderer'. Click '**Update**' and then click it again to confirm the change.

    b. Turn on time tracking, so that you will be able to log the amount of time you spend working on issues:
        - Click '**Time Tracking**' in the left-hand panel (in the 'Global Settings' section).
        - In the '**Hours Per Day**' field, select the number of hours in your organisation's working day (e.g. 8).
        - In the '**Days Per Week**' field, select the number of days in your organisation's working week (e.g. 5).
        - Leave the '**Time Format**' as '`pretty`' and leave the '**Default Unit**' as '`minute`'.
        - Click '**Activate**'.

    c. Turn on the public API and allow unassigned issues:
        - Click '**General Configuration**' in the left-hand panel (in the 'Global Settings' section).
        - Click '**Edit Configuration**'.
        - Select the '**`ON`**' radio button next to '**Accept remote API calls**'.
        - Select the '**`ON`**' radio button next to '**Allow unassigned issues**'.
        - Click '**Update**'.

    d. Enable sub-tasks:
        - Click '**Sub-Tasks**' in the left-hand panel (in the 'Global Settings' section).
        - The 'Sub-Tasks' screen will appear. Click '**Enable**'.
4. Log out of JIRA, but leave JIRA running. (Click the dropdown arrow next to the name '**Charlie of Atlassian**', then select '**Log Out**'.)

Screenshot 2: The JIRA Dashboard when you first log in

Problems? Please go immediately to the Dragon Slayers' Forum.
Victory? Please continue.

Step 4. Hook JIRA up to Crowd

In this step you will define the JIRA application in Crowd and configure JIRA to use Crowd for SSO and centralised user management.

1. If Crowd is not already running, start it up by running {CROWD_INSTALL}\start_crowd.bat (on Windows) or {CROWD_INSTALL}/start_crowd.sh (on UNIX).
2. Go to your Crowd URL in your browser, e.g. http://www.foobar.com:8095/crowd.
3. Log in to Crowd with username **charlie**.
4. Click '**Applications**' in the top navigation bar.
5. The 'Application Browser' will appear. Click '**Add Application**' in the left-hand menu.
6. This will display the first screen for the 'Add Application' wizard for Crowd. Enter the following information:
   - Application Type: JIRA.
   - Name: jira.
   - Description: Atlassian JIRA.
   - Password – Enter the password that JIRA will use to access Crowd and enter it again to confirm it.
   - URL – Enter the base URL of your JIRA site, e.g. http://www.foobar.com:8080.
   - Click '**Resolve IP Address**' to ask Crowd to find the '**Remote IP Address**' for you. The value will be something like this: 127.0.0.1.
   - Select the '**Crowd**' directory.
   - Select '**Allow all users to authenticate**'.
   - Click '**Add Application**'.
7. Check the IP addresses for your JIRA application:
   - Click the '**Remote Addresses**' tab.
   - Add your JIRA host name, e.g. http://www.foobar.com:8080.
   - If it's not already present, add: 127.0.0.1.
8. Leave Crowd up and running, but shut down JIRA. (Press Ctrl+C in your JIRA server command window or run {JIRA_INSTALL}\bin\shutdown.bat (on Windows) or {JIRA_INSTALL}/bin/shutdown.sh (on UNIX).)
9. Copy the Crowd client libraries and configuration files to your JIRA installation folder:
   - Delete the existing crowd-integration-client-1.6.1.jar file from {JIRA_INSTALL}/atlassian-jira/WEB-INF/lib folder.
   - Copy {CROWD_INSTALL}/client/crowd-integration-client-2.0.1.jar to {JIRA_INSTALL}/atlassian-jira/WEB-INF/lib.
   - Copy {CROWD_INSTALL}/client/conf/crowd.properties to {JIRA_INSTALL}/atlassian-jira/WEB-INF/classes.
   - Copy {CROWD_INSTALL}/client/conf/crowd-ehcache.xml to {JIRA_INSTALL}/atlassian-jira/WEB-INF/classes.
10. Edit the {JIRA_INSTALL}/atlassian-jira/WEB-INF/classes/crowd.properties file and change the following properties:
    - application.name: jira
    - application.password – Enter the password that JIRA will use to access Crowd. This must be the same password as you entered in the Crowd 'Add Application' wizard above.
11. Edit the {JIRA_INSTALL}/atlassian-jira/WEB-INF/classes/osuser.xml file. Comment out any existing authentication providers and uncomment the Crowd providers, as instructed in the text of the file itself.
12. Edit the {JIRA_INSTALL}/atlassian-jira/WEB-INF/classes/seraph-config.xml file. Comment out the 'JiraOsUserAuthenticator' class and uncomment the 'JIRAAuthenticator' class, as instructed in the text of the file itself.
13. Start your JIRA server again, and go to your JIRA URL in your browser, e.g. http://www.foobar.com:8080.
14. Log in to JIRA with username **charlie** and Charlie's password in Crowd.

    You are now authenticating via Crowd!
15. Turn on external user management in JIRA, so that all user management happens in Crowd rather than JIRA:
    - Click '**Administration**' in the top navigation bar.
    - Click '**General Configuration**' in the left-hand panel (in the 'Global Settings' section).
    - Click '**Edit Configuration**'.
    - Change '**Mode**' to '**Private**'.
    - Select the '**ON**' radio buttons next to '**External user management**' and '**External password management**'.
    - Click '**Update**'.

Screenshot 3: The JIRA application defined in Crowd – 'Remote Addresses' tab

Full details are in the Crowd documentation.

Problems? Please go immediately to the Dragon Slayers' Forum.
Victory? Please continue.

Step 5. Set up a Project and Create your JIRA Dashboard

In this step you will create some data in JIRA, including a project and an issue, for use in the subsequent stages of this integration procedure. Then you will create your own JIRA dashboard with a couple of gadgets.

1. Create a project in JIRA:
   - Click '**Administration**' in the top navigation bar.
   - Click '**Projects**' in the left-hand panel, then click '**Add Project**'.
   - Enter the following information:
     - Name: **`Dragons`**.
     - Key: **`DRA`**.
     - Project Lead: **`charlie`**.
     - Description: `Atlassian Dragon Quest`.
   - Leave the rest of the fields with their default values. Click '**Add**'.
2. Add two versions (1.0 and 2.0):
   - Click '**Manage** versions'.
   - Enter the following information then click '**Add**':
     - Version Name: `1.0`.
     - Description: `Version 1.0`.
   - Follow the same steps to add `Version 2.0`.
3. Add an issue to your project:
   - Click '**Create Issue**' at top right of the screen, select the following options then click '**Create**':
     - Project: `Dragons`.
     - Issue Type: `Bug`.
   - Enter the following information about your new issue then click '**Create**':
     - Summary: `Dragon slayer's equipment is defective`
     - Affects Version/s: `1.0`.
     - Assignee: `Charlie of Atlassian` – Click '**Assign to me**'.
     - Description: `There's a hole in the dragon slayer's water bucket.`
     - Original Estimate: `1d`.
   - You now have an issue with a key of '**DRA-1**'.
4. Create a new dashboard for all your dragon-related tasks, issues and general fire fighting:
   - Click '**Dashboards**' at top left of your JIRA screen.
   - Click '**Tools**' at top left of the screen, then '**Create Dashboard**'.
   - The 'Create New Dashboard' screen will appear. Enter the following information:
     - Name: **`Dragon Development Dashboard`**.
     - Description: `A dashboard for dragon slayers, fire fighters and like-minded brave souls.`
   - Leave the other fields at their default values and click the '**Add**' button at the **bottom** of the 'Create New Dashboard' screen (not the one next to 'Add Shares').
5. You now have a new, empty dashboard. Add the 'Projects' gadget to the dashboard:
   - Click '**Add Gadget**'.
   - The 'Gadget Directory' will appear, showing a list of the available gadgets for your JIRA dashboard. Enter 'projects' into the search box at top right of the Gadget directory screen.
   - The list of gadgets will change, to show only the gadgets that match your search term. Find the '**Projects**' gadget and click '**Add it Now**'. The gadget will be highlighted for a short time and the button's wording will change to 'Adding', while JIRA adds the gadget to the dashboard.
6. Find and add the '**Assigned To Me**' gadget in the same way.
7. Click '**Finished**' to go back to your dashboard.
8. Drag the 'Assigned to Me' gadget to the top right of your dashboard:
   - Move your mouse pointer over the gadget's blue title bar.
   - The cursor icon will change to a four-pointed arrow. Click the gadget title bar with the left mouse button then drag the gadget to the right. Drop it in the space labelled 'Drag your gadget here.'
9. Configure the 'Assigned to Me' gadget to point to your 'Dragons' project:
   - Refresh the dashboard, if necessary, to show the 'Number of Results' and other configuration fields in the gadget.
   - Leave the default values as configured for '**Number of Results**' and '**Columns to display**'.
   - Click the dropdown arrow next to '**Refresh Interval**' and select '**Every 15 Minutes**'.
   - Click '**Save**'.
10. Configure the 'Projects' gadget:
    - Leave the default values as configured for '**Projects**', '**View**' and '**Number of Columns**'.
    - Click the dropdown arrow next to '**Refresh Interval**' and select '**Every 15 Minutes**'.
    - Click '**Save**'.

Problems? Please go immediately to the Dragon Slayers' Forum.
Victory? Please continue.

Victory!

You can now see your project dashboard with 2 gadgets on it! The 'Projects' gadget shows the project lead **Charlie of Atlassian**. The 'Assigned to Me' gadget shows the single **DRA-1** issue assigned to Charlie.

Screenshot 4 (click to enlarge): JIRA dashboard with 2 gadgets

Problems? Please go immediately to the Dragon Slayers' Forum.
Victory? Please continue.

Don your Chain Mail and Move to the Next Stage

- **Tweet?** Tweet.
- Go to Dragons Stage 3 - Install GreenHopper into JIRA.

Dragons Stage 3 - Install GreenHopper into JIRA

Beware of low-flying worms. You are embarking on stage 3 of the Atlassian Dragon Quest.

In this stage, you will install GreenHopper into JIRA, for agile project management.

Time estimate: This stage will take approximately **30 minutes**.

On this page:

- Step 1. Install GreenHopper Plugin into JIRA
- Step 2. Add Another JIRA Issue and a Sprint
- Step 3. Add the GreenHopper Gadget to your JIRA Dashboard
- Victory!

Step 1. Install GreenHopper Plugin into JIRA

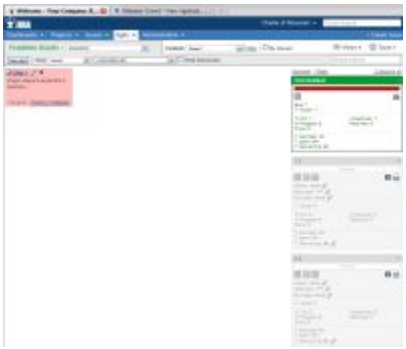
Requirements: **GreenHopper 4.0 for JIRA 4.0.x**.

1. Go to the Atlassian download centre.
2. Download **GreenHopper 4.0 for JIRA 4.0.x**.
3. Shut down your JIRA server. (Press Ctrl+C in your JIRA server command window or run {`JIRA_INSTALL}\bin\shutdown.bat` (on Windows) or {`JIRA_INSTALL}/bin/shutdown.sh` (on UNIX).)
4. Copy the downloaded Greenhopper JAR file into your {`JIRA_HOME}/plugins/installed-plugins` directory, where {`JIRA_HOME`} is the JIRA Home directory that you specified when installing JIRA in Dragons Stage 2. For example:
    - On Windows: `C:\data\jira-home\plugins\installed-plugins`
    - On UNIX: `/var/jira-home/plugins/installed-plugins`
5. Start your JIRA server again, and go to your JIRA URL in your browser, e.g. http://www.foobar.com:8080.
6. Log in to JIRA with username **`charlie`**.
7. Set up your GreenHopper license key:
    - Click '**Administration**' in the top navigation bar.
    - Click '**GreenHopper Licence**' in the left-hand panel (in the 'System' section).
    - The 'GreenHopper License Information' screen will appear. Paste your Greenhopper license key into the '**GreenHopper Licence**' textbox. If you do not already have a GreenHopper license, follow the prompts on the 'GreenHopper License' screen to get an evaluation license key.

      Make sure you have a **GreenHopper 4** license. Existing 3.x licenses will not work.
    - Click '**Add**'.
8. Click '**Agile**' in the top navigation bar.

   You will see the **DRAGONS** planning board, supplied by GreenHopper in JIRA.
   (If you do not see the planning board, click the down arrow next to 'Agile' and select '**Planning Board**'.)

Screenshot 1 (click to enlarge): The GreenHopper planning board in JIRA



There's more about getting started with GreenHopper in the GreenHopper documentation.

Problems? Please go immediately to the Dragon Slayers' Forum.
Victory? Please continue.

Step 2. Add Another JIRA Issue and a Sprint

Now that you have GreenHopper, you can choose to add JIRA issue and do other updates via GreenHopper or via the standard JIRA interface. Now that you have GreenHopper you can choose to add, edit and move JIRA issues through the workflow stages via the Planning and Task Boards under the Agile tab or via the standard JIRA interface. For this exercise, you will do your updates via GreenHopper. First you will create a couple of 'sprints', also known as 'milestones'. A sprint is a short period of time, e.g. two weeks, in which your developers focus on a particular set of tasks. Then you will create a new issue and include it in one of the sprints, and add your existing issue to the same sprint.

1. Click '**Manage**' on the planning board.
2. The 'Manage Versions' screen will appear. Add a sprint with the following information:
    - Version Name: `2.0.S1`
    - Description: `Version 2.0 Sprint 1`
3. Add another sprint with the following information:
    - Version Name: `2.0.S2`
    - Description: `Version 2.0 Sprint 2`
4. Click '**Agile**' in the top navigation bar to go back to your planning board.
5. Your two new sprints will appear as boxes on the right of the planning board, underneath the '**2.0**' box. Now you need to include the two sprints into the existing version 2.0. Click the '**Edit Master**' icon next to '**Master**' in the box for sprint '**2.0.S1**'.
6. A dropdown list will appear. Select '**2.0**'.
7. The '**2.0.S1**' sprint will become part of version 2.0 – the gap between the boxes will disappear and a small downward and rightward-pointing arrow will appear next to the heading '2.0.S1'.
8. Edit the '**Master**' for sprint '**2.0.S1**' in the same way.
9. You now have two sprints within version 2.0. Next, you need to add a new issue (card). Click '**New Card**' on the planning board. Enter the following values:
    - Card type: `Bug`
    - Priority: `Blocker`
    - Summary: `Exploding flame extinguishers`
    - Version: `Unscheduled`
    - Component: `Unknown`
    - Original estimate: `2d`
    - Assignee: `Charlie of Atlassian` – Click the '**Assign to me**' icon.
10. Click '**Create and Close**', to create the issue. You will see your planning board again.
11. Click the card for your existing issue **DRA-1**, drag the card to the right and drop it onto the box for sprint '**2.0.S1**'.
12. Drag **DRA-2** to sprint '**2.0.S1**' as well.
13. Your planning board will now be empty. Select version '**2.0.S1**' in the version dropdown list at the top of your planning board. You should now see your two cards, i.e. issues '**DRA-1**' and '**DRA-2**'.
14. Click the version number '**2.0**' at the top of the version 2.0 box on the right. Notice the following points:
    - The number in the version dropdown box at the top of the planning board also changes to '2.0'.
    - Your two issue cards are included in version 2.0 as well as in sprint 2.0.S1.
    - You can click the down arrows at top right of each version box, to minimise the box.
15. Now you can mark one of your issues as complete:
    - Click the down arrow next to 'Agile' and select '**Task Board**'.
    - Your task board will appear, with your two issue cards in the 'TO DO' column on the left. Click the card for '**DRA-1**', drag it to the right and drop it in the '**DONE**' column.
    - The '**Transitioning Issue**' screen will appear. Click '**Close Issue**' and leave all other values at their defaults.
16. Go back to your planning board to see the changes reflected there too.

Screenshot 2 (click to enlarge): The GreenHopper planning board for version 2.0

Problems? Please go immediately to the Dragon Slayers' Forum.
Victory? Please continue.

Step 3. Add the GreenHopper Gadget to your JIRA Dashboard

Now you will add the GreenHopper 'Agile' gadget to your **Dragon Development Dashboard**.

1. Click '**Dashboards**' at top left of your JIRA screen.
2. Your 'Dragon Development Dashboard' will appear. Click '**Add Gadget**'.
3. The 'Gadget Directory' will appear, showing a list of the available gadgets for your JIRA dashboard. Enter 'agile' into the search box at top right of the Gadget directory screen.
4. The list of gadgets will change, to show only the gadgets that match your search term. Find the '**Agile**' gadget and click '**Add it Now**'. The gadget will be highlighted for a short time and the button's wording will change to 'Adding', while JIRA adds the gadget to the dashboard.
5. Click '**Finished**' to go back to your dashboard.
6. Configure the '**Agile**' gadget:
    - Select '**Dragons**' in the '**Projects**' dropdown list.
    - Leave the default value for '**Display chart values**'.
    - Click the dropdown arrow next to '**Refresh Interval**' and select '**Every 15 Minutes**'.
    - Click '**Save**'.
    - Click the version dropdown arrow and select '**2.0**'.
    - The gadget will display the '**Hours**' burndown chart. Click the '**Issues**' tab to see the issues burndown chart. (The burndown charts will become more interesting when you have more issues in your project.)
7. Choose a different colour for your '**Agile**' gadget:
    - Move your cursor pointer over the gadget and click the downward-pointing arrow at top right of the gadget frame.
    - Select the **green** square in the row of colours.

Problems? Please go immediately to the Dragon Slayers' Forum.
Victory? Please continue.

Victory!

Your JIRA dashboard now has 3 gadgets:

- The GreenHopper 'Agile' gadget
- the 'Assigned to Me' gadget
- The 'Projects' gadget

Screenshot 3 (click to enlarge): JIRA dashboard with 3 gadgets

Problems? Please go immediately to the Dragon Slayers' Forum.
Victory? Please continue.

Grab your Sword and Move to the Next Stage

- **Tweet?** Tweet.
- Go to Dragons Stage 4 - Install Confluence.

Dragons Stage 4 - Install Confluence

There will be much flapping of wings and breathing of fire. You are embarking on stage 4 of the Atlassian Dragon Quest.

In this stage, you will install Atlassian Confluence, the enterprise wiki. You will create a wiki space and add a dynamic list of JIRA issues to a wiki page. You will also hook Confluence up to Crowd for SSO and centralised user management.

Time estimate: This stage will take approximately **30 minutes**.

On this page:

- Step 1. Create your Confluence Database in PostgreSQL
- Step 2. Install Confluence
- Step 3. Set Up Confluence
- Step 4. Hook Confluence up to Crowd
- Step 5. Configure JIRA to Trust Confluence
- Step 6. Create a Wiki Space
- Victory!

Step 1. Create your Confluence Database in PostgreSQL

Now you will create a database where the Atlassian Confluence application will store its data, and the user that Confluence will use to connect to the database. We are assuming that you have already created your PostgreSQL database server in Dragons Stage 1.

We are using pgAdmin III, the administration user interface supplied with PostgreSQL. If you used the one-click installer in Dragons Stage 1, pgAdmin III will be already installed on your computer.

1. Start **pgAdmin III**.
2. Add a new login role called 'confuser':
   - Right-click '**Login Roles**' and select '**New Login Role**'.
   - Enter the role '**Role name**': `confuser`.
   - Enter a '**Password**' and enter it again to confirm it.
   - Select '**Can create database objects**'.
   - Select '**Can create roles**'.
   - Click '**OK**' to create the user.
3. Add a new database called 'confluence':
   - Right-click '**Databases**' and select '**New Database**'.
   - Enter the database '**Name**': `confluence`.
   - Select the '**Owner**': `confuser`.
   - Click '**OK**' to create the database.

Alternatively, If you are on UNIX and do not have pgAdmin III, you can use the command line interface instead. Assuming that you are using the default installation directory of `/opt/PostgreSQL/8.3/bin/`, enter the following commands:

```
sudo -s -H -u postgres
# Create the Confluence user:
/opt/PostgreSQL/8.3/bin/createuser -S -d -r -P -E confuser
# Create the Confluence database:
/opt/PostgreSQL/8.3/bin/createdb -O confuser confluence
exit
```

Screenshot 1 (click to enlarge): Confluence database and user in PostgreSQL

Step 2. Install Confluence

Requirements: **Confluence 3.0.1**.

For Windows: (click to expand)

1. Go to the Atlassian download centre.
2. Download the '**Standalone (ZIP Archive)**' file for **Confluence 3.0.1**.
3. Unpack the zip archive into a directory of your choice, avoiding spaces in the directory name.
4. Tell Confluence where to put its Confluence Home directory:
   - Edit the properties file at {`CONFLUENCE_INSTALL}\confluence\WEB-INF\classes\confluence-init.properties`.
   - Remove the hash sign (#) in front of the following line, and enter the directory name:
     `# confluence.home=c:/confluence/data`
     For example:
     `confluence.home=c:/data/confluence-home`
     (Note the forward slashes.)
   - Save the file.
5. Because Confluence will be running on the same machine as JIRA (already installed), you need to ensure that the application server ports for Confluence and JIRA are different. By default, both applications use port 8080. Change the default Confluence port as follows:
   - Edit the configuration file at {`CONFLUENCE_INSTALL}\conf\server.xml`.
   - Change the value of the `port` attribute in the `Connector` element to `8090`.
6. Copy the PostgreSQL JDBC driver JAR (downloaded in Dragons Stage 1) to your {`CONFLUENCE_INSTALL}\confluence\WEB-INF\lib` directory.
7. Start your Confluence server by running {`CONFLUENCE_INSTALL}\bin\startup.bat`.

For UNIX or Linux: (click to expand)

1. Go to the Atlassian download centre.
2. Click the 'Linux' tab and download the '**Standalone (TAR.GZ Archive)**' file for **Confluence 3.0.1**.
3. Unpack the tar.gz archive into a directory of your choice, avoiding spaces in the directory name.
4. Tell Confluence where to put its Confluence Home directory:
   - Edit the properties file at {`CONFLUENCE_INSTALL}/confluence/WEB-INF/classes/confluence-init.properties`.
   - Remove the hash sign (#) in front of the following line, and enter the directory name:
     `# confluence.home=c:/confluence/data`
     For example:
     `confluence.home=/var/confluence-home`
   - Save the file.
5. Because Confluence will be running on the same machine as JIRA (already installed), you need to ensure that the application server ports for Confluence and JIRA are different. By default, both applications use port 8080. Change the default Confluence port as follows:
   - Edit the configuration file at {`CONFLUENCE_INSTALL}/conf/server.xml`.
   - Change the value of the `port` attribute in the `Connector` element to `8090`.
6. Copy the PostgreSQL JDBC driver JAR (downloaded in Dragons Stage 1) to your {`CONFLUENCE_INSTALL}/confluence/WEB-INF/lib` directory.
7. Start your Confluence server by running {`CONFLUENCE_INSTALL}/bin/startup.sh`.

Problems? Please go immediately to the Dragon Slayers' Forum.
Victory? Please continue.

Step 3. Set Up Confluence

Now you can run Confluence's Setup Wizard and change some configuration settings.

1. To access Confluence, go to your web browser and type this address: http://localhost:8090.
2. The Confluence Setup Wizard will start up, to guide you through the process of setting up your Confluence server and creating an administration user. Detailed instructions are in the Confluence documentation.
3. Enter your Confluence license into the 'License Key' field. If you do not already have a Confluence license, follow the prompts on the Setup Wizard screen to get an evaluation license key.
4. Connect Confluence to your PostgreSQL database:
    - Click the '**Custom Installation**' button.
    - The 'Choose a Database Configuration' screen will appear. In the 'External Database' section, ensure that 'PostgreSQL' is selected and click the '**External Database**' button.
    - The 'Configure Database' screen will appear. Click the '**Direct JDBC**' button in the 'Direct JDBC Connection' section.
    - In the 'Setup Standard Database' section, enter the following information:
        - Driver Class Name: `org.postgresql.Driver` – This is the default value.
        - Database URL: `jdbc:postgresql//localhost:5432/confluence`
        - Username: `confuser` – This is the user you created in step 1 (above).
        - Password – This is the password you chose in step 1 (above).
    - Click the '**Next**' button.
5. # On the 'Load Content' screen, click the '**Example Site**' button to include the demonstration space content into your Confluence installation:.

    You might need to wait a few minutes while Confluence sets up its database and the demonstration space content.
6. The 'Setup System Administrator' screen will appear. Enter the following information:
    - Username: **`charlie`**
    - Password – Enter a password for the administrator account and enter it again in the 'Confirm' field to confirm it.
    - Name: **`Charlie of Atlassian`**
    - Email – We recommend that you give your own email address here.
7. Click the '**Next**' button.
8. The 'Confluence Setup Successful' screen will appear. Click '**Start using Confluence now**'.
9. The 'Confluence Demonstration Space' home page will appear.
10. Finally, you need to change your Confluence Server Base URL to the full (website) address at which Confluence is running, not just 'localhost':
    - Open the '**Browse**' menu at the top of the screen and select '**Confluence Admin**'.
    - The 'Administration Console' screen will appear. Click '**General Configuration**' under 'Configuration' in the left-hand panel.
    - The 'General Configuration' screen will appear. Click any of the '**Edit**' buttons.
    - In the '**Server Base Url**' field of the 'Site Configuration' section, enter the full website address at which Confluence is running, as it should not be 'localhost'. For example, if your computer name is 'coopers' then the server base URL should be: `http://coopers:8090`. Alternatively, specify a website address such as `http://www.foobar.com:8090`.
    - Scroll down to the end of the page and click the '**Save**' button.
11. Log out of Confluence, but leave the Confluence server running. (Move your cursor over the name '**Charlie of Atlassian**' and click '**Log Out**'.)

Screenshot 2 (click to enlarge): Home page of the Confluence demo space

Problems? Please go immediately to the Dragon Slayers' Forum.
Victory? Please continue.

Step 4. Hook Confluence up to Crowd

Follow the steps below to hook Confluence up to Crowd for SSO and centralised user management.

1. If Crowd is not already running, start it up by running {CROWD_INSTALL}/start_crowd.bat and go to your Crowd URL in your browser, e.g. http://www.foobar.com:8095/crowd.
2. Log in to Crowd with username **charlie**.
3. Click '**Applications**' in the top navigation bar.
4. The 'Application Browser' will appear. Click '**Add Application**' in the left-hand menu.
5. This will display the first screen for the 'Add Application' wizard for Crowd. Enter the following information:
    - Application Type: Confluence
    - Name: confluence
    - Description: Atlassian Confluence
    - Password – Enter a password that Confluence will use to access Crowd and enter it again to confirm it.
    - URL – Enter the base URL of your Confluence site, as configured in step 3 above, e.g. http://www.foobar.com:8090.
    - Click '**Resolve IP Address**' to ask Crowd to find the '**Remote IP Address**' for you. The value will be something like this: 127.0.0.1.
    - Select the '**Crowd**' directory that you created in Dragons Stage 1.
    - Select '**Allow all users to authenticate**'.
    - Click '**Add Application**'.
6. Check the IP addresses for your Confluence application:
    - Click the '**Remote Addresses**' tab.
    - Add your Confluence host name, e.g. http://www.foobar.com:8090.
    - If it's not already present, add: 127.0.0.1.
7. Leave Crowd up and running, but shut down Confluence. (Press Ctrl+C in your Confluence server command window or run {CONFLUENCE_INSTALL}\bin\shutdown.bat (on Windows) or {CONFLUENCE_INSTALL}/bin/shutdown.sh (on UNIX).)
8. Copy the Crowd client libraries and configuration files to your Confluence installation folder:
    - Copy {CROWD_INSTALL}/client/crowd-integration-client-2.0.1.jar to {CONFLUENCE_INSTALL}/confluence/WEB-INF/lib
    - Copy {CROWD_INSTALL}/client/conf/crowd.properties to {CONFLUENCE_INSTALL}/confluence/WEB-INF/classes
    - Copy {CROWD_INSTALL}/client/conf/crowd-ehcache.xml to {CONFLUENCE_INSTALL}/confluence/WEB-INF/classes
9. Edit the {CONFLUENCE_INSTALL}/confluence/WEB-INF/classes/crowd.properties file and change the following properties:
    - application.name: confluence
    - application.password – Enter the password that Confluence will use to access Crowd. This must be the same password as you entered in the Crowd 'Add Application' wizard above.
10. Edit the {CONFLUENCE_INSTALL}/confluence/WEB-INF/classes/atlassian-user.xml file. Uncomment the Crowd provider and comment the default repository so that the contents of the file is:

```
<atlassian-user>
   <repositories>
      <crowd key="crowd" name="Crowd Repository"/>
   </repositories>
</atlassian-user>
```

11. Edit the {CONFLUENCE_INSTALL}/confluence/WEB-INF/classes/seraph-config.xml file. Comment out the 'authenticator' element:
    <authenticator class="com.atlassian.confluence.user.ConfluenceAuthenticator"/>
    and replace it with:
    <authenticator class="com.atlassian.crowd.integration.seraph.ConfluenceAuthenticator"/>
    Your modifications should look similar to this:

```
<security-config>
   ...
   <!-- <authenticator
      class="com.atlassian.confluence.user.ConfluenceAuthenticator"/> -->
   <authenticator class=
      "com.atlassian.crowd.integration.seraph.ConfluenceAuthenticator"/>
   ...
</security-config>
```

12. Start your Confluence server again, and go to your Confluence URL in your browser, e.g. http://www.foobar.com:8090.
13. Log in to Confluence with username **charlie** and Charlie's password in Crowd.

    You are now authenticating via Crowd!

Full details are in the Crowd documentation.

Problems? Please go immediately to the Dragon Slayers' Forum.
Victory? Please continue.

Step 5. Configure JIRA to Trust Confluence

In this step you will set up a trusted communication channel between your JIRA and Confluence sites, so that you can display JIRA information on a Confluence wiki page.

1. Keep Confluence open in your browser, and open another browser window/tab. Go to your JIRA site in the second window/tab.
   Because you are using Crowd for single sign-on, you should be automatically logged in to JIRA with username `charlie`.
2. Click '**Administration**' in the top navigation bar.
3. The JIRA Administration console will open. Click '**Trusted Applications**' in the left-hand panel (in the 'System' section).
4. The 'View Trusted Applications' screen will appear, with a section called '**Request New Trusted Application Details**'. Copy the base URL for your Confluence site (e.g. `http://coopers:8090` or `http://www.foobar.com:8090`) and paste it into the '**Base URL**' field.
5. Click '**Send Request**'.
6. The 'Add New Trusted Application' screen will appear. Enter the following information:
   - Application Name: `Confluence` – The default will be the URL you entered on the previous screen. You can safely change it to a more meaningful name.
   - IP Address Matches: `127.0.0.1` – Add this address to a new line in the box.
7. Leave the other fields at their default values.
8. Click '**Add**'.

Full details are in the JIRA documentation.

Problems? Please go immediately to the Dragon Slayers' Forum.
Victory? Please continue.

Step 6. Create a Wiki Space

Now you can create a space in Confluence for use in the subsequent stages of the Atlassian Dragon Quest. A 'space' is a logical collection of pages, comparable to a library. A space is configurable and managed independently within a wiki site. It's almost like a wiki within a wiki.

The Atlassian Confluence demonstration space was created for you when you set up Confluence above.

1. Click '**Dashboard**' at the top left of the Confluence screen.
2. Click '**Create a space**' on the left side of the screen.
3. The 'Create Space' screen will appear. Configure your space settings:
    - Enter a space name: **`Dragons`**
    - Enter a space key: **`DRA`**
    - Who can use this space? – Leave the default settings as they are.
    - Choose Theme – Leave the default settings as they are (that is, '**Default Theme**').
    - Click '**OK**'.
4. The 'Home' page of your new 'Dragons' space will appear, with some default content. Now you can edit the home page as you like. For this exercise, add a **Charlie badge**:
    - Right-click on the image of the **Charlie badge** at the bottom of this documentation page and save it to your desktop. The file name is '`dragon_badge04.png`'.
    - Click '**Edit**' at the top right of your new Dragons home page in your own Confluence site.
    - The wiki rich text editor will open. Select and delete the following text in the editor pane:

        > *This is the home of the Dragons space.*
        > *To help you on your way, we've inserted some of our favourite macros on this home page. As you start creating pages, adding news items and commenting you'll see the macros below fill up with all the activity in your space.*

    - Press the 'Enter' key twice to make some space.
    - Make sure your cursor is at the top of the editor pane.
    - Click the '**Insert/Edit Image**' icon in the editor toolbar.
    - The 'Insert Image' popup window will appear. Browse to your desktop and select the **Charlie badge** image that you saved earlier. Click '**Attach**' to upload the image.
    - The image file name will appear in the 'Filename' textbox. Click '**OK**'.
    - The image will appear in the editor pane of your home page. Click '**Save**' to save your updated wiki page.
5. Now you will add a JIRA Issues macro to your page, to display a dynamic list of issues drawn from your '**Dragons**' project on your JIRA site. The first step is to define a filter in JIRA:
    - Keep Confluence open in your browser, and open another browser window/tab. Go to your JIRA site in the second window/tab.

        Because you are using Crowd for single sign-on, you should be automatically logged in to JIRA with username **`charlie`**.
    - Click the down arrow next to '**Issues**' in the top navigation bar, then select '**Search for Issues**'.
    - The 'Issue Navigator' will appear. Select '**Dragons**' in the '**Project**' list on the left.
    - Click '**View**'.
    - A list of issues will appear in the 'Issue Navigator'. You should see your two issues, **`DRA-1`** and **`DRA-2`**. Click '**Save it as a filter**' in the left-hand panel.
    - The 'Save Current Filter*' screen will appear. Enter the following information:
        - Name: `Dragons`
        - Description: `Dragons`
    - Click the '**Save**' button at the bottom of the screen (*not* the one next to 'Add Shares').
    - The saved filter will appear, showing the same two issues. Click '**Views**' at top right of the screen, right-click '**XML**' and copy the link location for the '*XML' view into your clipboard.
6. Add the JIRA Issues macro to your Confluence page:
    - Go back to your Confluence browser window/tab.
    - Edit your 'Dragons' home page again.
    - Place your cursor immediately after your **Charlie badge** image and press '**Enter**' a few times to make some blank lines.
    - Paste the content of your clipboard into a blank line, for safe keeping. You will delete it again soon. It should be a JIRA filter URL that looks something like this:

        ```
        http://coopers
           :8080/sr/jira.issueviews:searchrequest-xml/10000/SearchRequest-10000.xml?tempMax=1000
        ```

    - Copy the following text into the next line on the Confluence page:

        ```
        {jiraissues:url=CONTENT}
        ```

7. Replace '`CONTENT`' with the JIRA filter URL from the line above, then delete the line containing the filter URL.
    - Save the page.

Victory!

You now have a 'Dragons' space in your Confluence wiki. Your space home page has a Charlie badge and a dynamically-updated list of JIRA issues.

Screenshot 3 (click to enlarge): Home page of the Dragons space

Problems? Please go immediately to the Dragon Slayers' Forum.
Victory? Please continue.

Grab your Shield and Move to the Next Stage

- **Tweet?** Tweet.
- Go to Dragons Stage 5 - Install FishEye.

Dragons Stage 5 - Install FishEye

You are embarking on stage 5 of the Atlassian Dragon Quest, a place filled with flame and serpents and dragons.

In this stage, you will install FishEye for breathtaking overviews of your source code repository. Prepare to be blown away by FishEye's integration with JIRA, Crowd and Bamboo.

Time estimate: This stage will take approximately **30 minutes**.

On this page:

- Step 1. Create your FishEye Database in PostgreSQL
- Step 2. Install FishEye
- Step 3. Set Up FishEye and Connect to PostgreSQL Database
- Step 4. Connect FishEye to Subversion
- Step 5. Hook FishEye up to Crowd
- Victory!

Step 1. Create your FishEye Database in PostgreSQL

Now you will create a database where FishEye will store its data, and the user that FishEye will use to connect to the database. We are assuming that you have already created your PostgreSQL database server in Dragons Stage 1.

We are using pgAdmin III, the administration user interface supplied with PostgreSQL. If you used the one-click installer in Dragons Stage 1, pgAdmin III will be already installed on your computer.

1. Start **pgAdmin III**.
2. Add a new login role called 'fishuser':
   - Right-click '**Login Roles**' and select '**New Login Role**'.
   - Enter the role '**Role name**': `fishuser`.
   - Enter a suitable '**Password**' and enter it again to confirm it.
   - Select '**Can create database objects**'.
   - Select '**Can create roles**'.
   - Click '**OK**' to create the user.
3. Add a new database called 'fisheye':
   - Right-click '**Databases**' and select '**New Database**'.
   - Enter the database '**Name**': `fisheye`.
   - Select the '**Owner**': `fishuser`.
   - Click '**OK**' to create the database.

Alternatively, If you are on UNIX and do not have pgAdmin III, you can use the command line interface instead. Assuming that you are using the default installation directory of `/opt/PostgreSQL/8.3/bin/`, enter the following commands:

```
sudo -s -H -u postgres
# Create the FishEye user:
/opt/PostgreSQL/8.3/bin/createuser -S -d -r -P -E fishuser
# Create the FishEye database:
/opt/PostgreSQL/8.3/bin/createdb -O fishuser fisheye
exit
```

Screenshot 1 (click to enlarge): FishEye database and user in PostgreSQL

Problems? Please go immediately to the Dragon Slayers' Forum.
Victory? Please continue.

Step 2. Install FishEye

Requirements: **FishEye 2.0.5**.

For Windows: (click to expand)

1. Go to the Atlassian download centre.
2. Download the '**FishEye 2.0.5**' zip archive.
3. Unpack the zip archive into a directory of your choice, avoiding spaces in the directory name. For example: `c:\fisheye`. We will now refer to this location as the FishEye installation directory.
4. Now you will create another directory where FishEye will store its local data, separate from the installation directory:
    - Create the new directory, e.g. `C:\data\fisheye`.
    - Create an environment variable called '`FISHEYE_INST`' and point it to your new directory. (Open your Windows '**Control Panel**'. Click '**System**' to open the 'System Properties'. Click the '**Advanced**' tab. Click '**Environment Variables**'. Add a new '**System variable**' with the name '`FISHEYE_INST`' and a value of your new directory's location of e.g. `C:\data\fisheye`.)
    - Copy the `config.xml` file from the root of your FishEye installation directory to the root of your new `FISHEYE_INST` directory.
5. Now you will make your PostgreSQL driver available to FishEye:
    - Create a `\lib` directory as a sub-directory of your new `FISHEYE_INST` directory
    - Copy the PostgreSQL JDBC driver JAR (downloaded in Dragons Stage 1) to the new `\lib` directory.
6. Start FishEye from the command line by running `bin\run.bat` from your FishEye installation directory.
    - Wait a few minutes for the server to launch. This message will appear on the command line once ready: '**INFO - Server started on :8060 (http) (control port on *your-server-IP-address*:8059)**'.

For Linux: (click to expand)

1. Go to the Atlassian download centre.
2. Download the '**FishEye 2.0.5**' zip archive.
3. Unpack the zip archive into a directory of your choice, avoiding spaces in the directory name. For example: `/opt/fecru-x.x.x`. We will now refer to this location as the FishEye installation directory.
4. Now you will create another directory where FishEye will store its local data, separate from the installation directory:
    - Create the new directory, e.g. `/opt/fisheye`.
    - Create an environment variable called '`FISHEYE_INST`' and point it to your new directory.
    - Copy the `config.xml` file from the root of your FishEye installation directory to the root of your new `FISHEYE_INST` directory.
5. Now you will make your PostgreSQL driver available to FishEye:
    - Create a `/lib` directory as a sub-directory of your new `FISHEYE_INST` directory.
    - Copy the PostgreSQL JDBC driver JAR (downloaded in Dragons Stage 1) to the new `/lib` directory.
6. Start FishEye from the command line by running `./bin/run.sh` from your FishEye installation directory.
    - Wait a few minutes for the server to launch. This message will appear on the command line once ready: '**INFO - Server started on :8060 (http) (control port on *your-server-IP-address*:8059)**'.

Problems? Please go immediately to the Dragon Slayers' Forum.
Victory? Please continue.

Step 3. Set Up FishEye and Connect to PostgreSQL Database

1. To access FishEye, go to your web browser and type this address: `http://localhost:8060/` (or type the host name or IP address instead of `localhost`).
2. Choose an administration password, enter it and then enter it again to confirm it.
3. Enter your FishEye license into the '**FishEye License Key**' field. If you do not already have a FishEye license, follow the prompts on the FishEye/Crucible Setup screen to get an evaluation license key.

   If you have a Crucible license as well as a FishEye license, you could enter the Crucible license on this screen too. But for the purposes of this integration exercise, we will assume you do not want to enable Crucible at this stage.
4. Click '**Save**'.
5. Click '**Admin Interface**' to go to the FishEye administration screens.
6. The 'Repository List' screen appears. Click '**Database Configuration**' in the left-hand panel.
7. Select '**PostgreSQL**' from the '**Type**' dropdown list.
8. Enter the following details:
   - URL: `jdbc:postgresql://localhost:5432/fisheye`
   - User Name: `fishuser` – This is the user you created in step 1 (above).
   - Password – This is the password you chose in step 1 (above).
9. Click '**Test Connection**' to verify that FishEye can log in to the database. If this fails, verify that you have the PostgreSQL JDBC driver JAR file in the `FISHEYE_INST/lib` directory (see step 2 above). Note that this is *not* your installation directory. Also ensure that the database user can log in to the database from the machine that FishEye is running on and that all the required privileges are present. Hint: If you have a virus checker running, there may be a delay in the driver's availability after you have placed the driver JAR into the directory, while the virus checker scans the file. It's worth waiting a while and trying again.
10. Click '**Save & Migrate Data**'.

Screenshot 2 (click to enlarge): FishEye database migration successful

Problems? Please go immediately to the Dragon Slayers' Forum.
Victory? Please continue.

Step 4. Connect FishEye to Subversion

For the purposes of this integration exercise, we have provided a read-only Subversion repository that you can connect to your FishEye 'Dragons' repository. We recommend this repository because:

- We have committed a code change with a JIRA issue key in the commit message, to match a JIRA issue you created earlier. This will allow you to see the JIRA and FishEye integration immediately, without having to do your own commit.
- The sample repository is small, so that FishEye's initial repository indexing process will be fast.

FishEye supports Subversion and a number of other repository types. When you start using FishEye outside this integration exercise, you will need to create another FishEye repository and connect it to your source repository as described in the FishEye documentation. For this integration exercise, follow the steps below to connect to our sample repository.

1. Click the '**(new)**' part of '**Repository List (new)**' in the left-hand panel of the FishEye Admin interface.
2. The 'Add Repository' screen will appear. Enter the following information:
    - Name: `Dragons`.
    - Description: `Dragons repository`.
    - Repository type: `Subversion`.
    - Enable immediately: `Yes`.
    - SVN URL: `http://dragons.atlassian.com/dragons`.
    - Path – Not required for our sample repository.
    - Username and Password – Not required for our sample repository, because the repository allows anonymous access.
3. Click '**Add**'.
4. The 'View Repository' screen will appear. Click '**Test Connection**' to verify that Subversion is properly connected to FishEye.
5. Click the FishEye logo to return to the FishEye dashboard. You should be able to see the activity stream showing recent commit messages for the repository.

   If you do not see any activity, please wait a while for FishEye to finish scanning (indexing) the repository. With our sample SVN repository, this should only take a few minutes.

Screenshot 3 (click to enlarge): FishEye dashboard with activity stream

Full details are in the FishEye documentation.

Problems? Please go immediately to the Dragon Slayers' Forum.
Victory? Please continue.

Step 5. Hook FishEye up to Crowd

Follow the steps below to hook FishEye up to Crowd for SSO (single sign-on) and centralised user management.

1. If Crowd is not already running, start it up by running `{CROWD_INSTALL}/start_crowd.bat`. Open up a new browser window/tab and go to your Crowd URL, e.g. `http://www.foobar.com:8095/crowd`.
2. If not already logged in, log in to Crowd with username **`charlie`**.
3. Click '**Applications**' in Crowd's top navigation bar.
4. The 'Application Browser' will appear. Click '**Add Application**' in the left-hand menu.
5. This will display the first screen for the 'Add Application' wizard for Crowd. Enter the following information:
   - Application Type: `FishEye`
   - Name: `fisheye`
   - Description: `Atlassian FishEye`
   - Password – Enter a password that FishEye will use to access Crowd and enter it again to confirm it.
   - URL – Enter the base URL of your FishEye site, e.g. `http://fisheye.foobar.com:8060`.
   - Click '**Resolve IP Address**' to ask Crowd to find the '**Remote IP Address**' for you. The value will be something like this: `127.0.0.1`.
   - Select the '**`Crowd`**' directory that you created in Dragons Stage 1.
   - Select '**Allow all users to authenticate**'.
   - Click '**Add Application**'.
6. Check the IP addresses for your FishEye application:
   - Click the '**Remote Addresses**' tab.
   - Add your FishEye host name, e.g. `http://fisheye.foobar.com:8060`.
   - If it's not already present, add: `127.0.0.1`.
7. Go back to your FishEye browser window/tab.
8. Click '**Administration**' at the bottom of the FishEye screen, to go to the FishEye Admin screens.
9. Click '**Security**' in the left-hand panel.
10. The 'Authentication Settings' screen will appear. Select '**Setup Crowd authentication**'.
11. The '**Crowd Authentication Settings**' screen will appear. Enter the following information:
    - Application name: `fisheye`.
    - Application password – Enter the password you specified in Crowd's 'Add Application' wizard, as described above.
    - Leave the other fields at their default values.
12. Click '**Apply**'.

Full details are in the Crowd documentation.

Problems? Please go immediately to the Dragon Slayers' Forum.
Victory? Please continue.

Victory!

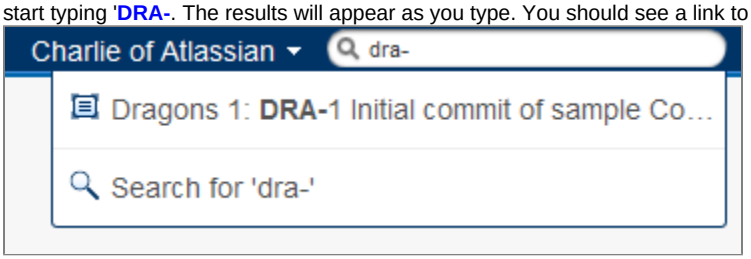
You can now see your source in FishEye. Go to the FishEye dashboard, click the '**Source**' tab and click '**`Dragons`**' to browse the contents of your new 'Dragons' repository.

Hint: Mark the **Dragons** repository as **Charlie's** favourite, so that Charlie will see all the repository activity on his FishEye activity streams. How? Make sure you are logged in as '**`charlie`**', click '**Source**' in the top navigation bar, then click the star next to the '**`Dragons`**' repository.

If your repository is large, FishEye may take a while to index all your files. If the index scanning is still underway, you will see a message at the top of the screen saying '`NOTE: The repository is being scanned, some statistics may not be up to date. ...`

Want an RSS feed of your repository activity? Go to the '**Activity**' tab on the Dashboard or on the 'Source' view. Click '**Tools**' then '**RSS**'.

Try FishEye's 'Quick Nav' feature for a rapid search of your repository contents. Go to the '**Search**' box at top right of the FishEye screen and start typing '**DRA-**. The results will appear as you type. You should see a link to the initial commit in the repository, like this:



Click through from the search results (shown above) or via FishEye's '**Source**' tab to see FishEye's view of your source code.

Screenshot 4 (click to enlarge): FishEye source repository viewer

Problems? Please go immediately to the Dragon Slayers' Forum.
Victory? Please continue.

Don your Armour and Move to the Next Stage

- **Tweet?** Tweet.
- Go to Dragons Stage 6 - Get JIRA and FishEye Talking.

Dragons Stage 6 - Get JIRA and FishEye Talking

You are embarking on stage 6 of the Atlassian Dragon Quest. Be prepared to ride on the dragon's back, for he is swift and strong and will take you where you need to go.

In this stage you will configure JIRA and FishEye, so that you will be able to see code commits in JIRA and see JIRA issues in FishEye.

Time estimate: This stage will take approximately **15 minutes**.

On this page:

- Step 1. Add a JIRA Server to FishEye and Enable Remote API
- Step 2. Configure the FishEye Plugin in JIRA
- Step 3. Add a FishEye Gadget to JIRA
- Victory!

Step 1. Add a JIRA Server to FishEye and Enable Remote API

In this step you will define a JIRA server in the FishEye administration interface, configure FishEye to trust JIRA and enable FishEye's remote API.

1. Go to your FishEye URL in your web browser, e.g. `http://localhost:8060/`.
2. Click '**Administration**' at the bottom of the FishEye screen, to go to the FishEye Admin screens.
3. Enter your FishEye administration password. This is the password you chose when setting up FishEye in Dragons stage 5.
4. Click '**JIRA Servers**' in the left-hand panel.
5. The 'View JIRA Servers' screen will appear. Click '**Add a JIRA Server**'.
6. The '**Add JIRA Server**' screen will appear. Enter the following information:
    - Name: **`Dragons JIRA`**.
    - URL – Enter the base URL of your JIRA server, e.g. `http://coopers:8080` or `http://www.foobar.com:8080`.
    - Username: **`charlie`**.
    - Password – Enter Charlie's password in Crowd.
    - Include in Activity Streams – Select this checkbox.
    - Authenticate as Trusted Application – Select this checkbox.
7. Click '**Test**' to ensure that your details are correct.
8. Click '**Save**'.
9. Click '**Trusted Applications**' in the left-hand panel.
10. The 'Trusted Applications List' will appear. Click '**Add a Trusted Application**'.
11. The 'Trusted Application' screen will appear. Enter the following information:
    - URL – Enter the base URL of your JIRA server, e.g. `http://coopers:8080` or `http://www.foobar.com:8080`.
12. Leave the other fields at their default values and click '**Get Id**'.
13. FishEye will add the trusted application ID into the '**Id**' field. Click '**Save**'.
14. Click '**Server Settings**' in the left-hand panel.
15. The 'Server Settings' screen will appear. Click '**Edit Settings**'.
16. The 'Edit Web Settings' screen will appear. Select the '**On**' radio button next to '**Allow remote API calls**'.
17. Click '**Update**'.

Step 2. Configure the FishEye Plugin in JIRA

The FishEye plugin for JIRA is bundled as part of the JIRA package, so there is no need to install it. Now you will configure the plugin for your installation and configure JIRA to trust FishEye.

1. Go to your JIRA URL in your browser, e.g. `http://www.foobar.com:8080`.
2. Click '**Administration**' in the top navigation bar.
3. The JIRA Administration console will open. Click '**FishEye Configuration**' in the left-hand panel (in the 'Global Settings' section).
4. The 'JIRA FishEye Plugin' screen will appear. Click '**Setup FishEye**'.
5. The 'FishEye' screen will appear. Enter the following information:
    - FishEye URL – Enter the URL of your FishEye server, e.g. `http://coopers:8060` or `http://fisheye.foobar.com:8060`.
    - Wiki Rendering: `ON`
6. Leave all the other fields at their default values and click '**Update**'.
7. The '**Associate FishEye Repositories with JIRA Projects**' screen will appear. Enter project key '**DRA**' next to the 'Dragons' repository under '**Mapped Project Keys**'.
8. Click '**Update**'.
9. Click '**Trusted Applications**' in the left-hand panel (in the 'System' section).
10. The 'View Trusted Applications' screen will appear, with a section called '**Request New Trusted Application Details**'. Copy the base URL for your FishEye site (e.g. `http://coopers:8060` or `http://fisheye.foobar.com:8060`) and paste it into the '**Base URL**' field.
11. Click '**Send Request**'.
12. The 'Add New Trusted Application' screen will appear. Enter the following information:
    - Application Name: `FishEye` – The default will be the URL you entered on the previous screen. You can safely change it to a more meaningful name.
    - IP Address Matches: `127.0.0.1` – Add this address to a new line in the box.
13. Leave the other fields at their default values.
14. Click '**Add**'.

There is now a 'Source' tab on your JIRA issues. Open your '**DRA-1**' issue and click the new '**Source**' tab. The tab shows the changesets related to the issue, i.e. changesets where the JIRA issue key was included in the commit message.

Screenshot 1 (click to enlarge): Source tab on a JIRA issue

There is now also a 'Source' tab on your JIRA project. Open your '**Dragons**' project to see the new tab. (Click the dropdown arrow next to '**Projects**' in the top navigation bar, then click the '**Dragons (DRA)**' project.) The 'Source' tab shows the most recent changesets related to any issue in the project.

Screenshot 2 (click to enlarge): Source tab on a JIRA project

You can click through from JIRA to view a changeset or other repository views in FishEye. Try clicking the changeset number ('**1**') or the repository name ('**Dragons**') on the 'Source' tab in JIRA.

You can also click through from FishEye to JIRA, by clicking the issue key. Try it by clicking '**DRA-1**' in your FishEye view.

Full details are in the FishEye plugin documentation.

Problems? Please go immediately to the Dragon Slayers' Forum.
Victory? Please continue.

Step 3. Add a FishEye Gadget to JIRA

Now you will add the 'FishEye Charts' gadget to your **Dragon Development Dashboard**.

1. Click '**Dashboards**' at top left of your JIRA screen.
2. Your 'Dragon Development Dashboard' will appear. Click '**Add Gadget**'.
3. The 'Gadget Directory' will appear, showing a list of the available gadgets for your JIRA dashboard. Enter 'FishEye' into the search box at top right of the Gadget directory screen.
4. The list of gadgets will change, to show only the gadgets that match your search term. Find the '**FishEye Charts**' gadget and click '**Add it Now**'. The gadget will be highlighted for a short time and the button's wording will change to 'Adding', while JIRA adds the gadget to the dashboard.
5. Click '**Finished**' to go back to your dashboard.
6. Configure the '**FishEye Charts**' gadget:
    - Enter '**Dragons**' in the '**Repository**' field.
    - Click the dropdown arrow next to '**Refresh Interval**' and select '**Every 15 Minutes**'.
    - Click '**Save**'.
7. Re-arrange your dashboard by dragging the '**Projects**' gadget to the right and dropping it under the '**Assigned to Me**' gadget. Drag the '**Agile**' gadget to the bottom right too.
8. Choose a different colour for your '**FishEye Recent Changesets**' gadget:
    - Move your cursor pointer over the gadget and click the downward-pointing arrow at top right of the gadget frame.
    - Select the **orange** square in the row of colours.

Problems? Please go immediately to the Dragon Slayers' Forum.
Victory? Please continue.

Victory!

Your JIRA dashboard now has 4 gadgets:

- The 'FishEye Charts' gadget
- The 'Assigned to Me' gadget
- The 'Projects' gadget

- The GreenHopper 'Agile' gadget

Screenshot 3 (click to enlarge): JIRA dashboard with 4 gadgets

Problems? Please go immediately to the Dragon Slayers' Forum.
Victory? Please continue.

Grab a Bigger Sword and Move to the Next Stage

- **Tweet?** Tweet.
- Go to Dragons Stage 7 - Install Bamboo.

Dragons Stage 7 - Install Bamboo

You are embarking on stage 7 of the Atlassian Dragon Quest. The dragon may be growing in strength and power, but so are you.

In this stage, you will install Atlassian Bamboo for continuous integration. Then you will get Bamboo talking to JIRA and Crowd, and run your first Bamboo build.

Time estimate: This stage will take approximately **60 minutes**.

On this page:

- Step 1. Create your Bamboo Database in PostgreSQL
- Step 2. Install Bamboo
- Step 3. Set Up Bamboo
- Step 4. Hook Bamboo up to Crowd
- Step 5. Get Bamboo and JIRA Talking

- Step 6. Set up a Project and Run a Build
- Victory!

Step 1. Create your Bamboo Database in PostgreSQL

Now you will create a database where Bamboo will store its data, and the user that Bamboo will use to connect to the database. We are assuming that you have already created your PostgreSQL database server in Dragons Stage 1.

We are using pgAdmin III, the administration user interface supplied with PostgreSQL. If you used the one-click installer in Dragons Stage 1, pgAdmin III will be already installed on your computer.

1. Start **pgAdmin III**.
2. Add a new login role called 'bamuser':
    - Right-click '**Login Roles**' and select '**New Login Role**'.
    - Enter the role '**Role name**': `bamuser`.
    - Enter a '**Password**' and enter it again to confirm it.
    - Select '**Can create database objects**'.
    - Select '**Can create roles**'.
    - Click '**OK**' to create the user.
3. Add a new database called 'bamboo':
    - Right-click '**Databases**' and select '**New Database**'.
    - Enter the database '**Name**': `bamboo`.
    - Select the '**Owner**': `bamuser`.
    - Click '**OK**' to create the database.

Alternatively, If you are on UNIX and do not have pgAdmin III, you can use the command line interface instead. Assuming that you are using the default installation directory of `/opt/PostgreSQL/8.3/bin/`, enter the following commands:

```
sudo -s -H -u postgres
# Create the Bamboo user:
/opt/PostgreSQL/8.3/bin/createuser -S -d -r -P -E bamuser
# Create the Bamboo database:
/opt/PostgreSQL/8.3/bin/createdb -O bamuser bamboo
exit
```

Screenshot 1 (click to enlarge): Bamboo database and user in PostgreSQL

Step 2. Install Bamboo

Requirements: **Bamboo 2.4.0**.

For Windows: (click to expand)

1. Go to the Atlassian download centre.
2. Download the '**Standalone (Windows Installer)**' file for **Bamboo 2.4.0**.
3. Launch the Bamboo Windows installer (`atlassian-bamboo-2.4-standalone.exe`).
    - When prompted, enter the '**folder where you would like Bamboo to be installed**'. For example: `C:\Program Files\Bamboo` or `C:\atlassian\bamboo`.
      From this point onwards, we will refer to this installation directory as {{{BAMBOO_INSTALL}.
    - When prompted, tell Bamboo where to put its '**Bamboo home**' directory. For example: `C:\data\bamboo-home`.
4. Click '**Finish**' to close the setup window when the installer has finished.
5. Copy the PostgreSQL JDBC driver JAR (downloaded in Dragons Stage 1) to your {`BAMBOO_INSTALL}webapp\WEB-INF\lib` directory.
6. Install Bamboo as a Windows Service, so that it starts each time you start Windows: Click '**Start**', '**Programs**', '**Bamboo**, '**Install Service**'.
7. Start Bamboo from your Windows '**Start**' menu: Click '**Start**', '**Programs**', '**Bamboo**, '**Start Service**'.

For UNIX or Linux: (click to expand)

1. Go to the Atlassian download centre.
2. Click the 'Linux' tab and download the '**Standalone (TAR.GZ Archive)**' file for **Bamboo 2.4.0**.
3. Unpack the tar.gz archive into a directory of your choice, avoiding spaces in the directory name.
4. Tell Bamboo where to put its Bamboo Home directory:
    - Edit the properties file at {`BAMBOO_INSTALL}/bamboo/webapp/WEB-INF/classes/bamboo-init.properties`.
    - Insert the property '`bamboo.home`' with an absolute path to your Bamboo Home directory. For example: `bamboo.home=/var/bamboo-home`
    - Save the file.
5. Copy the PostgreSQL JDBC driver JAR (downloaded in Dragons Stage 1) to your {`BAMBOO_INSTALL}webapp/WEB-INF/lib` directory.
6. Start your Bamboo server by running {`BAMBOO_INSTALL}/bamboo.sh start`.

Problems? Please go immediately to the Dragon Slayers' Forum.
Victory? Please continue.

Step 3. Set Up Bamboo

Now you can run Bamboo's Setup Wizard and then check your default Bamboo capabilities.

1. To access Bamboo, go to your web browser and type this address: http://localhost:8085/.
2. The Bamboo Setup Wizard will start up, to guide you through the process of setting up your Bamboo server and creating an administration user. Detailed instructions are in the Bamboo documentation. Below are the things you need to know for our Dragon Quest. Enter the '**Standard Installation Settings**' as follows:
    - License Key – If you do not already have a Bamboo license, follow the prompts on the Setup Wizard screen to get an evaluation license key.
    - Configuration Directory – Leave this at the default value.
    - Build Data Directory – Leave this at the default value.
    - Build Working Directory – Leave this at the default value.
    - Broker URL – Check that the URL contains a full URL and not '`localhost`'. If necessary, replace `localhost` with the real host name or IP address of your Bamboo server. For example, if your computer name is 'coopers' then the broker URL should look like this: `tcp://coopers.sydney.atlassian.com:54663?wireFormat.maxInactivityDuration=300000`.
3. Enter the following information to connect to your Bamboo database created in step 1 above:
    - Database configuration: `External Database`.
    - Database: `PostgreSQL`.
    - Database Connection: `Direct JDBC connection`.
    - Driver Class Name: `org.postgresql.Driver`.
    - Database URL: `jdbc:postgresql://localhost:5432/bamboo`.
    - User Name: `bamuser`.
    - Password – Enter the password you specified in step 1 above.
    - Overwrite existing data – Leave this checkbox unselected.
4. For your '**Starting Data**', select '**Create new Bamboo home**'.
5. Set up your '**Administrator User Details**':
    - Username: **`charlie`**.
    - Password – Enter a password for the administrator account and enter it again to confirm it.
    - Full Name: **`Charlie of Atlassian`**.
    - Email – Enter the address of your administrator email account. We recommend that you give your own email address here.
6. Enter the following '**Bamboo Configuration**' information:
    - Name of Bamboo instance: `Atlassian Bamboo`.
    - Base URL – Enter the full website address at which your Bamboo server is running, not just 'localhost'. For example, if your computer name is 'coopers' then the base URL should be: `http://coopers:8085`. Or specify a website address, such as `http://www.foobar.com:8085`.
    - Apply gzip compression – Select this checkbox.
    - Accept remote API calls – Select this checkbox, so that Bamboo's remote API is enabled.
7. Click '**Complete Installation**'.
8. Log in to Bamboo with username **`charlie`** and the password you specified when creating the administrator user in the steps immediately above.

    You can now see the Bamboo home page.
9. Now you will check that your Bamboo configuration includes your default builder and JDK. Click '**Administration**' in the top navigation bar.
10. The 'Bamboo Administration' screen will appear. Click '**Builders**' in the left-hand menu.
11. The 'Builders' screen will appear. Look through the list, to see if your build tool is included for '**All local agents**'. You can use any of the build tools supported by Bamboo, such as Maven 1, Maven 2, Ant, PHPUnit and others. See the Bamboo documentation. For this integration exercise, we assume that you are using Maven 2. In that case, you should see a row in the table for '**Maven 2**', where the '**Agent**' column includes '**All local agents**'.
12. If your builder is not included, click '**Add builder to server capabilities**'. The 'Add Capability' panel will appear. Enter the following information then click '**Add**':
    - Capability Type: `Builder`.
    - Type: `Maven 2.0`.
    - Label: `Maven 2`.
    - Path – Enter the path to your Maven installation. This should be the same as the value that you have specified in your `M2_HOME` environment variable. For example: `C:\maven2.2\apache-maven-2.2.0` (Windows) or `/usr/local/apache-maven/apache-maven-2.2.1` (UNIX).
13. Check that your Bamboo configuration includes your JDK. Click '**JDKs**' in the left-hand menu.
14. The 'JDKs' screen will appear. Look through the list, to check that your JDK is included for '**All local agents**'. You will need Sun JDK 1.5 or higher. Note that the JRE alone is not enough. Stage 1 of these instructions will guide you through the installation process. For this integration exercise, we assume that you are using JDK 1.6. In that case, you should see a row in the table for '**JDK 1.6.x_xx**', where the '**Agent**' column includes '**All local agents**'.
15. If your JDK is not included, click '**Add JDK to server capabilities**'. The 'Add Capability' panel will appear. Enter the following information then click '**Add**':
    - Capability Type: `JDK`.
    - Label: `JDK 1.6`.
    - Java Home – Enter the path to your JDK installation. This should be the same as the value that you have specified in your `JAVA_HOME` environment variable. For example: `C:\Sun\SDK\jdk` (Windows) or `/opt/java/java_sdk1.6` (UNIX).

Screenshot 2 (click to enlarge): Bamboo home page

Problems? Please go immediately to the Dragon Slayers' Forum.
Victory? Please continue.

Step 4. Hook Bamboo up to Crowd

Follow the steps below to hook Bamboo up to Crowd for SSO and centralised user management.

1. If Crowd is not already running, start it up by running {CROWD_INSTALL}/start_crowd.bat and go to your Crowd URL in your browser, e.g. http://www.foobar.com:8095/crowd.
2. Log in to Crowd with username **charlie**.
3. Click '**Applications**' in the top navigation bar.
4. The 'Application Browser' will appear. Click '**Add Application**' in the left-hand menu.
5. This will display the first screen for the 'Add Application' wizard for Crowd. Enter the following information:
   - Application Type: Bamboo.
   - Name: bamboo.
   - Description: Atlassian Bamboo.
   - Password – Enter a password that Bamboo will use to access Crowd and enter it again to confirm it.
   - URL – Enter the base URL of your Bamboo site, as configured in step 3 above, e.g. http://www.foobar.com:8085.
   - Click '**Resolve IP Address**' to ask Crowd to find the '**Remote IP Address**' for you. The value will be something like this: 127.0.0.1.
   - Select the '**Crowd**' directory that you created in Dragons Stage 1.
   - Select '**Allow all users to authenticate**'.
   - Click '**Add Application**'.
6. Check the IP addresses for your Bamboo application:
   - Click the '**Remote Addresses**' tab.
   - Add your Bamboo host name, e.g. http://www.foobar.com:8085.
   - If it's not already present, add: 127.0.0.1.
7. Leave Crowd up and running, but shut down Bamboo. (On Windows, open your '**Start**' menu and select '**Programs**', '**Bamboo**', '**Stop Service**'. On UNIX, run {BAMBOO_INSTALL}/bamboo.sh stop.)
8. Remove the following file from your Bamboo installation folder: {BAMBOO_INSTALL}/webapp/WEB-INF/lib/crowd-integration-client-1.6.1.jar
9. Copy the Crowd client libraries and configuration files to your Bamboo installation folder:
   - Copy {CROWD_INSTALL}/client/crowd-integration-client-2.0.1.jar to {BAMBOO_INSTALL}/webapp/WEB-INF/lib
   - Copy {CROWD_INSTALL}/client/conf/crowd.properties to {BAMBOO_INSTALL}/webapp/WEB-INF/classes
   - Copy {CROWD_INSTALL}/client/conf/crowd-ehcache.xml to {BAMBOO_INSTALL}/webapp/WEB-INF/classes
10. Edit the {BAMBOO_INSTALL}/webapp/WEB-INF/classes/crowd.properties file and change the following properties:
    - application.name: bamboo
    - application.password – Enter the password that Bamboo will use to access Crowd. This must be the same password as you entered in the Crowd 'Add Application' wizard above.
11. Edit the {BAMBOO_INSTALL}/webapp/WEB-INF/classes/atlassian-user.xml file. Uncomment the Crowd provider and comment the default repository so that the contents of the file is:

```
<atlassian-user>
   <repositories>
      <crowd key="crowd" name="Crowd Repository"/>
   </repositories>
</atlassian-user>
```

12. Edit the {BAMBOO_INSTALL}/webapp/WEB-INF/classes/seraph-config.xml file. Comment out the authenticator node :

```
<!--<authenticator
   class="com.atlassian.bamboo.user.authentication.BambooAuthenticator"/>-->
```

    and add a new one:

```
<authenticator class="com.atlassian.crowd.integration.seraph.BambooAuthenticator"/>
```

13. Start your Bamboo server again, and go to your Bamboo URL in your browser, e.g. http://www.foobar.com:8085.
14. If you are still logged in to Crowd, you will be automatically logged in to Bamboo with username **charlie**. If not, log in using Charlie's password in Crowd.

    You are now authenticating and using single sign-on via Crowd!

Full details are in the Crowd documentation.

Problems? Please go immediately to the Dragon Slayers' Forum.
Victory? Please continue.

Step 5. Get Bamboo and JIRA Talking

In this step you will set up the integration between Bamboo and JIRA, so that you can see your build information in JIRA and your issues in Bamboo.

1. First you will tell your JIRA server about your Bamboo server. Keep Bamboo open in your browser, and open another browser window/tab. Go to your JIRA site in the second window/tab.
   Because you are using Crowd for single sign-on, you should be automatically logged in to JIRA with username **`charlie`**.
2. Click '**Administration**' in JIRA's top navigation bar.
3. The 'Projects' administration screen will appear. Click '**Bamboo Servers**' (in the 'Global Settings' section).
4. The 'Bamboo Servers' screen will appear. Click '**Add Bamboo server**'.
5. The 'Add Bamboo server' screen will appear. Enter the following information:
   - Server name: `Atlassian Bamboo`.
   - Description: `Atlassian Bamboo`.
   - Host URL – Enter the base URL for your Bamboo site, e.g. `http://coopers:8085` or `http://www.foobar.com:8085`.
   - User name: **`charlie`** – This is the user name that Bamboo will use to log in to JIRA.
   - Password – Enter Charlie's password as specified in Crowd.
   - Associated JIRA projects – Leave this field empty.
6. Click '**Add**'.
7. Now you will tell your Bamboo server about your JIRA server. Go back to your Bamboo window/tab in your browser.
8. Click '**Administration**' in Bamboo's top navigation bar.
9. The 'Bamboo Administration' screen will appear. Click '**JIRA Server**' in the left-hand menu (in the 'Communication' section).
10. The 'Add a JIRA Server' screen will appear. Enter the following information:
    - Host URL – Enter the base URL for your JIRA site, e.g. `http://coopers:8080` or `http://www.foobar.com:8080`.
    - Username: **`charlie`** – This is the user name that JIRA will use to log in to Bamboo.
11. Password – Enter Charlie's password, as specified in Crowd.
12. Issue Key: **`DRA-1`** – This is the JIRA issue key for the issue that you created in Dragons stage 2.
13. Click '**Test**'.
    - You should see the following message: '**Successfully retrieved JIRA issue from remote server**'. You should also see your issue key and summary under the heading '**Server Response**'.
    - If you do not see a successful response, check that you can log in to your JIRA server using the JIRA account and password you have specified on this screen.
14. Click '**Save**'.

Full details are in the Bamboo documentation.

Problems? Please go immediately to the Dragon Slayers' Forum.
Victory? Please continue.

Step 6. Set up a Project and Run a Build

In this step you will create a Bamboo project and run a sample build. For the purposes of this integration exercise, we have provided a read-only Subversion repository that you can connect to your Bamboo 'Dragons' plan. We have committed a code change with a JIRA issue key in the commit message, to match a JIRA issue you created earlier. This will allow you to see the JIRA, FishEye and Bamboo integration immediately, without having to do your own commit.

1. Click '**Create Plan**' in Bamboo's top navigation bar.
2. The 'Create a new plan' screen will appear. Enter the following information:
    - Project Name: `Dragons`.
    - Project Key: `DRAG`.
    - Build Plan Name: `Main`
    - Build Plan Key: `MAIN`
3. Click '**Next**' and enter the following information.
    - Repository: `Subversion`.
    - Repository URL: `http://dragons.atlassian.com/dragons`.
    - Username and Password – Not required for our sample repository, because the repository allows anonymous access.
    - Authentication Type – Leave this at the default value of '`Password`'.
    - Web Repository URL – Enter the URL of your 'Dragons' source project in FishEye, e.g. `http://fisheye.foobar.com:8060/browse/Dragons`.
4. Leave the rest of the fields at their default values and click '**Next**'.
5. Enter the following information:
    - Builder – Select your build tool, e.g. `Maven 2`.
    - JDK – Select your JDK version, e.g. `JDK 1.6`.
6. Leave the rest of the fields at their default values and click '**Next**'.
7. Leave all the fields at their default values and simply click '**Next**' on each of the following screens: 'Requirements', 'Artifacts', 'Notifications' and 'Post Actions'.
8. Leave all the fields at their default values on the 'Permissions' screen and click '**Save**'.
9. Bamboo will immediately start a build, based on the plan that you have just created.
10. When the build has finished, you will see a build result for build '**DRAG-MAIN-1**'. This is the build '**1**' based on your new plan '**MAIN**' in your project '**DRAG**'. With any luck, the build should be successful.

Screenshot 3 (click to enlarge): Bamboo build in progress

Screenshot 4 (click to enlarge): Bamboo build completed

Click the '**Summary**' tab to see the plan summary.

Screenshot 5 (click to enlarge): Bamboo plan summary

Full details on creating a plan are in the Bamboo documentation.

Problems? Please go immediately to the Dragon Slayers' Forum.
Victory? Please continue.

Victory!

Your Bamboo, FishEye and JIRA servers are fully integrated. Here are some of the highlights for you to try.

You can link your builds to JIRA issues in various ways. For example, you can include a JIRA issue key in a commit comment. Details are in the Bamboo documentation. To see the integration happening right now, add a comment to your build:

- In Bamboo, click '**DRAG-MAIN-1**' to open the build result summary.
- Click '**Actions**' then '**Add comment**'.
- Add the following comment: `This build is related to DRA-1`.
- Click '**Save**'.
- Click the '**Summary**' tab to go back to the build result summary.

Notice the panel showing the JIRA issue details on the Bamboo build result screen. The issue key is hyperlinked so that you can open the issue in JIRA.

Screenshot 6 (click to enlarge): Bamboo build result with links to JIRA issue

Click the '**Issues**' tab to see the JIRA issues for a build result.

Screenshot 7 (click to enlarge): Bamboo build result showing a JIRA issues tab

Go to **JIRA** to see the Bamboo builds that relate to a particular JIRA issue, project or version. Details are in the JIRA documentation about viewing the Bamboo builds relating to a JIRA issue, project or version. The screenshot below shows the build for a particular issue.

Screenshot 8 (click to enlarge): JIRA issue showing a Bamboo build tab

When you link your FishEye and Bamboo projects to your own source repository and then commit changes, a source link will appear on your Bamboo build result. You will be able to click the source link to view the changed code in FishEye.

Unfortunately, you cannot reproduce this now because our sample repository is read-only. The screenshot below is for information only.

Screenshot 9 (click to enlarge): Bamboo build result showing link to FishEye source view

Problems? Please go immediately to the Dragon Slayers' Forum.
Victory? Please continue.

Grab a Bigger Shield and Go Conquer that Dragon

- **Tweet?** Tweet.
- Go to Dragons Stage 8 - Bamboo Gadgets and JIRA Victory.

Dragons Stage 8 - Bamboo Gadgets and JIRA Victory

You're nearly there. Stage 8 is the final step in the Atlassian Dragon Quest. The dragon is a softy!

In this stage, you will add the 'Bamboo Plans' and 'Plan Summary' gadgets to your JIRA dashboard.

Time estimate: This stage will take approximately **15 minutes**.

On this page:

- Step 1. Add JIRA as an OAuth Consumer in Bamboo
- Step 2. Add 2 Bamboo Gadgets to JIRA
- The Battle is Won, the Dragon is Slain

Step 1. Add JIRA as an OAuth Consumer in Bamboo

Some gadgets require you to set up an OAuth communication channel between the site where the information is coming from (e.g. Bamboo) and the site where the information will be displayed (e.g. your JIRA dashboard). The 'Bamboo Plans' and 'Plan Summary' gadgets do require this setup. You will need to configure Bamboo to allow your JIRA site as an OAuth consumer.

1. Go to your Bamboo URL in your browser, e.g. `http://www.foobar.com:8085`.
2. Click '**Administration**' in Bamboo's top navigation bar.
3. The 'Bamboo Administration' screen will appear. Click '**OAuth Consumers**' in the left-hand panel.
4. The 'OAuth Administration' screen will appear. Click '**Add OAuth Consumer**'.
5. Enter the base URL of your JIRA site into the field labelled '**Consumer Base URL**', e.g.`http://coopers:8080` or `http://www.foobar.com:8080`.
6. Click '**Add**'.

Problems? Please go immediately to the Dragon Slayers' Forum.
Victory? Please continue.

Step 2. Add 2 Bamboo Gadgets to JIRA

Now you will add the 'Bamboo Plans' and 'Plan Summary' gadgets to your JIRA **Dragon Development Dashboard**.

1. Go to your JIRA URL in your browser, e.g. `http://www.foobar.com:8080`.
2. Click '**Dashboards**' at top left of your JIRA screen.
3. Your 'Dragon Development Dashboard' will appear. Click '**Add Gadget**'.
4. The 'Gadget Directory' will appear, showing a list of the available gadgets for your JIRA dashboard. Enter 'Bamboo' into the search box at top right of the Gadget directory screen.
5. The list of gadgets will change, to show only the gadgets that match your search term. Find the '**Bamboo Plan Summary Chart**' gadget and click '**Add it Now**'. The gadget will be highlighted for a short time and the button's wording will change to 'Adding', while JIRA adds the gadget to the dashboard.
6. Find the '**Bamboo Plans**' gadget and add it too.
7. Click '**Finished**' to go back to your dashboard.
8. Configure the '**Bamboo Plans**' gadget:
    - Click '**Login & approve**'.
    - If prompted, log in to Bamboo as '**`charlie`**'. You will probably not be prompted, because you are currently logged in.
    - The 'Request for Access' screen will appear. Click '**Approve Access**'. This is how you, as the Bamboo user, allow your JIRA site to access your Bamboo data.
    - The 'Bamboo Plans' gadget on your JIRA dashboard will now display some configuration fields. Select '**Dragons**' in the '**Select Plans**' section.
    - Click the dropdown arrow next to '**Refresh Interval**' and select '**Every 15 Minutes**'.
    - Click '**Save**'.
9. Configure the '**Bamboo Plan Summary Chart**' gadget:
    - Click '**Login & approve**'.
    - The 'Request for Access' screen will appear. Click '**Approve Access**'.
    - The 'Bamboo Plan Summary Chart' gadget on your JIRA dashboard will now display some configuration fields. Click the dropdown arrow next to '**Chart Type**' and select '**Duration & Failed Tests (group by Build Number)**'.
    - Click the dropdown arrow next to '**Refresh Interval**' and select '**Every 15 Minutes**'.
    - Click '**Save**'.
10. Choose a different colour for your '**Bamboo Plans**' gadget:
    - Move your cursor pointer over the gadget and click the downward-pointing arrow at top right of the gadget frame.
    - Select the **purple** square in the row of colours.
11. Colour your '**Bamboo Plan Summary Chart**' gadget purple too.

Problems? Please go immediately to the Dragon Slayers' Forum.
Victory? Please continue.

The Battle is Won, the Dragon is Slain

Your JIRA dashboard now has 6 gadgets:

- The 'Bamboo Plans' gadget
- The Bamboo 'Plan Summary' gadget
- The 'FishEye Charts' gadget
- The 'Assigned to Me' gadget
- The 'Projects' gadget
- The GreenHopper 'Agile Gadget'

Screenshot 1 (click to enlarge): JIRA dashboard with 6 gadgets

Problems? Please go immediately to the Dragon Slayers' Forum.
Victory? Please continue.

Don a T-Shirt. You Rock

- **Tweet?** Tweet.
- Order your ***Atlassian DragonSlayer T-shirt*** and send us a screenshot of your JIRA dashboard via our website.

Dragon Slayers with JIRA Already Installed

Beware, all ye who enter, for here be dragons! This is the starting point for the Atlassian Dragon Quest.

By the time you reach the end of this set of instructions, you will have an awesome Atlassian integrated development suite (details here). There's a good chance that the Atlassian Integration Dragon will scorch the clothes off your back somewhere along the way, so we'll also send you a free, limited-edition Atlassian DragonSlayer T-shirt if you complete all the steps.

If you do not yet have JIRA installed, please ignore this page and start at Here Be Dragons instead.

Assumptions and Prerequisites

Before you start, please note the points below.

- **Overall requirements:** Check the hardware and software requirements.
- **JIRA Standalone:** You will need the standalone distribution of JIRA 4.0. If you have a WAR distribution, please consult our Support team .
- These instructions assume that your JIRA is running on **port 8080** (JIRA's default port). If not, please adjust the instructions accordingly.

Getting help
If you run into problems at any stage of the integration procedure, please go immediately to the Dragon Slayers' Forum. Please don't try to battle on alone. Instead, ask for help immediately. We'll be monitoring the forum 24 hours a day, and you're sure to meet other battle-weary dragon slayers there too.

Rushing into the Dragon's Lair

Don your armour and alert your serfs
If you like, you can tweet your status.

You're ready to start stage 1. Meet the dragon if you dare! Follow these stages first:

- Dragons with JIRA Stage 1 - Install Java, PostgreSQL and Crowd
- Dragons with JIRA Stage 2 - Set Up JIRA

Then join the rest of the brave dragon slayers at stage 3:

- Dragons Stage 3 - Install GreenHopper into JIRA
- Dragons Stage 4 - Install Confluence
- Dragons Stage 5 - Install FishEye
- Dragons Stage 6 - Get JIRA and FishEye Talking
- Dragons Stage 7 - Install Bamboo
- Dragons Stage 8 - Bamboo Gadgets and JIRA Victory

Follow yon Brave Dragon Slayers
On the Atlassian Dragons Twitter stream.

Dragons with JIRA Stage 1 - Install Java, PostgreSQL and Crowd

Beware, all ye who enter, for here there be dragons. You are embarking on stage 1 of the Atlassian Dragon Quest.

In this stage, you will install Java and a database (PostgreSQL) to hold the data for your Atlassian applications. Then you will set up Atlassian Crowd for centralised user management and single sign-on (SSO).

This procedure assumes that you **already have JIRA installed**. If you do not yet have JIRA, please ignore this page and start at Here Be Dragons instead.

Time estimate: This stage will take approximately **60 minutes**.

On this page:

- Step 1. Check your Java Development Kit
- Step 2: Install your PostgreSQL Database Server
- Step 3. Create your Crowd Database in PostgreSQL
- Step 4. Install Crowd
- Step 5. Set Up Crowd
- Victory!

Step 1. Check your Java Development Kit

Requirements: **Sun JDK 1.5 or higher**. Note that the JRE alone is not enough.

If you do not have the right version of the Java Development Kit (JDK) already installed, follow the steps below to get it.

1. Download the Sun Java SE Development Kit – Get the latest JDK 6.
2. Follow the Sun installation instructions.
3. Make sure you have a `JAVA_HOME` environment variable pointing to the root directory of the JDK. Some JDK installers set this automatically.
    - Check by typing one of the following into a command window, depending on your operating system.
        - On Windows: `echo %JAVA_HOME%`
        - On Linux or UNIX: `echo $JAVA_HOME`
    - If the above command does not show you the path to your JDK, please refer to the Crowd instructions on setting JAVA_HOME.

Step 2: Install your PostgreSQL Database Server

Below are the instructions for installing and setting up a PostgreSQL database server. If your JIRA installation is already using a different supported database server and you have a good technical knowledge of that server, you can choose to stick with it. However, for the purposes of this integrated setup exercise we do recommend PostgreSQL. Note that you will need the database server to hold the data for the other Atlassian applications that you will set up in later stages of this integration exercise.

Requirements: **PostgreSQL version 8.3.x**.

1. Download PostgreSQL – Get the latest 8.3.x. For the simplest installation, choose one of the one-click installers.
2. Install PostgreSQL. If you chose one of the PostgreSQL one-click installers, this is simple: Run the executable that you downloaded and follow the prompts. If necessary, you can refer to the PostgreSQL installation instructions.
3. Enter a password for the super user ('postgres').
4. Accept the default port `5432`.
5. Accept all the other default settings.
6. Download the PostgreSQL JDBC driver from http://jdbc.postgresql.org/download.html and save it locally for later use.
    - If you have installed JDK 6.x, get JDBC4 Postgresql Driver, Version 8.4-701.
    - If you have JDK 5.x, get JDBC3 Postgresql Driver, Version 8.4-701.

Step 3. Create your Crowd Database in PostgreSQL

Now you will create a database where the Atlassian Crowd application will store its data, and the user that Crowd will use to connect to the database.

We're using pgAdmin III, the administration user interface supplied with PostgreSQL. If you used the one-click installer, pgAdmin III will be already installed on your computer.

1. Start pgAdmin III.
2. Add a new login role called 'crowduser':
    - Right-click '**Login Roles**' and select '**New Login Role**'.
    - Enter the role '**Role name**': `crowduser`.
    - Enter a '**Password**' and enter it again to confirm it.
    - Select '**Can create database objects**'.
    - Select '**Can create roles**'.
    - Click '**OK**' to create the user.
3. Add a new database called 'crowd':
    - Right-click '**Databases**' and select '**New Database**'.
    - Enter the database '**Name**': `crowd`.
    - Select the '**Owner**': `crowduser`.
    - Click '**OK**' to create the database.

Alternatively, If you are on UNIX and do not have pgAdmin III, you can use the command line interface instead. Assuming that you are using the default installation directory of `/opt/PostgreSQL/8.3/bin/`, enter the following commands:

```
sudo -s -H -u postgres
# Create the Crowd user:
/opt/PostgreSQL/8.3/bin/createuser -S -d -r -P -E crowduser
# Create the Crowd database:
/opt/PostgreSQL/8.3/bin/createdb -O crowduser crowd
exit
```

Screenshot 1 (click to enlarge): Crowd database and user in PostgreSQL

Step 4. Install Crowd

Requirements: **Crowd 2.0.2**.

For Windows: (click to expand)

1. Go to the Atlassian download centre.
2. Download the '**Standalone (ZIP Archive)**' file for **Crowd 2.0.2**.
3. Unpack the zip archive into a directory of your choice, avoiding spaces in the directory name.
4. Tell Crowd where to find its Crowd Home directory:
   - Edit the properties file at `{CROWD_INSTALL}\crowd-webapp\WEB-INF\classes\crowd-init.properties`.
   - Complete the following line and remove the # at the beginning of the line:
     `crowd.home=`
     For example:
     `crowd.home=c:/data/crowd-home`
     (Note the forward slashes.)
5. Add the PostgreSQL JDBC driver JAR to your `{CROWD_INSTALL}\apache-tomcat\lib` directory.
6. Start your Crowd server by running `start_crowd.bat` in the directory where you unpacked Crowd.

For UNIX or Linux: (click to expand)

1. Go to the Atlassian download centre.
2. Click the 'Linux' tab and download the '**Standalone (TAR.GZ Archive)**' file for **Crowd 2.0.2**.
3. Unpack the archive into a directory of your choice, avoiding spaces in the directory name.
4. Tell Crowd where to find its Crowd Home directory:
   - Edit the properties file at `{CROWD_INSTALL}/crowd-webapp/WEB-INF/classes/crowd-init.properties`.
   - Complete the following line and remove the # at the beginning of the line:
     `crowd.home=`
     For example:
     `crowd.home=/var/crowd-home`
5. Create the above Crowd Home directory if it does not already exist, because in some cases Crowd may not create it for you.
6. Add the PostgreSQL JDBC driver JAR to your `{CROWD_INSTALL}/apache-tomcat/lib` directory.
7. Start your Crowd server by executing `start_crowd.sh` in the directory where you unpacked Crowd.

Full details are in the Crowd installation guide.

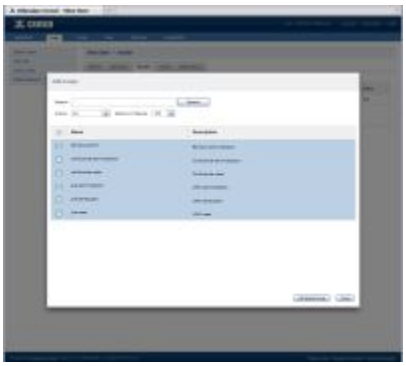
Problems? Please go immediately to the Dragon Slayers' Forum.
Victory? Please continue.

Step 5. Set Up Crowd

Now you can run Crowd's Setup Wizard, then add **Charlie of Atlassian** and the groups needed for JIRA, Confluence and the other applications.

1. To access Crowd, go to your web browser and type this address: http://localhost:8095/crowd.
2. The Crowd Setup Wizard will start up, to guide you through the process of setting up your Crowd server and creating an administration user. Detailed instructions are in the Crowd documentation. Here are the things you need to know for our Dragon Quest:
    - License – If you do not already have a Crowd license, follow the prompts on the Setup Wizard screen to get an evaluation license key.
    - Installation type – Select '**New Installation**'.
    - Database configuration – Select '**JDBC Connection**' then enter the following information to connect to your Crowd database (created above):
        - Database: `PostgreSQL`.
        - Driver Class Name – Leave this at the default value, i.e. `org.postgresql.Driver`.
        - JDBC URL – Leave this at the default value, i.e. `jdbc:postgresql://localhost:5432/crowd`.
        - Username: `crowduser`.
        - Password – The password you specified when creating your Crowd database above.
        - Hibernate Dialect – Leave this at the default value, i.e. `org.hibernate.dialect.PostgreSQLDialect`.
    - Deployment title – Enter a short, descriptive name. If you will only have one Crowd installation, then 'Crowd' is good enough.
    - Base URL – Enter the full website address at which Crowd is running, not just 'localhost'. For example, if your computer name is 'coopers' then the base URL should be: `http://coopers:8095/crowd`. Or specify a website address, such as `http://www.foobar.com:8095/crowd`
    - Email details – Enter the details of your administrator email account. We recommend that you give your own email account details here.
    - Internal directory – This is the Crowd directory that will hold your users and groups. Enter the following information, and leave the other fields at the default values:
        - Name: `Crowd`.
        - Description: `Crowd User Directory`.
    - Default administrator – This is the Crowd super user. Enter the following information:
        - Email address – Enter the address of your administrator email account. We recommend that you give your own email address here.
        - Username – Enter the administrator's login name: **`charlie`**.
        - Password – Enter a password for the administrator account and enter it again to confirm it.
        - Enter a first name for your administrator: **`Charlie`**.
        - Enter a last name for your administrator: **`of Atlassian`**.
    - Integrated applications – Leave both selected, as is the default.
3. Log in to Crowd with username **`charlie`**.
4. Add the group that will hold all your JIRA users:
    - Click '**Groups**' the top navigation bar and then click '**Add Group**'.
    - Enter the following information:
        - Group name: `jira-users`.
        - Description: `JIRA users`.
        - Directory: `Crowd`.
        - Active – Leave this checkbox selected.
    - Click '**Create**' to add the group.
5. Add the following groups too, all in the same 'Crowd' directory. These groups are needed for JIRA, Confluence and Bamboo:
    - `jira-developers` — `JIRA developers`
    - `jira-administrators` — `JIRA administrators`
    - `confluence-users` — `Confluence users`
    - `confluence-administrators` — `Confluence administrators`
    - `bamboo-admin` — `Bamboo administrators`
6. Make **Charlie of Atlassian** an administrator in JIRA, Confluence and Bamboo by adding him to the relevant groups:
    - Click '**Users**' in the the top navigation bar and find '**Charlie of Atlassian**'.
    - Click the name to view Charlie's user information.
    - Click the '**Groups**' tab, then click '**Add Groups**'.
    - The 'Add Groups' screen will appear. Click '**Search**' to see all the groups in the directory.
    - Select the checkbox at top left, next to the 'Name' column, to select all groups.
    - Click '**Add Selected Groups**' to add Charlie to the groups.

Screenshot 2 (click to enlarge): Adding Charlie to groups in Crowd



Problems? Please go immediately to the Dragon Slayers' Forum.
Victory? Please continue.

Victory!

Charlie of Atlassian can now log into Crowd. If he checks his profile (using the 'My Profile' link at top right of the Crowd screen), he will see the groups he belongs to.

Screenshot 3 (click to enlarge): Charlie's profile showing the groups he belongs to

Problems? Please go immediately to the Dragon Slayers' Forum.
Victory? Please continue.

Take a Bow and Move to the Next Stage

- **Tweet?** Tweet.
- Go to Dragons with JIRA Stage 2 - Set Up JIRA.

Dragons with JIRA Stage 2 - Set Up JIRA

Beware of fiends and dragons on the gargoyled eaves. You are embarking on stage 2 of the Atlassian Dragon Quest.

In this stage, you will configure Atlassian JIRA for bug tracking and issue management. You will also hook JIRA up to Crowd, for SSO and centralised user management.

Time estimate: This stage will take approximately **60 minutes**.

On this page:

- Step 1. *Optional:* Create your JIRA Database in PostgreSQL
- Step 2. Upgrade JIRA If Necessary
- Step 3. Configure JIRA Options
- Step 4. Import your JIRA Users into Crowd
- Step 5. Hook JIRA up to Crowd
- Step 6. Set up a Project and Create your JIRA Dashboard
- Victory!

Step 1. *Optional:* Create your JIRA Database in PostgreSQL

Below are the instructions for creating a JIRA database in a PostgreSQL database server.

- If your JIRA installation is already using a different supported database server and you have a good technical knowledge of that server, you can choose to stick with that server and skip this step.
- If your JIRA installation is using the default HSQLDB, supplied with JIRA for evaluation purposes, you will need to migrate to another database before using JIRA in a production environment. Please follow the instructions on migrating your JIRA data to an external database.

Now you will create a database where the Atlassian JIRA application will store its data, and the user that JIRA will use to connect to the database. We are assuming that you have already created your PostgreSQL database server in **Dragons Stage 1**.

We are using pgAdmin III, the administration user interface supplied with PostgreSQL. If you used the one-click installer when installing PostgreSQL, pgAdmin III will be already installed on your computer.

1. Start **pgAdmin III**.
2. Add a new login role called 'jirauser':
   - Right-click '**Login Roles**' and select '**New Login Role**'.
   - Enter the role '**Role name**': `jirauser`.
   - Enter a '**Password**' and enter it again to confirm it.
   - Select '**Can create database objects**'.
   - Select '**Can create roles**'.
   - Click '**OK**' to create the user.
3. Add a new database called 'jira':
   - Right-click '**Databases**' and select '**New Database**'.
   - Enter the database '**Name**': `jira`.
   - Select the '**Owner**': `jirauser`.
   - Click '**OK**' to create the database.

Alternatively, If you are on UNIX and do not have pgAdmin III, you can use the command line interface instead. Assuming that you are using the default installation directory of `/opt/PostgreSQL/8.3/bin/`, enter the following commands:

```
sudo -s -H -u postgres
# Create the JIRA user:
/opt/PostgreSQL/8.3/bin/createuser -S -d -r -P -E jirauser
# Create the JIRA database:
/opt/PostgreSQL/8.3/bin/createdb -O jirauser jira
exit
```

Screenshot 1 (click to enlarge): JIRA database and user in PostgreSQL

Step 2. Upgrade JIRA If Necessary

Requirements: **JIRA 4.0.0**.

1. Check your version of JIRA.
2. If you do not have **JIRA 4.0.0** or later, follow the instructions on upgrading to JIRA 4.0.

Problems? Please go immediately to the Dragon Slayers' Forum.
Victory? Please continue.

Step 3. Configure JIRA Options

In this step you will enable some JIRA features that are required for the later stages in this integration procedure.

1. Log in to JIRA with an administrator account.
2. Create a new administrator account for **Charlie of Atlassian**:
    - Click '**Administration**' in JIRA's top navigation bar.
    - The 'Projects' administration screen will appear. Click '**User Browser**' in the left-hand panel.
    - The 'User Browser' screen will appear. Click '**Add User**'.
    - The 'Create New User' screen will appear. Enter the following information:
        - Username: **`charlie`**.
        - Password – Enter a password for the administrator account and enter it again to confirm it.
        - Full name: **`Charlie of Atlassian`**.
        - Email address – We recommend that you give your own email address here.
    - Click '**Create**'.
    - Now you will add Charlie to the '`jira-administrators`' group. Click '**Group Browser**' in the left-hand panel.
    - Click the '**`jira-administrators`**' group.
    - Click '**Edit Members**'.
    - Select '**`charlie`**' in the list under '**Join**'.
    - Click '**Join**'.
3. Check JIRA's base URL:
    - Click '**General Configuration**' in the left-hand panel.
    - Change the '**Base URL**' if necessary. It must contain the full website address at which JIRA is running, not just 'localhost'. For example, if your computer name is 'coopers' then the base URL should be: `http://coopers:8080`. Or specify a website address, such as `http://www.foobar.com:8080`.
4. Check the following configurations and update them if necessary:
    a. Configure JIRA to use the wiki renderer for comments and descriptions:
        - Click '**Administration**' in the top navigation bar.
        - Click '**Field Configurations**' in the left-hand panel (in the 'Issue Fields' section).
        - The 'View Field Configurations' screen will appear. Click '**Configure**' next to 'Default Field Configuration'.
        - The 'View Field Configuration' screen will appear.
        - To change the renderer for comments, click '**Renderers**' next to '**Comment**' and then select '**`Wiki Style Renderer`**' as the 'Active Renderer'. Click '**Update**' and then click it again to confirm the change.
        - To change the renderer for issue descriptions, click '**Renderers**' next to '**Description**' and then select '**`Wiki Style Renderer`**' as the 'Active Renderer'. Click '**Update**' and then click it again to confirm the change.
    b. Turn on time tracking, so that you will be able to log the amount of time you spend working on issues:
        - Click '**Time Tracking**' in the left-hand panel (in the 'Global Settings' section).
        - In the '**Hours Per Day**' field, select the number of hours in your organisation's working day (e.g. 8).
        - In the '**Days Per Week**' field, select the number of days in your organisation's working week (e.g. 5).
        - Leave the '**Time Format**' as '`pretty`' and leave the '**Default Unit**' as '`minute`'.
        - Click '**Activate**'.
    c. Turn on the public API and allow unassigned issues:
        - Click '**General Configuration**' in the left-hand panel (in the 'Global Settings' section).
        - Click '**Edit Configuration**'.
        - Select the '**`ON`**' radio button next to '**Accept remote API calls**'.
        - Select the '**`ON`**' radio button next to '**Allow unassigned issues**'.
        - Click '**Update**'.
    d. Enable sub-tasks:
        - Click '**Sub-Tasks**' in the left-hand panel (in the 'Global Settings' section).
        - The 'Sub-Tasks' screen will appear. Click '**Enable**'.
5. Log out of JIRA, but leave JIRA running. (Click the dropdown arrow next to the name '**Charlie of Atlassian**', then select '**Log Out**'.)

Problems? Please go immediately to the Dragon Slayers' Forum.
Victory? Please continue.

Step 4. Import your JIRA Users into Crowd

In this step you will import your existing JIRA users and groups into Crowd.

For the purposes of this integration exercise, we assume that you currently have your users and groups defined in JIRA. If you are using **LDAP**, please do the following:

- Follow the steps in the Crowd documentation.
- Skip the rest of this step.

1. Ensure that the database drivers for your JIRA database are on Crowd's classpath:
    - If you are using the PostgreSQL database described in step 1 above, then the database drivers are already in Crowd. There is no need to do anything here.
    - If you are using a different database server, copy the JDBC driver JAR for your particular JIRA database across to your Crowd installation directory:
        - In Windows: `{CROWD_INSTALL}\apache-tomcat\common\lib`
        - In UNIX: `{CROWD_INSTALL}/apache-tomcat/common/lib`
    - Restart Crowd.
2. If Crowd is not already running, start it up by running `{CROWD_INSTALL}\start_crowd.bat` (on Windows) or `{CROWD_INSTALL}/start_crowd.sh` (on UNIX).
3. Go to your Crowd URL in your browser, e.g. http://www.foobar.com:8095/crowd.
4. Log in to Crowd with username **`charlie`**.
5. Click '**Users**' in Crowd's top navigation bar.
6. The 'User Browser' will appear. Click '**Import Users**'.
7. The 'Import Type' screen will appear. Click '**Atlassian Importer**'.
8. The 'Options' screen will appear. Enter the following information:
    - Atlassian Product: `JIRA`.
    - Directory: `Crowd`
    - Import Passwords – Select this checkbox.
    - Product Database URL – Enter the URL of your JIRA instance's database. The exact syntax will depend on your database server. If you are using the PostgreSQL database described in step 1 above, then the value will be: `jdbc:postgresql://localhost:5432/jira`.
    - Database Driver – Enter the class name of your JIRA JDBC driver. If you are using the PostgreSQL database described in step 1 above, then the value will be: `org.postgresql.Driver`.
    - Username – Enter the username that Crowd will use to access your JIRA database. If you are using the PostgreSQL database described in step 1 above, then the value will be: `jirauser`
    - Password – Enter the password of the above database user.
9. Click '**Continue**' to import the users from your JIRA installation into your Crowd directory.
10. The '**Results**' screen will show how many users and groups have been imported into your Crowd directory.

Problems? Please go immediately to the Dragon Slayers' Forum.
Victory? Please continue.

Step 5. Hook JIRA up to Crowd

In this step you will define the JIRA application in Crowd and configure JIRA to use Crowd for SSO and centralised user management.

1. If Crowd is not already running, start it up by running {CROWD_INSTALL}\start_crowd.bat (on Windows) or {CROWD_INSTALL}/start_crowd.sh (on UNIX).
2. Go to your Crowd URL in your browser, e.g. http://www.foobar.com:8095/crowd.
3. Log in to Crowd with username **charlie**.
4. Click '**Applications**' in the top navigation bar.
5. The 'Application Browser' will appear. Click '**Add Application**' in the left-hand menu.
6. This will display the first screen for the 'Add Application' wizard for Crowd. Enter the following information:
    - Application Type: JIRA.
    - Name: jira.
    - Description: Atlassian JIRA.
    - Password – Enter the password that JIRA will use to access Crowd and enter it again to confirm it.
    - URL – Enter the base URL of your JIRA site, e.g. http://www.foobar.com:8080.
    - Click '**Resolve IP Address**' to ask Crowd to find the '**Remote IP Address**' for you. The value will be something like this: 127.0.0.1.
    - Select the '**Crowd**' directory.
    - Select '**Allow all users to authenticate**'.
    - Click '**Add Application**'.
7. Check the IP addresses for your JIRA application:
    - Click the '**Remote Addresses**' tab.
    - Add your JIRA host name, e.g. http://www.foobar.com:8080.
    - If it's not already present, add: 127.0.0.1.
8. Leave Crowd up and running, but shut down JIRA. (Press Ctrl+C in your JIRA server command window or run {JIRA_INSTALL}\bin\shutdown.bat (on Windows) or {JIRA_INSTALL}/bin/shutdown.sh (on UNIX).)
9. Copy the Crowd client libraries and configuration files to your JIRA installation folder:
    - Delete the existing crowd-integration-client-1.6.1.jar file from {JIRA_INSTALL}/atlassian-jira/WEB-INF/lib folder.
    - Copy {CROWD_INSTALL}/client/crowd-integration-client-2.0.1.jar to {JIRA_INSTALL}/atlassian-jira/WEB-INF/lib.
    - Copy {CROWD_INSTALL}/client/conf/crowd.properties to {JIRA_INSTALL}/atlassian-jira/WEB-INF/classes.
    - Copy {CROWD_INSTALL}/client/conf/crowd-ehcache.xml to {JIRA_INSTALL}/atlassian-jira/WEB-INF/classes.
10. Edit the {JIRA_INSTALL}/atlassian-jira/WEB-INF/classes/crowd.properties file and change the following properties:
    - application.name: jira
    - application.password – Enter the password that JIRA will use to access Crowd. This must be the same password as you entered in the Crowd 'Add Application' wizard above.
11. Edit the {JIRA_INSTALL}/atlassian-jira/WEB-INF/classes/osuser.xml file. Comment out any existing authentication providers and uncomment the Crowd providers, as instructed in the text of the file itself.
12. Edit the {JIRA_INSTALL}/atlassian-jira/WEB-INF/classes/seraph-config.xml file. Comment out the 'JiraOsUserAuthenticator' class and uncomment the 'JIRAAuthenticator' class, as instructed in the text of the file itself.
13. Start your JIRA server again, and go to your JIRA URL in your browser, e.g. http://www.foobar.com:8080.
14. Log in to JIRA with username **charlie** and Charlie's password in Crowd.

    You are now authenticating via Crowd!
15. Turn on external user management in JIRA, so that all user management happens in Crowd rather than JIRA:
    - Click '**Administration**' in the top navigation bar.
    - Click '**General Configuration**' in the left-hand panel (in the 'Global Settings' section).
    - Click '**Edit Configuration**'.
    - Change '**Mode**' to '**Private**'.
    - Select the '**ON**' radio buttons next to '**External user management**' and '**External password management**'.
    - Click '**Update**'.

Screenshot 3: The JIRA application defined in Crowd – 'Remote Addresses' tab

Full details are in the Crowd documentation.

The default JIRA groups are: jira-administrators, jira-developers and jira-users.
If your JIRA installation includes additional groups, over and above the default three, you will need to give the imported groups access to the JIRA application in Crowd. See Specifying which Groups can access an Application.

Problems? Please go immediately to the Dragon Slayers' Forum.
Victory? Please continue.

Step 6. Set up a Project and Create your JIRA Dashboard

In this step you will create some data in JIRA, including a project and an issue, for use in the subsequent stages of this integration procedure. Then you will create your own JIRA dashboard with a couple of gadgets.

1. Create a project in JIRA:
    - Click '**Administration**' in the top navigation bar.
    - Click '**Projects**' in the left-hand panel, then click '**Add Project**'.
    - Enter the following information:
        - Name: **`Dragons`**.
        - Key: **`DRA`**.
        - Project Lead: **`charlie`**.
        - Description: `Atlassian Dragon Quest`.
    - Leave the rest of the fields with their default values. Click '**Add**'.
2. Add two versions (1.0 and 2.0):
    - Click '**Manage** versions'.
    - Enter the following information then click '**Add**':
        - Version Name: `1.0`.
        - Description: `Version 1.0`.
    - Follow the same steps to add `Version 2.0`.
3. Add an issue to your project:
    - Click '**Create Issue**' at top right of the screen, select the following options then click '**Create**':
        - Project: `Dragons`.
        - Issue Type: `Bug`.
    - Enter the following information about your new issue then click '**Create**':
        - Summary: `Dragon slayer's equipment is defective`
        - Affects Version/s: `1.0`.
        - Assignee: `Charlie of Atlassian` – Click '**Assign to me**'.
        - Description: `There's a hole in the dragon slayer's water bucket.`
        - Original Estimate: `1d`.
    - You now have an issue with a key of '**DRA-1**'.
4. Create a new dashboard for all your dragon-related tasks, issues and general fire fighting:
    - Click '**Dashboards**' at top left of your JIRA screen.
    - Click '**Tools**' at top left of the screen, then '**Create Dashboard**'.
    - The 'Create New Dashboard' screen will appear. Enter the following information:
        - Name: **`Dragon Development Dashboard`**.
        - Description: `A dashboard for dragon slayers, fire fighters and like-minded brave souls.`
    - Leave the other fields at their default values and click the '**Add**' button at the **bottom** of the 'Create New Dashboard' screen (not the one next to 'Add Shares').
5. You now have a new, empty dashboard. Add the 'Projects' gadget to the dashboard:
    - Click '**Add Gadget**'.
    - The 'Gadget Directory' will appear, showing a list of the available gadgets for your JIRA dashboard. Enter 'projects' into the search box at top right of the Gadget directory screen.
    - The list of gadgets will change, to show only the gadgets that match your search term. Find the '**Projects**' gadget and click '**Add it Now**'. The gadget will be highlighted for a short time and the button's wording will change to 'Adding', while JIRA adds the gadget to the dashboard.
6. Find and add the '**Assigned To Me**' gadget in the same way.
7. Click '**Finished**' to go back to your dashboard.
8. Drag the 'Assigned to Me' gadget to the top right of your dashboard:
    - Move your mouse pointer over the gadget's blue title bar.
    - The cursor icon will change to a four-pointed arrow. Click the gadget title bar with the left mouse button then drag the gadget to the right. Drop it in the space labelled 'Drag your gadget here.'
9. Configure the 'Assigned to Me' gadget to point to your 'Dragons' project:
    - Refresh the dashboard, if necessary, to show the 'Number of Results' and other configuration fields in the gadget.
    - Leave the default values as configured for '**Number of Results**' and '**Columns to display**'.
    - Click the dropdown arrow next to '**Refresh Interval**' and select '**Every 15 Minutes**'.
    - Click '**Save**'.
10. Configure the 'Projects' gadget:
    - Leave the default values as configured for '**Projects**', '**View**' and '**Number of Columns**'.
    - Click the dropdown arrow next to '**Refresh Interval**' and select '**Every 15 Minutes**'.
    - Click '**Save**'.

Problems? Please go immediately to the Dragon Slayers' Forum.
Victory? Please continue.

Victory!

You can now see your project dashboard with 2 gadgets on it! The 'Projects' gadget shows the project lead **Charlie of Atlassian**. The 'Assigned to Me' gadget shows the single **DRA-1** issue assigned to Charlie.

Screenshot 4 (click to enlarge): JIRA dashboard with 2 gadgets

Problems? Please go immediately to the Dragon Slayers' Forum.
Victory? Please continue.

Take a Bow and Move to the Next Stage

- **Tweet?** Tweet.
- Join the mainstream dragon slayers! Go to Dragons Stage 3 - Install GreenHopper into JIRA.