



ATLASSIAN

DOCUMENTATION

1. Guide to Installing an Atlassian Integrated Suite	2
1.1 Here Be Dragons	2
1.1.1 Dragons Stage 1 - Install Java, PostgreSQL and Crowd	4
1.1.2 Dragons Stage 2 - Install JIRA	8
1.1.3 Dragons Stage 3 - Install GreenHopper into JIRA	15
1.1.4 Dragons Stage 4 - Install Confluence	19
1.1.5 Dragons Stage 5 - Install FishEye and Crucible	26
1.1.6 Dragons Stage 6 - Get JIRA and FishEye Talking	32
1.1.7 Dragons Stage 7 - Get JIRA and Crucible Talking	35
1.1.8 Dragons Stage 8 - Install Bamboo	39
1.1.9 Dragons Stage 9 - Bamboo Gadgets and JIRA Victory	48
1.1.10 After Dragons	50
1.1.10.1 Adding Users to your Atlassian Integrated Suite	54
1.1.10.2 Running Bamboo in a Console Window	54
1.2 Dragon Slayers with JIRA Already Installed	54
1.2.1 Dragons with JIRA Stage 1 - Install Java, PostgreSQL and Crowd	55
1.2.2 Dragons with JIRA Stage 2 - Set Up JIRA	60

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 [raise a support ticket](#) for the product you're stuck on. Please don't try to battle on alone. Instead, ask for help immediately. You can also seek assistance on the [Dragon Slayers' Forum](#), where you're sure to meet other battle-weary dragon slayers.

Rushing into the Dragon's Lair



Don your armour and alert your serfs

If you like, you can [tweet your status](#).



Follow yon brave dragon slayers

On the [Atlassian Dragons Twitter stream](#).

- 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 and Crucible](#)
- [Dragons Stage 6 - Get JIRA and FishEye Talking](#)
- [Dragons Stage 7 - Get JIRA and Crucible Talking](#)
- [Dragons Stage 8 - Install Bamboo](#)
- [Dragons Stage 9 - Bamboo Gadgets and JIRA Victory](#)
- [After Dragons](#)

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.
- [Crucible](#) for code reviews.
- [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 **6 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, Crucible 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](#).
- This guide assumes that you will set up all the applications on the same machine.

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](#).

i If you wish to use **Maven 2** and do not yet have it installed, we recommend the [Atlassian Plugin SDK](#). The SDK includes Maven 2 and a correctly-configured Maven `settings.xml` file, as well as a number of shell scripts to speed up and simplify plugin development. It also includes the Java Activation JAR (`javax.activation:activation:jar`) which you will need for a successful Maven build. If you would like to download the Java Activation JAR separately instead, see the [FAQ](#).

- **Java Development Kit:** You will need Sun JDK 1.6 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 1.6 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 version of the JDK 1.6, at least **version 6u23 or later**.
 - If you are running 64-bit Windows, please ensure that you use **32-bit JDK** and not the 'x64' JDK.
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 8.4.x.**

1. Download [PostgreSQL](#) – Get the latest 8.4.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. Ensure that you choose UTF8 (unicode) encoding when selecting the locale. 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 8.4.x JDBC driver from <http://jdbc.postgresql.org/download.html> and save it locally for later use. Get the [JDBC4 Postgresql Driver, Version 8.4-702](#).

 Internet Explorer may rename the file extension from '.jar' to '.zip' when you download it. If you are using Internet Explorer, please rename the file so that it has a '.jar' extension after downloading it.

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.

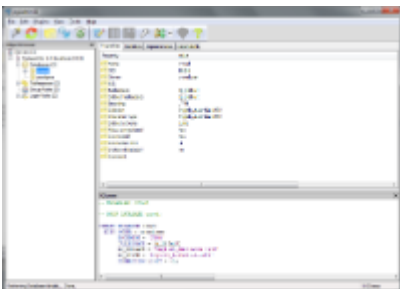
 We're using pgAdmin III, the administration user interface supplied with PostgreSQL. If you used the one-click installer, pgAdmin III will be already installed on your computer.

1. Start pgAdmin III.
2. Double-click the name of the PostgreSQL server in the **'Object browser'** window and enter the password for the 'postgres' super user as prompted.
3. Add a new login role called 'crowduser':
 - Right-click **'Login Roles'** and select **'New Login Role'**.
 - Enter the **'Role name'**: crowduser.
 - Enter a **'Password'** and enter it again to confirm it.
 - Click the **'Role privileges'** tab.
 - Select **'Can create database objects'**.
 - Select **'Can create roles'**.
 - Click **'OK'** to create the user.
4. 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/8.4/bin/, enter the following commands:

```
sudo -s -H -u postgres
# Create the Crowd user:
/opt/PostgreSQL/8.4/bin/createuser -S -d -r -P -E crowduser
# Create the Crowd database:
/opt/PostgreSQL/8.4/bin/createdb -O crowduser crowd
exit
```

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



Step 4. Install Crowd

Requirements: **Crowd 2.2.2.**

► For Windows: (click to expand)

1. Go to the Atlassian [download centre](#).
2. Download the '**Standalone (ZIP Archive)**' file for **Crowd 2.2.2**.
3. Unpack the zip archive into a directory of your choice, avoiding spaces in the directory name.
4. Tell Crowd where to put 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 2.2.2**.
3. Unpack the archive into a directory of your choice, avoiding spaces in the directory name.
4. Tell Crowd where to put 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 [raise a support ticket](#) for the product you're stuck on, or try the [Dragon Slayers' Forum](#).

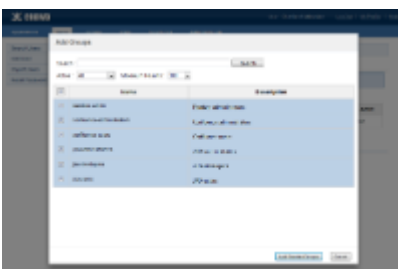
Victory? Please [continue](#).

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.
 - Session Timeout – Leave this at the default value, i.e. 30
 - 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.foo.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'** in 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 top navigation bar and find **'Charlie of Atlassian'**.
 - Click the name to view Charlie's user information.
 - Click the **'Groups'** tab under **'View User'**, 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 raise a support ticket for the product you're stuck on, or try 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 raise a support ticket for the product you're stuck on, or try 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**.

ℹ We are using **pgAdmin III**, the administration user interface supplied with PostgreSQL. If you used the one-click installer when installing

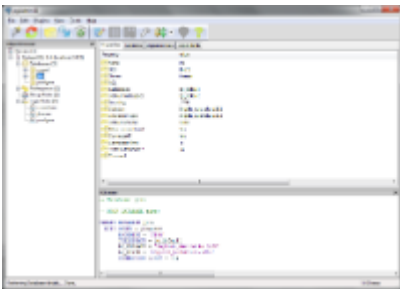
PostgreSQL, pgAdmin III will be already installed on your computer.

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.
 - Click the '**Role privileges**' tab.
 - 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/8.4/bin/, enter the following commands:

```
sudo -s -H -u postgres
# Create the JIRA user:
/opt/PostgreSQL/8.4/bin/createuser -S -d -r -P -E jirauser
# Create the JIRA database:
/opt/PostgreSQL/8.4/bin/createdb -O jirauser jira
exit
```

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



Step 2. Install JIRA

Requirements: **JIRA 4.3**.

⚠ Do *not* use the 'Windows Installer' for this integration exercise. Please make sure you follow the instructions below to download the 'Standalone (**ZIP Archive**)' file.

▼ 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 4.3**.
 - ⚠ 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. Run the JIRA Configuration Tool at `{JIRA_INSTALL}\bin\config.bat`
 - Tell JIRA where to put its JIRA Home directory under the '**JIRA Home**' tab. For example:
C:/data/jira-home
 - Configure the database connection under the '**Database**' tab.
 - **Database type:** PostgreSQL.
 - **Hostname** – Enter the name or IP address of the server that you installed your PostgreSQL database on, i.e. localhost.
 - **Port** – Enter the default port that you set up PostgreSQL with, i.e. 5432.
 - **Database** – This is the name of the database that you created in step 1 above, i.e. jira.
 - **Username** – This is the user you created in step 1 above, i.e. jirauser.
 - **Password** – Enter the password you chose in step 1 above.
 - **Schema** – Accept the default '**public**' schema.
6. Click the '**Test Connection**' button to test the connection settings. The tool will attempt to connect to the database, and give a message with the results.
7. Accept the default '**Pool Size**' setting.
8. Click '**Save**' when you have a working connection and click '**Close**'.
9. 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 4.3**.
3. Unpack the archive into a directory of your choice, avoiding spaces in the directory name.
4. Run the JIRA Configuration Tool at `{JIRA_INSTALL}/bin/config.sh`
 - Tell JIRA where to put its JIRA Home directory under the '**JIRA Home**' tab. For example:
`/usr/local/jira-home/`
 - Configure the database connection under the '**Database**' tab.
 - **Database type:** PostgreSQL.
 - **Hostname** – Enter the name or IP address of the server that you installed your PostgreSQL database on, i.e. localhost.
 - **Port** – Enter the default port that you set up PostgreSQL with, i.e. 5432.
 - **Database** – This is the name of the database that you created in step 1 above, i.e. jira.
 - **Username** – This is the user you created in step 1 above, i.e. jirauser.
 - **Password** – Enter the password you chose in step 1 above.
 - **Schema** – Accept the default '**public**' schema.
5. Click the '**Test Connection**' button to test the connection settings. The tool will attempt to connect to the database, and give a message with the results.
6. Accept the default '**Pool Size**' setting.
7. Click '**Save**' when you have a working connection and click '**Close**'.
8. Start your JIRA server by running `{JIRA_INSTALL}/bin/startup.sh`.

Full details are in the JIRA installation guide.

Problems? Please raise a support ticket for the product you're stuck on, or try 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:
 - Application Title – Accept the default application title.
 - Mode – Accept the default mode.
 - 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. Turn on the public API and allow unassigned issues:
 - Click '**Administration**' in the top navigation bar.
 - Click '**General Configuration**' in the left-hand panel (in the 'Global Settings' section).
 - Enter your password as prompted, to confirm that you want administrator access. (Note that the Atlassian applications will request this confirmation at various steps in the process. This guide will not mention this step again.)
 - Click '**Edit Configuration**'.
 - Select the 'on' radio button next to '**Allow unassigned issues**'.
 - Select the 'on' radio button next to '**Accept remote API calls**'.
 - Click '**Update**'.

Screenshot 2: The JIRA Dashboard when you first log in

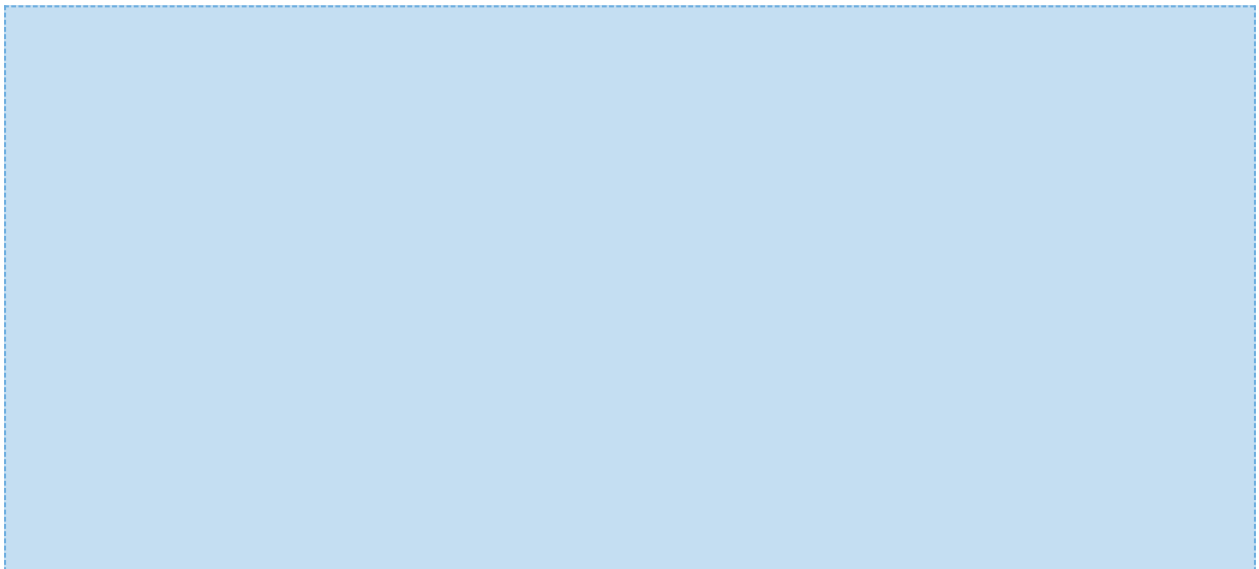


Problems? Please [raise a support ticket](#) for the product you're stuck on, or try the [Dragon Slayers' Forum](#).

Victory? Please continue.

Step 4. Hook JIRA up to Crowd

In this step you will configure JIRA to use Crowd for SSO and centralised user management. To do that, you will define the JIRA application in Crowd, define the Crowd user directory in JIRA, and configure the SSO property files.



Screenshot 3: User directories in JIRA





Full details are in the [Crowd documentation](#) and the [JIRA administrator's guide](#).

Problems? Please [raise a support ticket](#) for the product you're stuck on, or try 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 right 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 [raise a support ticket](#) for the product you're stuck on, or try 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 raise a support ticket for the product you're stuck on, or try 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. Use the Scrum Template for your Project and Add a Story
- Step 4. Add the GreenHopper Gadget to your JIRA Dashboard
- Victory!

Step 1. Install GreenHopper Plugin into JIRA

Requirements: **GreenHopper 5.5 for JIRA 4.3**.

In this step you will install the GreenHopper plugin into JIRA.

1. Go to your JIRA URL in your browser, e.g. <http://www.foobar.com:8080>.
2. Select '**Plugins**' from the '**System**' section of the 'Administration' menu.
3. The plugin management page will appear, showing the 'Manage Existing' tab. Click the '**Install**' tab.
4. Enter `greenhopper` into the search box and click '**Search**'.
5. The GreenHopper plugin will appear in the list of plugins. Click the '**GreenHopper**' plugin name.
6. The plugin details will appear. Click '**Install Now**'.
7. When the plugin has been successfully installed, 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).)
8. Start your JIRA server again, and log in to JIRA as **charlie**.
9. Set up your GreenHopper license key:
 - Select '**License Details**' from the '**GreenHopper**' section of the 'Administration' menu.
 - The 'GreenHopper License Information' screen will appear. Paste your Greenhopper license key into the '**GreenHopper License**' textbox. If you do not already have a GreenHopper license, follow the prompts on the 'GreenHopper License' screen to get a '**GreenHopper for JIRA 4: Evaluation**' license key.
 - Click '**Add**'.
10. 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 raise a support ticket for the product you're stuck on, or try the [Dragon Slayers' Forum](#).

Victory? Please continue.

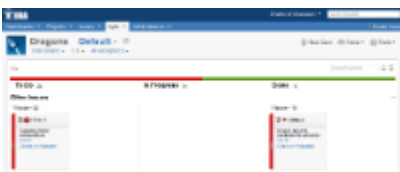
Step 2. Add Another JIRA Issue and a Sprint

Now that you have GreenHopper you can choose to update your issues via the GreenHopper tabs or via the standard JIRA interface. For this exercise you will do your updates via GreenHopper.

First you will create two 'sprints', also known as 'milestones'. A sprint is a short period of time, two weeks for example, 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. Finally you will add your existing issue to the same sprint.

1. Click the **'Add'** button above the version cards, near the top right of the planning board.
2. The 'Add Version' screen will appear. Add a sprint with the following information:
 - Version Name: 2.0.S1
 - Description: Version 2.0 Sprint 1
 - Leave the default values for the other fields.
 - Click **'Create and Close'**.
3. Follow the above steps to add another sprint with the following information:
 - Version Name: 2.0.S2
 - Description: Version 2.0 Sprint 2
4. 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 sprint box for sprint **'2.0.S1'**.
5. Within the sprint box click the gear icon (top right) and select **'Toggle visibility'** or double click the sprint title bar to expand the box.
6. The **'Parent'** is currently set to 'none'. Click **'Parent'**.
7. A dropdown list will appear. Select **'2.0'**.
8. 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'.
9. Edit the **'Parent'** for sprint **'2.0.S2'** in the same way.
10. You now have two sprints within version 2.0. Next you will add a new issue, also known as a 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.
11. Click **'Create and Close'**, to create the issue.
12. You will see your planning board again. It currently shows no cards. Click the version number dropdown near the top left of the screen (currently showing '2.0.s2') and select **'Unscheduled'** to see all the cards.
13. Click the card for issue **DRA-1**, drag the card to the right and drop it onto the box for sprint **'2.0.S1'**.
14. Drag **DRA-2** to sprint **'2.0.S1'** as well.
15. Your planning board will now be empty, because it is currently showing cards for version 'Unscheduled'. Click the version number **'2.0.S1'** at the top of the version 2.0.S1 box on the right. You should now see your two cards **'DRA-1'** and **'DRA-2'**.
16. Click the version number **'2.0'** at the top of the version 2.0 box on the right. Notice the following points:
 - The value in the version number 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 double click the title bar of each version box, to minimise or expand the box.
17. Mark one of your issues as complete:
 - Click the down arrow next to 'Planning Board' 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. Select **'Close Issue'** and click **'Update'**.
 - The **'Close Issue'** screen will appear. Click **'Close Issue'**.

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



Problems? Please [raise a support ticket](#) for the product you're stuck on, or try the [Dragon Slayers' Forum](#).

Victory? Please continue.

Step 3. Use the Scrum Template for your Project and Add a Story

GreenHopper includes a project template for projects using the Scrum methodology. The template adds custom fields like ranking fields, story points, and so on, to your project. It also allows you to create stories, epics and technical tasks. In this step you will apply the Scrum template to your project and create a story.

1. Open the **'Tools'** menu on the planning board and click **'Configuration'**.
2. The project configuration page will appear. Click the **'General'** tab.
3. Select **'Scrum'** from the **'Project Template'** dropdown menu.
4. Click **'Change template'** on the confirmation window.
5. Next you will create a new story. Go back to your planning board and click **'New Card'**. Enter the following values:
 - Card type: Story
 - Priority: Major
 - Summary: As a dragon slayer I would like to wield an extremely big sword.
 - Version: 2.0
 - Component: Unknown
 - Business Value: 10
 - Assignee: Charlie of Atlassian – Click the **'Assign to me'** icon.
 - Story Points: 10
6. Click **'Create and Close'**. You will see your planning board again, showing your story.

Screenshot 3 (click to enlarge): The GreenHopper planning board with story



Problems? Please raise a support ticket for the product you're stuck on, or try the [Dragon Slayers' Forum](#).

Victory? Please continue.

Step 4. 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 **'GreenHopper 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 **'GreenHopper Agile Gadget'**:
 - Start typing **'Dragons'** in the **'Project or Saved Filter'** box and select **'Dragons (DRA)'** from the dropdown list that appears.
 - Leave the default value for **'Display chart values'** and **'Display chart legend'**.
 - Click the dropdown arrow next to **'Refresh Interval'** and select **'Every 15 Minutes'**.
 - Click **'Save'**.
 - Click the version dropdown arrow next to 'Unscheduled' 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 **'GreenHopper Agile Gadget'** 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 raise a support ticket for the product you're stuck on, or try 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 4 (click to enlarge): JIRA dashboard with 3 gadgets



Problems? Please raise a support ticket for the product you're stuck on, or try 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 hook Confluence up to [Crowd](#) for SSO and centralised user management, and get your JIRA and Confluence sites talking to each other. Then you will create a wiki space, add a dynamic display of JIRA issues to a wiki page, and add a Confluence activity stream to your JIRA dashboard.

Time estimate: This stage will take approximately **60 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. Get JIRA and Confluence Talking](#)
- [Step 6. Create a Wiki Space](#)
- [Step 7. Add Some JIRA Issues to your Confluence Page](#)
- [Step 8. Add a Confluence Gadget to JIRA](#)
- [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](#)

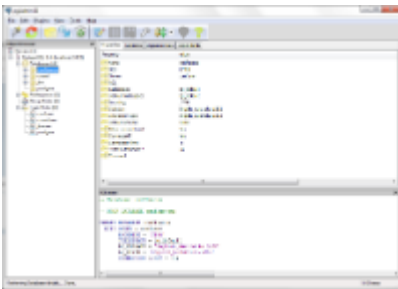
Stage 1, pgAdmin III will be already installed on your computer.

1. Start **pgAdmin III**.
2. Add a new login role called 'confuser':
 - Right-click '**Login Roles**' and select '**New Login Role**'.
 - Enter the '**Role name**': `confuser`.
 - Enter a '**Password**' and enter it again to confirm it.
 - Click the '**Role privileges**' tab.
 - 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/8.4/bin/createuser -S -d -r -P -E confuser
# Create the Confluence database:
/opt/PostgreSQL/8.4/bin/createdb -O confuser confluence
exit
```

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



Step 2. Install Confluence

Requirements: **Confluence 3.5**.

▼ For Windows: (click to expand)

1. Go to the Atlassian [download centre](#).
2. Download the '**Standalone (ZIP Archive)**' file for **3.5**.
 - ⚠ 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. If you cannot see the 'Zip Archive', click the '**Show all**' link above the download buttons to see all the download file types.
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. 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 3.5**.
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. Start your Confluence server by running {`CONFLUENCE_INSTALL`}/bin/startup.sh.

Problems? Please [raise a support ticket](#) for the product you're stuck on, or try 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 generate an evaluation license online.
4. Click '**Production Installation**' under 'Choose Installation Type'.
5. The 'Choose a Database Configuration' screen will appear. Connect Confluence to your PostgreSQL database:
 - 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.
 - Enter the following information:
 - Driver Class Name: `org.postgresql.Driver` – This is the default value.
 - Database URL: `jdbc:postgresql://localhost:5432/confluence` – This is the default value.
 - Username: `confuser` – This is the user you created in step 1 (above).
 - Password – Enter the password that you chose in step 1 (above).
 - Click the '**Next**' button. You might need to wait a few minutes while Confluence sets up its database.
6. On the 'Load Content' screen, click the '**Example Site**' button to include the demonstration space content into your Confluence installation:.
7. 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.
8. Click the '**Next**' button.
9. The 'Confluence Setup Successful' screen will appear. Click '**Start using Confluence now**'.
10. The home page of the Confluence 'Demonstration Space' will appear.
11. 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**'. Confirm your password when prompted.
 - 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**' links.
 - In the '**Server Base Url**' field of the 'Site Configuration' section, enter the full website address at which Confluence is running. This address 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.

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




Problems? Please raise a support ticket for the product you're stuck on, or try the [Dragon Slayers' Forum](#).

Victory? Please continue.

Step 4. Hook Confluence up to Crowd

In this step you will configure Confluence to use Crowd for SSO and centralised user management. To do that, you will define the Confluence application in Crowd, define the Crowd user directory in Confluence, and configure the SSO property files.

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, excluding the "http://www." prefix and the ":8090" port number. e.g. `foobar.com`.
 - If this address is not already present, add it: `127.0.0.1`.
 7. Connect Confluence to the Crowd user directory:
 - Go to your Confluence URL in your browser, e.g. `http://www.foobar.com:8090`.
 - Select **'User Directories'** from the **'Security'** section of the Confluence Administration Console.
 - Click **'Add Directory'**, select type **'Atlassian Crowd'** and click **'Next'**.
 - The Crowd server configuration screen will appear. Enter the following information:
 - Name – Accept the default value, **Crowd Server**.
 - Server URL – Enter the web address of your Crowd server, e.g. `http://www.foobar.com:8095/crowd`
 - Application Name: **confluence**. This is the application name that you used to define Confluence in the Crowd **'Add Application'** wizard above.
 - Application Password – Enter the password that you defined for Confluence in the Crowd **'Add Application'** wizard above.
 - Crowd Permissions – Select **Read/Write**.
 - Leave the other settings at their default values and click the **'Test Settings'** button to test the connection.
 - When you have a working connection, click **'Save'**.
 - The **'User Directories'** screen will appear. Now you will move the Crowd directory to the top of the list of directories. Click the **blue upward arrow** in the **'Order'** column next to the **'Crowd Server'**, so that the Crowd directory moves to the top of the list.
- Here is a summary of how the directory order affects the processing:
- The order of the directories is the order in which they will be searched for users and groups.
 - Changes to users and groups will be made only in the first directory where the application has permission to make changes.
8. Log out of Confluence, but leave Confluence running.
 9. Log in to Confluence again, with the same username **charlie** and Charlie's password in Crowd.
 10.  You are now authenticating via Crowd!
 10. 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).)


11. Configure the Confluence property files for SSO:

- Edit the `{CONFLUENCE_INSTALL}/confluence/WEB-INF/classes/seraph-config.xml` file.
- Comment out the default authenticator node:

```
<!-- <authenticator
class="com.atlassian.confluence.user.ConfluenceAuthenticator"/> -->
```

- Uncomment the line that contains the new authenticator:

```
<authenticator
class="com.atlassian.confluence.user.ConfluenceCrowdSSOAuthenticator"/>
```

- Save the `seraph-config.xml` file.
 - Copy the `crowd.properties` file from `{CROWD_INSTALL}/client/conf/` to `{CONFLUENCE_INSTALL}/confluence/WEB-INF/classes`, replacing the existing file.
 - 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.
 - Save the `crowd.properties` file.
12.  You now have SSO across Confluence, JIRA and Crowd! Try it:
- Start your Confluence server again, and go to your Confluence URL in your browser, e.g. <http://www.foobar.com:8090>.
 - If you are already logged in to Crowd, you will not need to log in to Confluence. SSO ensures that you are already logged in as **charlie**.
 - Log out of Confluence.
 - Go to Crowd and click an option. Crowd will prompt you to log in. When you logged out of Confluence, SSO ensured that you logged out of Crowd and JIRA too.
 - Log in to JIRA, Confluence or Crowd. You will be logged in to all three.

Full details are in the Crowd documentation and the Confluence administrator's guide.

Problems? Please raise a support ticket for the product you're stuck on, or try the Dragon Slayers' Forum.

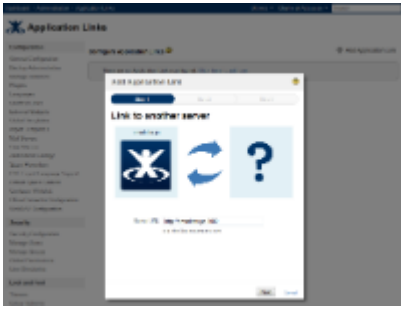
Victory? Please continue.

Step 5. Get JIRA and Confluence Talking

In this step you will set up the trusted communication channel between your JIRA and Confluence sites, so that you can display JIRA information on Confluence pages and Confluence information on the JIRA dashboard. This will be a two-way trust relationship: Confluence will trust JIRA and JIRA will trust Confluence. You will also make your Confluence gadgets available in JIRA.

1. Make sure that both JIRA and Confluence are running.
2. Go to your Confluence URL in your browser, e.g. <http://www.foobar.com:8090>.
3. Select **'Application Links'** from the **'Administration'** section of the Confluence Administration Console.
4. The 'Configure Application Links' screen will appear. Click **'Add Application Link'**.
5. The first screen of the 'Add Application Link' wizard will appear. Copy the base URL for your JIRA site (e.g. <http://coopers:8080> or <http://www.foobar.com:8080>) and paste it into the **'Server URL'** field.
6. Click **'Next'**.
7. The 'Link to JIRA' screen will appear. Enter the following information:
 - Create a link back to this server – This option is selected by default. Let it remain selected.
 - Username: `charlie`. This is the username of the administrator on your JIRA site.
 - Password – Enter Charlie's password in JIRA.
 - Reciprocal Link URL – Leave this field at its default value, pointing to your Confluence site.
8. Click **'Next'**.
9. The 'Set Users and Trust' screen will appear. Enter the following information:
 - The servers have the same set of users – This option is selected by default. Let it remain selected.
 - These servers fully trust each other – This option is selected by default. Let it remain selected.
10. Click **'Create'**.
11. Now you will make your Confluence gadgets available in JIRA, so that JIRA users will be able to add any Confluence gadget to their dashboards. Go to your JIRA browser window and click **'Dashboards'** in JIRA's top navigation bar.
12. Click **'Add Gadget'**.
13. The 'Gadget Directory' popup window will appear. Click **'Gadget Subscriptions'**.
14. The 'Gadget Subscriptions' popup window will appear. Click **'Add Subscription'**.
15. The 'Add Subscriptions' popup window will appear. Copy the base URL for your Confluence site (e.g. <http://coopers:8090> or <http://www.foobar.com:8090>) and paste it into the text box on the screen.
16. Click **'Add Subscription'**.
17. Click **'Finished'**.

 The Confluence gadgets are now available in your JIRA gadget directory. You have not yet added them to your JIRA dashboard. We will do that in a later step.

Screenshot 3 (click to enlarge): Adding JIRA as a trusted application in Confluence


The Confluence documentation has the details about [application links and trusted applications](#).


Problems? Please [raise a support ticket](#) for the product you're stuck on, or try the [Dragon Slayers' Forum](#).

Victory? Please continue.

Step 6. Create a Wiki Space

Now you can create a space in Confluence. A 'space' is a logical collection of pages, comparable to a library. A space is configurable and managed independently within the wiki site. It is 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 '**Add Space**' on the left-hand 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, '**Global Look and Feel**').
 - 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, you will 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. If prompted, allow '**Gears**' access to your site. This will allow you to drag and drop images and other attachments onto your wiki page.
 - Select and delete all the text in the editor pane. You will start with an empty page.
 - 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 upload the **Charlie badge** image that you saved earlier. Alternatively, you can drag and drop the image from your desktop into the 'Insert Image' window.
 - The image will appear in the preview panel of the 'Insert Image' window. Click '**Insert**'.
 - The image will appear in the editor pane of your home page.
 - Click '**Save**' to save your updated wiki page.


Problems? Please [raise a support ticket](#) for the product you're stuck on, or try the [Dragon Slayers' Forum](#).

Victory? Please continue.

Step 7. Add Some JIRA Issues to your Confluence Page


Now you can put some interesting JIRA content into your page. What's more, you can insert an issue into JIRA directly from your Confluence page and then display the issue on the page.

1. Edit the Confluence page again.
2. Place your cursor immediately after your **Charlie badge** image and press 'Enter' to start a new line.
3. Enter the following text onto the page: My JIRA issues.
4. Select the text that you have just entered and format it as a heading level 2. (Click the formatting dropdown menu on the left-hand edge of the editor toolbar. By default it formats your text as 'Paragraph'.)
5. Deselect the text and then press 'Enter' to start a new line.
6. Follow the steps below to add a JIRA macro to your page, showing a dynamic list of issues drawn from the **'Dragons'** project on your JIRA site.

- Click the **'Insert JIRA Issue'** icon  in the editor toolbar.
- The 'Insert JIRA Issue' popup window will appear. Click **'Search'** in the left-hand panel.
- Enter the following JQL (JIRA Query Language) into the search box: `project = DRA`.
- Click **'Search'**.
- A list of issues will appear, matching your search query. Select the option to **'Insert all query results as a table'**.
- Click **'Insert'**.
- The JIRA macro will appear on your page. In edit mode it looks like this:

```
{jira:project = DRA|server=Your Company JIRA}
```

7. Click the **'Preview'** tab to see a preview of the page. You will see a list of the JIRA issues in your **'Dragons'** project.
8. Click the **'Rich Text'** tab to return to the editor.
9. Enter the following text onto the page, and format it as a heading level 2: Reporting a new issue.
10. Start a new line.
11. Follow the steps below to add a new issue into JIRA and display the issue on your Confluence page:

- Click the **'Insert JIRA Issue'** icon  in the editor toolbar.
- The 'Insert JIRA Issue' popup window will appear. Click **'Create New Issue'** in the left-hand panel.
- Enter the following information:
 - Project: Dragons
 - Issue Type: Bug
 - Summary: The purple dragon melted my chocolate
 - Version/s: 2.0.S2
 - Description: The dragon tried to eat my chocolate and flamed it by mistake.
- Click **'Insert'**.
- You now have a new issue **DRA-4** in JIRA. In addition, a new JIRA macro will appear on your Confluence page, showing only the issue that you have created in JIRA. In edit mode the macro looks like this:

```
{jira:DRA-4|server=Your Company JIRA}
```

12. Save the page.

Screenshot 4 (click to enlarge): Your updated Dragons home page in Confluence



Problems? Please [raise a support ticket](#) for the product you're stuck on, or try the [Dragon Slayers' Forum](#).
Victory? Please continue.

Step 8. Add a Confluence Gadget to JIRA

Now you will add the Confluence 'Activity Stream' gadget to your JIRA [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 'activity' 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. You will see two **Activity Stream** gadgets, once for JIRA and one for Confluence. To find the Confluence one, look at the **gadget URL** and find the URL that contains port **'8090'**.
5. Click **Add it Now** under the appropriate gadget. 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. Click **Finished** to go back to your dashboard.
7. Configure the **Activity Stream** gadget:
 - Enter **Confluence Dragons Activity** in the **Title** field.
 - Select **Dragons** in the **Projects** field.
 - Click the dropdown arrow next to **Refresh Interval** and select **Every 15 Minutes**.
 - Leave the other fields at their default values and click **Save**.
8. Re-arrange your dashboard:
 - Drag the **Projects** gadget to the right and drop it under the **Assigned to Me** gadget.
 - Drag the **Agile Gadget** to the bottom right.
9. Choose a different colour for your **Activity Stream** gadget:
 - Move your cursor pointer over the gadget and click the downward-pointing arrow at top right of the gadget frame.
 - Select the **red** square in the row of colours.

Problems? Please raise a support ticket for the product you're stuck on, or try the [Dragon Slayers' Forum](#).

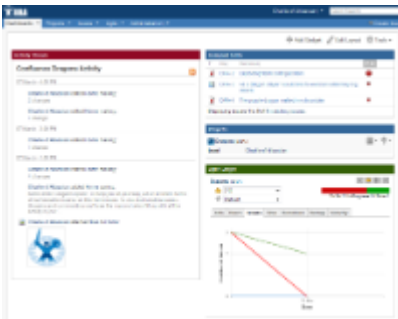
Victory? Please continue.

Victory!

Your JIRA dashboard now has 4 gadgets:

- The Confluence 'Activity Stream' gadget
- The 'Assigned to Me' gadget
- The 'Projects' gadget
- The GreenHopper 'Agile Gadget'

Screenshot 5 (click to enlarge): JIRA dashboard with 4 gadgets



Problems? Please raise a support ticket for the product you're stuck on, or try 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 and Crucible



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 and **Crucible** for piercingly insightful code reviews. Prepare to be blown away by FishEye/Crucible's integration with JIRA, Crowd and Bamboo.

Time estimate: This stage will take approximately **40 minutes**.

On this page:

- Step 1. Create your FishEye Database in PostgreSQL
- Step 2. Install FishEye and Crucible
- Step 3. Set Up FishEye/Crucible and Connect to PostgreSQL Database
- Step 4. Hook FishEye/Crucible up to Crowd
- Step 5. Connect FishEye to Subversion
- 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. Crucible will use this database as well. We are assuming that you have already created your PostgreSQL database server in [Dragons Stage 1](#).

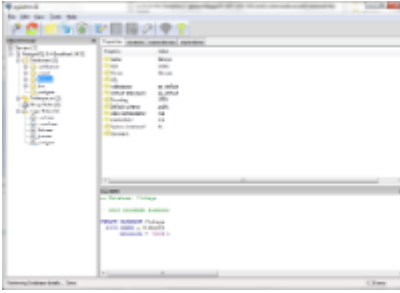
i We are using **pgAdmin III**, the administration user interface supplied with PostgreSQL. If you used the one-click installer in [Dragons Stage 1](#), **pgAdmin III** will be already installed on your computer.

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/8.4/bin/createuser -S -d -r -P -E fishuser
# Create the FishEye database:
/opt/PostgreSQL/8.4/bin/createdb --owner fishuser --encoding utf8 fisheye
exit
```

[Screenshot 1 \(click to enlarge\): FishEye/Crucible database and user in PostgreSQL](#)



Problems? Please raise a support ticket for the product you're stuck on, or try the [Dragon Slayers' Forum](#).

Victory? Please continue.

Step 2. Install FishEye and Crucible

Requirements: **FishEye/Crucible 2.5.2.**

▼ For Windows: (click to expand)

1. Go to the Atlassian [download centre](#).
2. Download the **'FishEye 2.5.2'** zip archive. This archive actually includes Crucible 2.5.2 as well.
3. Unpack the zip archive into a directory of your choice, avoiding spaces in the directory name. For example: `c:\fishyecrucible`. We will now refer to this location as the FishEye/Crucible installation directory.
4. Now you will create another directory where FishEye/Crucible will store local data, separate from the installation directory:
 - Create the new directory, e.g. `C:\data\fishyecrucible`.
 - Create an environment variable called 'FECRU_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 'FECRU_INST' and a value of your new directory's location of e.g. `C:\data\fishyecrucible`.)
 - Copy the `config.xml` file from the root of your FishEye/Crucible installation directory to the root of your new FECRU_INST directory.
5. Now you will make your PostgreSQL driver available to FishEye/Crucible:
 - Create a `\lib` directory as a sub-directory of your new FECRU_INST directory
 - Copy the PostgreSQL JDBC driver JAR (downloaded in [Dragons Stage 1](#)) to the new `\lib` directory.
6. Start FishEye/Crucible from the command line by running `bin\run.bat` from your FishEye/Crucible 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 2.5.2'** zip archive. This archive actually includes Crucible 2.5.2 as well.
3. Unpack the zip archive into a directory of your choice, avoiding spaces in the directory name. For example: `/opt/feclu-x.x.x`. We will now refer to this location as the FishEye/Crucible installation directory.
4. Now you will create another directory where FishEye/Crucible will store local data, separate from the installation directory:
 - Create the new directory, e.g. `/opt/fishyecrucible`.
 - Create an environment variable called 'FECRU_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 FECRU_INST directory.
5. Now you will make your PostgreSQL driver available to FishEye/Crucible:
 - Create a `/lib` directory as a sub-directory of your new FECRU_INST directory.
 - Copy the PostgreSQL JDBC driver JAR (downloaded in [Dragons Stage 1](#)) to the new `/lib` directory.
6. Start FishEye/Crucible from the command line by running `./bin/run.sh` from your FishEye/Crucible 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)'**.

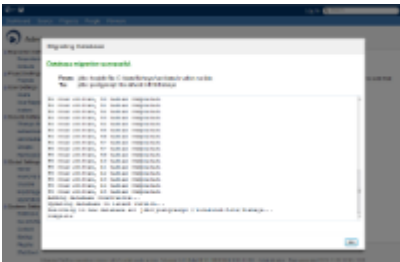
Problems? Please raise a support ticket for the product you're stuck on, or try the [Dragon Slayers' Forum](#).

Victory? Please continue.

Step 3. Set Up FishEye/Crucible and Connect to PostgreSQL Database

1. To access FishEye/Crucible, go to your web browser and type this address: `http://localhost:8060/` (or type the host name or IP address instead of `localhost`).
2. The FishEye/Crucible set-up wizard will start.
 - If you already have a license key, click **'Enter existing license'**. Enter your FishEye license into the **'FishEye License Key'** field and your Crucible license into the **'Crucible License Key'** field.
 - If you don't have a license key, click **'Obtain evaluation license'** and follow the instructions on screen. Ensure that you leave the **'Yes! Please include Crucible as part of this evaluation.'** checkbox selected during the process.
3. Choose an administration password, enter it and then enter it again to confirm it.
4. Click **'Next'**.
5. Click **'Add repository'** to exit the wizard.
6. Log in to FishEye/Crucible when prompted.
7. In the left-hand **'Admin'** menu, click **'Database'** under **Systems Settings**.
8. Click **'Edit'** and enter the following details:
 - **Type** – Select **'PostgreSQL'** from the **'Type'** dropdown list.
 - **Driver Location** – Select **'User Supplied - FISHEYE_INST/lib'**.
 - **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/Crucible can log in to the database. If this fails, verify that you have the PostgreSQL JDBC driver JAR file in the `FECRU_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/Crucible 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'**.

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



Problems? Please raise a support ticket for the product you're stuck on, or try the [Dragon Slayers' Forum](#).

Victory? Please continue.

Step 4. Hook FishEye/Crucible up to Crowd

Follow the steps below to hook FishEye and Crucible 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`. Do not close your FishEye/Crucible browser/tab.
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:

Note: You only need to configure a FishEye application in Crowd. Crucible will share the authentication mechanism and integration that you set up with FishEye and Crowd.

 - Application Type: FishEye
 - Name: `fisheye`
 - Description: Atlassian FishEye
 - Password – Enter a password that FishEye/Crucible will use to access Crowd and enter it again to confirm it.
 - URL – Enter the base URL of your FishEye/Crucible 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/Crucible host name, excluding the "http://www." prefix and the ":8060" port number. e.g. `foobar.com`.
 - If it's not already present, add: `127.0.0.1`.
7. Go back to your the FishEye/Crucible Admin screens in your FishEye/Crucible browser window/tab.
8. In the left-hand **'Admin'** menu, click **'Authentication'** under **Security Settings**.
9. The **'Authentication Settings'** screen will appear. Select **'Setup Crowd authentication'**.
10. 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.
11. Click **'Apply'**.
12. Now you will grant `charlie` administrator rights in your FishEye instance.
13. Log in to FishEye using the Crowd user you set up in Stage 1, i.e. `'charlie'`.
14. Click the username (i.e. `'Charlie of Atlassian'`) then **'Administration'** from the dropdown menu, to go to the FishEye Admin screens.
15. In the left-hand **'Admin'** menu, click **'Administrators'** under **Security Settings**.
16. The **'Manage Admin Users and Groups'** screen will appear. Select `charlie` under **'Non-Admin Users'** and click **'Add >>'** to grant `charlie` administrator rights.

Full details are in the [Crowd documentation](#).


Problems? Please [raise a support ticket](#) for the product you're stuck on, or try the [Dragon Slayers' Forum](#).


Victory? Please continue.

Step 5. Connect FishEye to Subversion

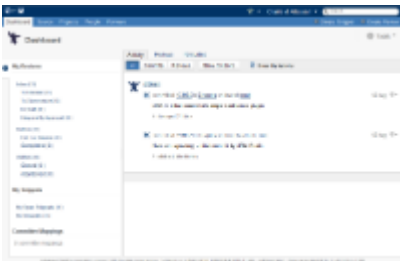
For the purposes of this integration exercise, we have provided a read-only Subversion repository that you can connect to your FishEye 'Dragons' repository. We recommend this repository because:

- 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. In the left-hand **'Admin'** menu, click **'Repositories'** under **Repository Settings**.
 2. Click the **'Add...'** button. The 'New Repository - Page 1 of 3' screen will appear.
 3. Enter the following information:
 - Repository Type: Subversion.
 - Name: Dragons.
 - Description: Dragons repository.
 4. Click **'Next'** and enter the following information:
 - SVN URL: <https://studio.plugins.atlassian.com/svn/>.
 - Path: DRA.
 - Username and Password – Not required for our sample repository, because the repository allows anonymous access.
 5. Click **'Next'** and enter the following information:
 - Store Diff Info – Select this checkbox.
 - Enable immediately – Select this checkbox.
 6. Click **'Test Connection'** to verify that Subversion is properly connected to FishEye. Click **'Close'**.
 7. Click **'Add'**. The **'Repositories'** page will display the 'Dragons' repository.
 8. Click the **'Source'** tab at the top of the screen.
 9. Click the star symbol next to the **'Dragons'** repository to select it as a favourite.
 10. Click the FishEye logo at top left of the screen 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 [raise a support ticket](#) for the product you're stuck on, or try 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.

 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'**.

 Click the changeset number (153935) from the activity stream 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 [raise a support ticket](#) for the product you're stuck on, or try 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-FishEye Application Link
- Step 2. Link a FishEye Repository to a JIRA Project and Subscribe JIRA to FishEye Gadgets
- Step 3. Configure the FishEye Plugin in JIRA
- Step 4. Add a FishEye Gadget to JIRA
- Victory!

Step 1. Add a JIRA-FishEye Application Link

In this step you will link your JIRA and FishEye servers by creating an application link.

1. Go to your FishEye URL in your web browser, e.g. <http://localhost:8060/>.
2. Click the username (i.e. 'Charlie of Atlassian') then '**Administration**' from the dropdown menu, to go to the FishEye Admin screens.
3. Click '**Application Links**' in the left-hand panel.
4. The 'Configure Application Links' screen will appear. Click '**Add Application Link**'.
5. The first screen of the 'Add Application Link' wizard will appear. Copy the base URL for your JIRA site (e.g. <http://coopers:8080> or <http://www.foobar.com:8080>) and paste it into the '**Server URL**' field.
6. Click '**Next**'.
7. The 'Link to JIRA' screen will appear. Enter the following information:
 - Create a link back to this server – This option is selected by default. Let it remain selected.
 - Username: `charlie`. This is the username of the administrator on your JIRA site.
 - Password – Enter Charlie's password in JIRA.
 - Reciprocal Link URL – Leave this field at its default value, pointing to your FishEye site.
8. Click '**Next**'.
9. The 'Set Users and Trust' screen will appear. Enter the following information:
 - The servers have the same set of users – This option is selected by default. Let it remain selected.
 - These servers fully trust each other – This option is selected by default. Let it remain selected.
10. Click '**Create**'. The application link will be created and displayed on the 'Configure Application Links' page.
11. Click '**JIRA Settings**' next to the '**Your Company JIRA**' application link.
12. Change the '**Application Name**' to `Dragons JIRA` and click '**Update**'.
13. Click '**JIRA Settings**' next to the '**Dragons JIRA**' application link.
14. The 'Update JIRA Server' screen will appear. Select the '**Include in Activity Streams**' checkbox.
15. Click '**Save**'.
16. Click the '**Server**' link in the left-hand menu under 'Global Settings'.
17. The 'Server Settings' screen will appear. Click '**Edit Settings**'.
18. The 'Edit Web Settings' screen will appear. Select the '' radio button next to '**Allow remote API calls**'.
19. Click '**Update**'.

Screenshot 1 (click to enlarge): Adding JIRA as a trusted application in FishEye



Step 2. Link a FishEye Repository to a JIRA Project and Subscribe JIRA to FishEye Gadgets

In this step you link your FishEye repository to your JIRA project via an "entity link". This will enable the hyperlink on JIRA issue keys in FishEye and the summary popup window that appears when you move your cursor over a JIRA issue key. You will also make your FishEye gadgets available for use in JIRA.

1. Go to your FishEye URL in your web browser, e.g. <http://localhost:8060/>.
 2. Click the '**Repositories**' link in the left-hand menu under 'Repository Settings'.
 3. The 'Repositories' screen will appear. Click the cog icon next to your '**Dragons**' repository and click '**Application Links**' from the menu that appears.
 4. Click '**Add Link**' then click '**Dragons JIRA (JIRA)**' from the dropdown menu.
 5. Click '**Create**'. The FishEye repository-JIRA project link will be created.
 6. Now you will make your FishEye gadgets available in JIRA, so that JIRA users will be able to add any FishEye gadget to their dashboards. Go to your JIRA browser window and click '**Dashboards**' in JIRA's top navigation bar.
 7. Click '**Add Gadget**'.
 8. The 'Gadget Directory' popup window will appear. Click '**Gadget Subscriptions**'.
 9. The 'Gadget Subscriptions' popup window will appear. Click '**Add Subscription**'.
 10. The 'Add Subscriptions' popup window will appear. Copy the base URL for your FishEye site (e.g. <http://coopers:8060> or <http://www.foobar.com:8060>) and paste it into the text box on the screen.
 11. Click '**Add Subscription**'.
 12. Click '**Finished**'.
- ✔ The FishEye gadgets are now available in your JIRA gadget directory. You have not yet added them to your JIRA dashboard. We will do that in a later step.

Step 3. 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.foo.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 **'Edit Primary Configuration'**.
5. Enter the following information:
 - Enable Crucible Integration: **Yes**
 - Review Search Method: **Both**
6. Leave all the other fields at their default values and click **'Update'**.
7. Click **'Application Links Configuration'** at the bottom of the screen.
8. Click **'Trusted Applications'** in the **'Outgoing Authentication'** column for your FishEye/Crucible link.
9. Click **'Modify'** and enter the following information:
 - **'IP Patterns'**: Enter the IP addresses for your FishEye/Crucible instance (separated by commas), e.g. `127.0.0.1, 172.20.5.95`.
You can find the IP addresses for your FishEye/Crucible instance by viewing the 'Remote Addresses' tab for the application in Crowd.
10. Click **'Apply'**.
11. Click **'Incoming Authentication'** in the left menu.
12. Click **'Modify'** and enter the following information:
 - **'IP Patterns'**: Enter the IP addresses for your JIRA instance (one per line), e.g. `127.0.0.1, 172.20.5.95`.
 - **'URL Patterns'**: Enter the following paths (one per line): `/plugins/servlet/streams, /sr/jira.issueviews:searchrequest, /secure/RunPortlet, /rest, /rpc/soap`
13. Click **'Apply'** and then **'Close'**.


Try It Out!

- 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 2 (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, provided the changes were committed in the last 30 days.

 We committed our changes to the 'Dragons' repository more than 30 days ago, so you will not see any commits on the JIRA project tab. The screenshot below shows you what they would look like.

Screenshot 3 (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. To try it, go to the 'Source' tab on your **'DRA-1'** issue and click the changeset number ('1') or the repository name ('Dragons').
- In FishEye, you can see a popup summary of issue information for an issue key. Try it by moving your cursor over **'DRA-1'** in your FishEye view. You can also click through from FishEye to JIRA by clicking the issue key.

Full details are in the [JIRA documentation](#).

Problems? Please raise a support ticket for the product you're stuck on, or try the [Dragon Slayers' Forum](#).

Victory? Please continue.

Step 4. 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. Choose a different colour for your **FishEye Charts** 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 raise a support ticket for the product you're stuck on, or try the [Dragon Slayers' Forum](#).

Victory? Please continue.

Victory!

Your JIRA dashboard now has 5 gadgets:

- The 'FishEye Charts' gadget
- The Confluence 'Activity Stream' gadget
- The 'Assigned to Me' gadget
- The 'Projects' gadget
- The GreenHopper 'Agile' gadget

Screenshot 3 (click to enlarge): JIRA dashboard with 5 gadgets



Problems? Please raise a support ticket for the product you're stuck on, or try the [Dragon Slayers' Forum](#).

Victory? Please continue.



Grab a Bigger Sword and Move to the Next Stage

- **Tweet?** Tweet.
- Go to [Dragons Stage 7 - Get JIRA and Crucible Talking](#).

Dragons Stage 7 - Get JIRA and Crucible Talking



You are embarking on stage 7 of the *Atlassian Dragon Quest*. Remember, the most important piece of equipment is your shield. If you must make a choice between a sword and a shield, take the shield!

In this stage you will configure JIRA and Crucible, so that you will be able to create a review, add comment to the review and then create a JIRA issue from that review comment. You'll also add a Crucible gadget to your JIRA dashboard.

Time estimate: This stage will take approximately **45 minutes**.

On this page:

- Step 1. Enable Sub-Tasks for Crucible
- Step 2. Create a Crucible Project and Link Your Crucible Project to Your JIRA Project
- Step 3. Create a Review
- Step 4. Create a JIRA Issue from a Crucible Review
- Step 5. Add a Crucible Gadget to JIRA
- Victory!

Step 1. Enable Sub-Tasks for Crucible

In this step, you will configure your FishEye/Crucible-JIRA application link to enable sub-tasks for Crucible. This is required for creating JIRA issues from Crucible reviews.

1. Go to your FishEye/Crucible URL in your web browser, e.g. <http://localhost:8060/>.
2. Click the username (i.e. 'Charlie of Atlassian') then '**Administration**' from the dropdown menu, to go to the FishEye/Crucible Admin screens.
3. Click '**Application Links**' in the left-hand panel.
4. Click '**JIRA Settings**' next to your 'Dragons JIRA' application link.
5. The 'Update JIRA settings' screen will appear. Select the '**Use Basic Authentication**' checkbox and enter the login details for [charlie](#).
6. Click the '**Test**' button in the 'Subtask Settings'. The page will refresh, however the 'Subtask Settings' section will display different controls. Enter the following information:
 - Subtask Type – Select '**Technical task**'.
 - Subtask Resolution Action ID: 2
 - Subtask Resolution – Leave unchanged.
 - Allow Unassigned – Select '**Yes**'.
7. Click '**Save**'.

Screenshot 1 (click to enlarge): Crucible-JIRA application link with sub-tasks configured

Update JIRA Server

Server Details
Name: Your Company JIRA

Authentication
 Use Basic Authentication
Username:
Password:

Options
 Include in Activity Streams
 Allow Time Tracking submission from reviews

Subtask Settings
Leave these fields blank to disable subtasks.
Subtask Type ID:
Subtask Resolution Action ID:
Subtask Resolution ID:
Allow Unassigned: Yes No
Pressing 'Test' will retrieve subtask types and resolutions from JIRA.

Full details are in the [Crucible documentation](#).

Problems? Please [raise a support ticket](#) for the product you're stuck on, or try the [Dragon Slayers' Forum](#).

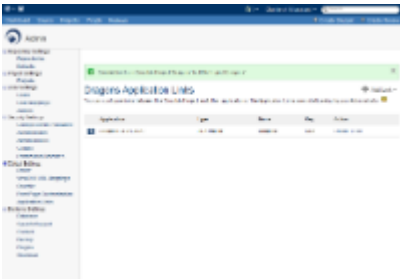
Victory? Please continue.

Step 2. Create a Crucible Project and Link Your Crucible Project to Your JIRA Project

Next, you will create a Crucible project and link it to your JIRA project.

1. Go to your FishEye/Crucible URL in your web browser, e.g. <http://localhost:8060/>.
2. Click the username (i.e. 'Charlie of Atlassian') then '**Administration**' from the dropdown menu, to go to the FishEye/Crucible Admin screens.
3. Click '**Projects**' in the left-hand panel.
4. The 'Projects' screen will appear. Click '**Create a New Project**'.
5. The '**Edit Project**' screen will appear. Enter the following information:
 - Name: `Dragons`
 - Key: `DRA`
 - By default, allow anyone to join reviews after creation – Select this checkbox.
6. Leave all other fields at their default values and click '**Save**'.
7. The 'Projects' screen will appear again, displaying your 'Dragons' project.
8. Click '**Application Links**' in the '**Operations**' column next to your 'Dragons' project.
9. The 'Dragons Application Links' screen will appear. Click '**Add Link**' and click '**Dragons JIRA (JIRA)**' in the dropdown menu that appears.
10. Click '**Create**'.

Screenshot 2 (click to enlarge): Crucible Dragons Project linked to JIRA Dragons Project



Full details are in the [Crucible documentation](#).

Problems? Please [raise a support ticket](#) for the product you're stuck on, or try the [Dragon Slayers' Forum](#).

Victory? Please continue.

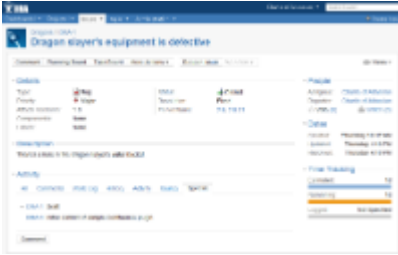
Step 3. Create a Review

In this step, you will create a Crucible review from the dashboard activity stream and create a comment in that review.

1. Click the '**Dashboard**' tab in FishEye/Crucible.
2. Click the '**Activity**' tab on the dashboard, if it is not already active, to display the activity stream.
3. Find changeset '**153936**' committed by '**drosen**' in the activity stream. Click the cog icon next to the date for the commit and click '**Create Review**' in the menu that appears.
4. The 'Edit Review Details' screen will appear. Click '**Start Review**'.
5. Click '**Confirm**' at the warning screen. The 'DRA-1' review screen will appear.
6. Click '**ExampleMacro.java**' under '**trunk/src/main/java/com/atlassian/dragons**' in the file tree on the left. The code for this file will appear in the right panel.
7. Scroll down to line 21 in the code and click the '**21**'.
8. A text area will open for you to enter a comment. Add the comment as follows:
 - Enter the following text in the text area: `This comment does not mention dragons. Please amend the text.`
 - Defect – Select this checkbox
9. Click '**Post**' to post your review comment. The review comment will appear.

✔ You can now view your review in the 'Reviews' tab on your 'DRA-1' JIRA issue. Open your 'DRA-1' issue and click the 'Reviews' tab. The tab shows the reviews related to the issue, i.e. the reviews involving changesets related to the issue.

Screenshot 3 (click to enlarge): Reviews tab on a JIRA issue

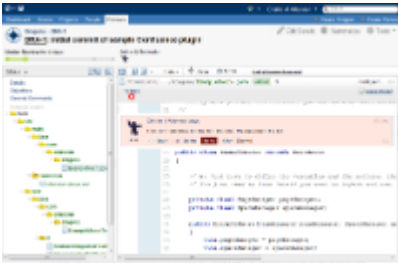


Step 4. Create a JIRA Issue from a Crucible Review

Next, you will reopen JIRA issue DRA-1 so that you can create subtasks for it, then create the JIRA subtask from the comment in your Crucible review, and resolve the issue via Crucible.

1. Go to your JIRA URL in your web browser, e.g. <http://localhost:8080/>.
2. Click the arrow next to **Issues** in the top navigation bar. Click the 'DRA-1' issue under the 'Recent Issues' in the dropdown that appears.
3. Click **Reopen issue**.
4. The 'Reopen issue' screen will appear. Click **Reopen issue**.
5. Go to your FishEye/Crucible URL in your web browser, e.g. <http://localhost:8060/>.
6. The FishEye/Crucible dashboard will appear. Click **Re: DRA-1** in the **Re: DRA-1 commented on review** text in the activity stream.
7. Click **This comment does not mention dragons** in the **Latest Comment** column on the review details screen.
8. The review comment will appear. Click the **Create Issue** link in the comment panel.
9. The **Create Issue** panel will appear. Leave the **Summary** and click the **Assign to Me**.
10. Click **Create**. The page will refresh. The key of your new issue (DRA-4) will be displayed in the review comment with a status of 'Open'.
11. Click the issue key, **DRA-4**. The issue will be displayed in JIRA on your page.
12. Click the back button of your browser to view your review again.
13. Click the **Resolve** link next to the issue key in your review comment. The status of the issue in the review comment will change to 'Closed'.
14. Click the issue key, **DRA-4**. The issue will be displayed in JIRA on your page with a status of 'Closed'.

Screenshot 4 (click to enlarge): Closed JIRA issue displayed in a Review comment



Step 5. Add a Crucible Gadget to JIRA

Now you will add the 'Crucible Charts' gadget to your [Dragon Development Dashboard](#).

1. Go to your JIRA URL in your web browser, e.g. <http://localhost: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 'Crucible' 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 **Crucible 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.
6. Click **Finished** to go back to your dashboard.
7. Configure the **Crucible Charts** gadget:
 - Enter **DRA** in the **Crucible Project Key** field.
 - Click the dropdown arrow next to **Refresh Interval** and select **Every 15 Minutes**.
 - Click **Save**.
8. Choose a different colour for your **Crucible Charts** gadget:
 - Move your cursor pointer over the gadget and click the downward-pointing arrow at top right of the gadget frame.
 - Select the **dark blue** square in the row of colours.

Problems? Please [raise a support ticket](#) for the product you're stuck on, or try the [Dragon Slayers' Forum](#).

Victory? Please continue.

Victory!

Your JIRA dashboard now has 6 gadgets:

- The 'Crucible Charts' gadget
- The 'FishEye Charts' gadget
- The Confluence 'Activity Stream' gadget
- The 'Assigned to Me' gadget
- The 'Projects' gadget
- The GreenHopper 'Agile' gadget

Screenshot 5 (click to enlarge): JIRA dashboard with 6 gadgets



Problems? Please raise a support ticket for the product you're stuck on, or try the [Dragon Slayers' Forum](#).

Victory? Please continue.



Grab a Bigger Sword and Move to the Next Stage

- **Tweet?** Tweet.
- Go to [Dragons Stage 8 - Install Bamboo](#).

Dragons Stage 8 - Install Bamboo



You are embarking on stage 8 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](#).

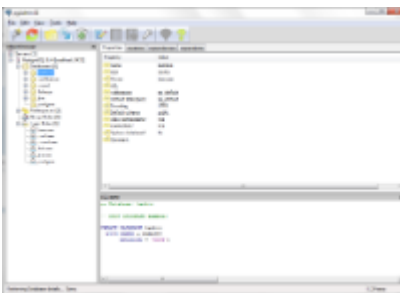
 We are using **pgAdmin III**, the administration user interface supplied with PostgreSQL. If you used the one-click installer in [Dragons Stage 1](#), pgAdmin III will be already installed on your computer.

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/8.4/bin/createuser -S -d -r -P -E bamuser
# Create the Bamboo database:
/opt/PostgreSQL/8.4/bin/createdb -O bamuser bamboo
exit
```

[Screenshot 1 \(click to enlarge\): Bamboo database and user in PostgreSQL](#)

**Step 2. Install Bamboo**

Requirements: **Bamboo 3.0.1**.

▼ For Windows: (click to expand)

1. Go to the Atlassian [download centre](#).
2. Download the '**Standalone (Windows Installer)**' file for **Bamboo 3.0.1**.
3. Launch the Bamboo Windows installer (`atlassian-bamboo-3.0.1-standalone-windows-x32.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. Install Bamboo as a Windows Service, so that it starts each time you start Windows by running `{BAMBOO_INSTALL}\InstallAsService.bat`. If you are running Bamboo in Windows Vista or Windows 7, you may need to run this file in administrative mode by right clicking it and selecting 'Run as administrator'.
6. Start your Bamboo server by running `{BAMBOO_INSTALL}\StartBamboo.bat`. You may also need to run this file in administrative mode as described in the previous step.

▼ 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 3.0.1**.
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}/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. Start your Bamboo server by running `{BAMBOO_INSTALL}/bamboo.sh start`.


Problems? Please [raise a support ticket](#) for the product you're stuck on, or try 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.

The instructions below assume that you already have a build tool set up. 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.

1. Set up your Maven 2 environment:
 - If you do not yet have Maven 2 installed, we recommend that you download and install the *Atlassian Plugin SDK* (note, you do not need to configure an IDE). The SDK includes Maven 2 and a correctly-configured Maven `settings.xml` file, as well as a number of shell scripts to speed up and simplify plugin development. It also includes the Java Activation and other JARs that you will need for a successful Maven build.
 - If you already have Maven 2, please ensure that you have the required additional JARs. See the [FAQ](#) for information on downloading these JARs.
2. To access Bamboo, go to your web browser and type this address: <http://localhost:8085/>.
3. The Bamboo Setup Wizard will start up, to guide you through the process of setting up your Bamboo server and creating an administration user.
 - Enter your 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.
 - Choose the '**Custom Installation**' setup method.
4. Detailed instructions on the custom installation setup method are in the [Bamboo documentation](#). Below are the things you need to know for our Dragon Quest. Enter the '**Bamboo Configuration**' information as follows:
 - 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.fooobar.com:8085`.
 - 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`.
5. Choose `External Database` for your database configuration and ensure that `PostgreSQL 8.2` and above is selected in the dropdown menu.
6. Enter the following information to connect to your Bamboo database created in step 1 above:
 - 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.
7. For your '**Starting Data**', select '**Create new Bamboo home**'.
8. 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.
9. Click '**Finish**'.
 You can now see the Bamboo home page.
10. Now you will check that your Bamboo configuration includes your default builder and JDK. Click '**Administration**' in the top navigation bar.
11. The 'Bamboo Administration' screen will appear. Click '**Builders**' in the left-hand menu.
12. The 'Builders' screen will appear. Look through the list on the left, to see if your build tool is included along with the default tools of '**Script**' and '**Bash**'. For this integration exercise, we assume that you are using Maven 2. In that case, you should see '**Maven 2**' listed in the tabs on the left.
13. If your builder is not included, click '**Add builder as a server capabilities**' near the top of the page. The 'Add Capability' panel will appear. Enter the following information:
 - Capability Type: `Builder`.
 - Type: `Maven 2.x`.
 - 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).
If you have installed the Atlassian PDK, Maven can be found in a sub-directory under your Atlassian PDK installation directory. For example, `C:\Atlassian\atlassian-plugin-sdk-3.2\apache-maven` (Windows) or `/usr/local/Atlassian/atlassian-plugin-sdk-3.2/apache-maven` (UNIX).
 - Click '**Add**'
14. Check that your Bamboo configuration includes your JDK. Click '**JDKs**' in the left-hand menu.
15. The 'JDKs' screen will appear. Look through the tabs on the left, to check that your JDK is included. 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 tab on the left '**JDK 1.6.x_xx (JRE)**', as well as a '**JDK**' and a '**JDK 1.6**' tab. Ensure that the Java Home is pointing to your JDK directory, not your JRE directory.
16. If your JDK is not included, click '**Add JDK as a server capabilities**'. The 'Add Capability' panel will appear. Enter the following information.
 - 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).
 - Click '**Add**':

Screenshot 2 (click to enlarge): Bamboo home page



Problems? Please [raise a support ticket](#) for the product you're stuck on, or try 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, excluding the "http://www." prefix and the ":8085" port number. e.g. `foobar.com`.
 - 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`. You may need to run this program in administrative mode as described above.)
8. Remove the following file from your Bamboo installation folder: `{BAMBOO_INSTALL}/webapp/WEB-INF/lib/crowd-integration-client-2.0.7.jar`
9. Copy the Crowd client libraries and configuration files to your Bamboo installation folder:
 - Copy `{CROWD_INSTALL}/client/crowd-integration-client-2.2.2.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 out all other lines of code. The code below should be the only lines of uncommented code in your file, after you have finished making these changes:



```
<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.v22.BambooAuthenticator" />
```

13. Start your Bamboo server again, and go to your Bamboo URL in your browser, e.g. `http://www.foobar.com:8085`.
 -  If you experience problems with the Windows service you can start Bamboo in a console window instead, by running `{BAMBOO_INSTALL}\bin\BambooConsole.bat`.
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 [raise a support ticket](#) for the product you're stuck on, or try 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 [raise a support ticket](#) for the product you're stuck on, or try 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 Plan' screen will appear. Click **Create a New Plan**.
3. The 'Create a New Plan' screen will appear. Enter the following information in the 'Plan Configuration' section:
 - Project Name: Dragons.
 - Project Key: DRAG.
 - Build Plan Name: Main
 - Build Plan Key: MAIN
 - Repository: Subversion.
 - Repository URL: <https://studio.plugins.atlassian.com/svn/DRA/trunk>.
 - 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'.
4. Leave the rest of the fields in the 'Plan Configuration' section at their default values.
5. Enter the following information in the 'Job Configuration' section:
 - Builder – Select your build tool, e.g. Maven 2.
 - Goal – Change `clean test` to `clean`.
 - Build JDK – Select your JDK version, e.g. JDK 1.6.
 - The build will produce test results – Uncheck this checkbox.
6. Leave the rest of the fields at their default values and click **Create**.
7. Bamboo will immediately start a build, based on the plan that you have just created. The build may take a few minutes to complete.
8. When the build has finished, the 'Plan Summary' for the 'Main' plan in the 'Dragons' will be displayed. The 'Recent History' will have a link to build result, [DRAG-MAIN-1](#). 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



Full details on creating a plan are in the [Bamboo documentation](#).

Problems? Please [raise a support ticket](#) for the product you're stuck on, or try the [Dragon Slayers' Forum](#).

Victory? Please continue.

Victory!

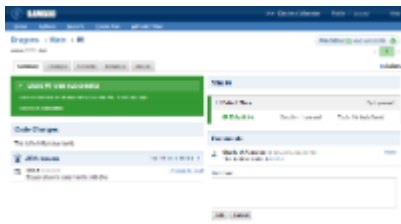
Your Bamboo, FishEye and JIRA servers are fully integrated. Here are some of the highlights for you to try.

✔ 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:

- On the Bamboo dashboard, click **#1** in the **'Recent History'** to open the build result summary.
- Click **'Comment'**.
- Add the following comment: `This build is related to DRAG-1.`
- Click **'Add'**.

✔ 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



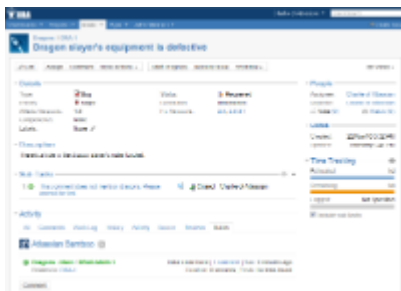
✔ Click the **'Issues'** tab to see the JIRA issues for a build result.

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 raise a support ticket for the product you're stuck on, or try the [Dragon Slayers' Forum](#).
Victory? Please continue.



Grab a Bigger Shield and Go Conquer that Dragon

- **Tweet?** [Tweet](#).
- Go to [Dragons Stage 9 - Bamboo Gadgets and JIRA Victory](#).

Dragons Stage 9 - Bamboo Gadgets and JIRA Victory



You're nearly there. Stage 9 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 and Subscribe JIRA to Bamboo Gadgets](#)
- [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 and Subscribe JIRA to Bamboo Gadgets

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**'.
7. Now you will make your Bamboo gadgets available in JIRA, so that JIRA users will be able to add any Bamboo gadget to their dashboards. Go to your JIRA browser window and click '**Dashboards**' in JIRA's top navigation bar.
8. Click '**Add Gadget**'.
9. The 'Gadget Directory' popup window will appear. Click '**Gadget Subscriptions**'.
10. The 'Gadget Subscriptions' popup window will appear. Click '**Add Subscription**'.
11. The 'Add Subscriptions' popup window will appear. Copy the base URL for your Bamboo site (e.g. <http://coopers:8085> or <http://www.foobar.com:8085>) and paste it into the text box on the screen.
12. Click '**Add Subscription**'.
13. Click '**Finished**'.

✔ The Bamboo gadgets are now available in your JIRA gadget directory. You have not yet added them to your JIRA dashboard. We will do that in a later step.

Problems? Please [raise a support ticket](#) for the product you're stuck on, or try 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.foo.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.
 - Uncheck **'Use my favourite plans'**. Enter 'dra' in the textbox and select **'Dragons - All Plans'** in dropdown menu that opens.
 - 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. Re-arrange your dashboard by dragging the **'Bamboo Plans'** gadget to the bottom right of your dashboard. Drag the **'Bamboo Plan Summary Chart'** gadget to the bottom right too. (This is *optional*, just to make everything fit into the screenshot you will take later, when you claim your Atlassian DragonSlayer T-shirt.)
11. 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.
12. Colour your **'Bamboo Plan Summary Chart'** gadget purple too.

Problems? Please [raise a support ticket](#) for the product you're stuck on, or try the [Dragon Slayers' Forum](#).

Victory? Please continue.

The Battle is Won, the Dragon is Slain

Your JIRA dashboard now has 8 gadgets:

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

[Screenshot 1 \(click to enlarge\): JIRA dashboard with 8 gadgets](#)



Problems? Please [raise a support ticket](#) for the product you're stuck on, or try 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](#).
- See the tips for life after Dragons.

After Dragons



Is there life after Dragons?

Now that you have successfully set up your Atlassian integrated suite, we have some useful information about what you may want to do next. There's no rush. Get to know the applications and show off your T-shirt for a while first. Then choose any of the points below that may be relevant to you.

On this page:

- [Using the Free IDE Connectors](#)
 - [Atlassian Connector for Eclipse](#)
 - [Atlassian Connector for IntelliJ IDEA](#)
- [Adding Another Atlassian Tool to your Suite](#)
 - [Clover for Code Coverage](#)
- [Hints after Initial Setup](#)
 - [Adding Users to your Atlassian Integrated Suite](#)
 - [Running Bamboo in a Console Window](#)

Using the Free IDE Connectors

This information is useful to developers who use [Eclipse](#) or [IntelliJ IDEA](#). You can work with JIRA issues, Bamboo builds and FishEye links directly within your IDE (integrated development environment), using the [Atlassian IDE Connectors](#). The connectors are **free**.

Atlassian Connector for Eclipse

Installation	<p>You can install the connector directly from the Eclipse software updates manager, or via the Mylyn Connector Discovery wizard, or from a zipped archive. Full instructions are in our installation guide. Here are the instructions for Eclipse 3.5 using the Mylyn Connector Discovery wizard:</p> <div style="border: 1px dashed #add8e6; padding: 10px; margin: 10px 0;"> <ol style="list-style-type: none"> 1. Ensure that you have already installed Mylyn 3.2.x. (If you are using an Eclipse package from the Eclipse download site, Mylyn 3.2 is already included in any package except the Classic download.) 2. In Eclipse Mylyn, open the Task Repositories view. (In Eclipse, click Windows, Show View, Other and select the Task Repositories view from the Tasks category.) 3. Click the Add Task Repository icon. 4. The Add Task Repository screen appears. Click the Install More Connectors button. 5. The Mylyn Connector Discovery screen appears. Select the Atlassian Connector and click Finish to install it. </div>
--------------	---

Overview	<p>Working with Bamboo builds in Eclipse, you can:</p> <ul style="list-style-type: none"> • View a list of the builds you are monitoring, in the Bamboo view in Eclipse. • Receive notification of failed builds and other build changes. • Open the Bamboo build details in an Eclipse editor. • Open the Bamboo build details in your web browser, displaying the Bamboo web interface. • Run a build on the Bamboo server. • View a Bamboo build log. • View test results. • View changed files in the build. • Comment on a Bamboo build. • Label a Bamboo build. • Add a new task based on a failed build. <p>Working with FishEye in Eclipse, you can open a file from Eclipse in FishEye and send your colleagues a FishEye link to your file.</p> <p>Working with JIRA issues in Eclipse:</p> <ul style="list-style-type: none"> • For information on setting up your JIRA server in Eclipse, please read the configuration guide. • Please refer to the JIRA Mylyn documentation for user guidelines.
Videos and tours	See our website .

Atlassian Connector for IntelliJ IDEA

Installation	<p>You can install the connector from the 'Plugins' menu in IntelliJ IDEA, as described in our installation guide. Here are the instructions in brief:</p> <div data-bbox="336 943 1414 1093" style="border: 1px dashed #ccc; background-color: #e6f2ff; padding: 10px;"> <ol style="list-style-type: none"> 1. Open the IDEA plugin manager. (Go to IDEA's 'File' menu and select 'Settings', 'IDE Settings', 'Plugins'.) 2. Right-click 'Atlassian Connector for IntelliJ IDEA' in the 'Available' plugins tab. 3. Select 'Download and Install'. </div>
Overview	<p>Working with Bamboo builds in IDEA, you can:</p> <ul style="list-style-type: none"> • Receive notifications of failed builds. • View the builds. • Re-run a build. • Open the Bamboo build details in an IDEA output tool window. • View the build history for a selected plan. • View a Bamboo build log. • View failed tests and stack traces. • Click a link in a stack trace to go directly to the code that failed. • Re-run a failed test. • View changed files. • Compare the build version of a file with your local version. • Compare the build version of a file with the previous repository version. • Open the repository version of a file in your IDEA editor. • Comment on a Bamboo build. • Label a Bamboo build. <p>Working with FishEye in IDEA, you can open a file from Eclipse in FishEye and send your colleagues a FishEye link to your file.</p> <p>Working with JIRA issues in IDEA, you can:</p> <ul style="list-style-type: none"> • View a filtered list of issues. • Make a JIRA issue your active issue. • Make a JIRA issue your active task • Create a new JIRA issue. • Comment on a JIRA issue and view existing comments. • Create a changelist from a JIRA issue. • Log work on a JIRA issue. • View a JIRA issue in an IDEA output tool window. • View stack traces from a JIRA issue and click through to the relevant source file. • View, download and upload attachments on an issue. • Assign an issue to yourself or another user. • Perform workflow actions on a selected issue. • Use the issue quick access options to open an issue in IDEA.

Videos and tours	See our website .
------------------	-----------------------------------

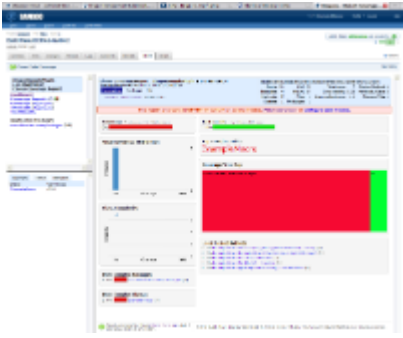
Adding Another Atlassian Tool to your Suite

This section tells you about another Atlassian developer tool that you can add to your integrated suite, [Clover](#) for code coverage.

Clover for Code Coverage

[Clover](#) is a code coverage tool for Java. 'Code coverage' means that Clover measures how much of your Java code is executed by your tests. Clover has several differentiating features, including the ability to [optimise](#) your test execution (make your builds faster), measure per-test coverage and produce interactive HTML reports. Clover provides plugins for [Eclipse](#) and [IntelliJ IDEA](#).

Installation	<p>Clover offers several different installation options, depending on your development and build tools. The details are in the Clover documentation. Below are the instructions for using Clover within Bamboo. (This configuration will work only for Java projects using Maven 2 or Ant.) Since you have already installed Bamboo, it is very simple to enable the Clover plugin for Bamboo. All you need is a license key:</p> <ol style="list-style-type: none"> 1. Go to your Bamboo URL in your browser, e.g. http://www.foo.com:8085. 2. Log in to Bamboo with username charlie. 3. Make sure you are on the 'All Plans' tab of the 'Home' screen. (Click 'Home' in the top navigation bar, then click 'All Plans'.) 4. Click the name of the plan, 'Main', to open the plan summary. 5. Click the 'Configuration' tab. 6. Click the 'Builder' tab. 7. Click the 'Edit Plan' link. 8. Select the checkbox labelled 'Use Clover to collect Code Coverage for this build'. 9. A new section of the screen will open. Select the radio button labelled 'Automatically integrate Clover into this build'. 10. More reporting options appear. At this stage it is fine to leave them unselected. 11. <i>Optional</i> – Enter your 'Clover License': <ul style="list-style-type: none"> • If you have a Clover license, enter the license key. • If you do not have a Clover license, leave the license field empty. You can use Clover in Bamboo for 30 days without obtaining a license. 12. Click 'Save'.
--------------	--

Overview	<p>Now that you have Clover in Bamboo, you can run a build and see the code coverage. You can also use Bamboo's report generator to see the Clover Lines of Code report and the Clover Code Coverage report.</p> <p>If you would like to try it out with our sample repository, follow the instructions below.</p> <p>i Please note: Running the build of our sample project will take quite a long time: approximately 20 minutes. This is because the build procedure will download and start the Confluence web application so that it can run the integration tests.</p> <div style="border: 1px dashed #ccc; padding: 10px; margin: 10px 0;"> <ol style="list-style-type: none"> 1. Click 'Build Actions' near the top right of the Bamboo build plan screen, for your Dragons Main plan. 2. A dropdown menu will appear. Click 'Run Build'. 3. The build will start. If this is the second time you have run a build, it will be called DRAG-MAIN-2. Have a cup of hot chocolate while the build runs. It will take some time – approximately 20 minutes. Here is a summary of what it will do: <ul style="list-style-type: none"> • Download the Clover plugin for Bamboo. • Run the unit tests. • Download the Confluence web application. It does this because our sample project is a Confluence plugin. • Start a Confluence server in Tomcat on port 1990. • Run the integration tests. • Shut down Confluence. • Collect the Clover artifacts. • Finally, report that the build is successful. (Or that it has failed.) • When the build has finished, click the build name, e.g. DRAG-MAIN-2, near the top right of the screen. 4. The 'Build Result' screen will appear. Click the 'Clover' tab. 5. The 'Clover Code Coverage' screen will appear. </div> <p><u>Screenshot 2 (click to enlarge): Clover in Bamboo</u></p> 
Classic Clover	<p>More things you can do with Clover:</p> <ul style="list-style-type: none"> • From within IntelliJ IDEA, view recently-run tests via the Test Run Explorer, see the Java code annotated with coverage information, view coverage cloud and treemap reports, and optimise your test builds. • From within Eclipse, view recently-run tests via the Test Run Explorer, see the Java code annotated with coverage information, view coverage cloud and treemap reports, see the unit tests and methods that generated coverage for the currently opensource file, and optimise your test builds. • Use Clover for Ant, interactively or in automated builds, with a range of current and historical reports, clouds and charts. • Use Clover for Maven 1 or for Maven 2, view a range of historical and custom reports, and optimise your test builds.
Videos and tours	<p>See our website.</p>

Hints after Initial Setup

These hints may be useful in the early days after you complete your initial setup. Click the links to see the details of each hint.

[Adding Users to your Atlassian Integrated Suite](#)

During the **Atlassian Dragon quest**, you added just one user to your integrated suite: **Charlie of Atlassian**. Very soon you will want to add more users, and in particular users who are not administrators. We recommend that you use Crowd for all user and group management. Below is a suggested plan of action and a hint about what to do if new users experience a delay before they can see their JIRA projects.

[Running Bamboo in a Console Window](#)

If you experience problems with running Bamboo as a Windows service you can start Bamboo in a console window instead, by running `{ BAMBOO_INSTALL }\bin\BambooConsole.bat`.

RELATED TOPICS

Here Be Dragons

Adding Users to your Atlassian Integrated Suite

During the [Atlassian Dragon quest](#), you added just one user to your integrated suite: **Charlie of Atlassian**. Very soon you will want to add more users, and in particular users who are not administrators. We recommend that you use Crowd for all user and group management. Below is a suggested plan of action and a hint about what to do if new users experience a delay before they can see their JIRA projects.

1. Configure Crowd to automatically give new users access to JIRA, Confluence and other applications when each user is added. Details are in the [Crowd documentation](#). Here is a summary:
 - Go to your Crowd URL in your browser, e.g. <http://coopers:8095/crowd>.
 - Log in with username [charlie](#).
 - Click the **'Directories'** link in the top navigation bar.
 - The **'Directory Browser'** appears. Click the link on the **'Crowd'** directory name.
 - The directory **'Details'** screen appears. Click the **'Options'** tab.
 - The **'Options'** screen appears. Click the **'Add Groups'** button.
 - The **'Add Groups'** popup screen appears. Leave the search box empty to match all group names, and click **'Search'**.
 - Select the groups by putting a tick in the checkbox next to one or more group names. For example, select the following groups to ensure that all new users are automatically given access to JIRA and Confluence:
 - confluence-users
 - jira-developers
 - jira-users
 - Click the **'Add Selected groups'** button.
2. Add your new users via Crowd. Details are in the [Crowd documentation](#). Here is a summary:
 - Click the **'Users'** tab in the top navigation bar.
 - The **'User Browser'** appears. Click **'Add User'** in the left-hand menu.
 - Enter the user information and then click **'Create'**.



Do you notice a delay before new users can see their JIRA projects?

After you have added a new user to Crowd, there may be a delay of a few minutes before that user can see all the information in JIRA, Confluence or another application. This is because the user and group information is stored in a client cache. We recommend that you leave the default cache settings as they are, unless there is an urgent need to change them. The client cache settings are explained in the [Crowd documentation](#).

RELATED TOPICS

[Dragons Stage 1 - Install Java, PostgreSQL and Crowd](#)
[Crowd documentation](#)

Running Bamboo in a Console Window

If you experience problems with running Bamboo as a Windows service you can start Bamboo in a console window instead, by running `{BAMBOO_INSTALL}\bin\BambooConsole.bat`.

RELATED TOPICS

[Dragons Stage 8 - Install Bamboo](#)

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 the latest JIRA release. 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 [raise a support ticket](#) for the product you're stuck on. Please don't try to battle on alone. Instead, ask for help immediately. You can also seek assistance on the [Dragon Slayers' Forum](#), where you're sure to meet other battle-weary dragon slayers.

Rushing into the Dragon's Lair



Don your armour and alert your serfs

If you like, you can [tweet your status](#).



Follow yon brave dragon slayers

On the [Atlassian Dragons Twitter stream](#).

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 and Crucible](#)
- [Dragons Stage 6 - Get JIRA and FishEye Talking](#)
- [Dragons Stage 7 - Get JIRA and Crucible Talking](#)
- [Dragons Stage 8 - Install Bamboo](#)
- [Dragons Stage 9 - Bamboo Gadgets and JIRA Victory After Dragons](#)

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 1.6 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 version of the JDK 1.6, at least **version 6u23 or later**.
 - If you are running 64-bit Windows, please ensure that you use **32-bit JDK** and not the 'x64' JDK.
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 use that database for your other applications too. 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 8.4.x**.

1. Download [PostgreSQL](#) – Get the latest 8.4.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. Ensure that you choose UTF8 (unicode) encoding when selecting the locale. 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 8.4.x JDBC driver from <http://jdbc.postgresql.org/download.html> and save it locally for later use. Get the [JDBC4 Postgresql Driver, Version 8.4-702](#).
 - ⚠ Internet Explorer may rename the file extension from '.jar' to '.zip' when you download it. If you are using Internet Explorer, please rename the file so that it has a '.jar' extension after downloading it.

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.

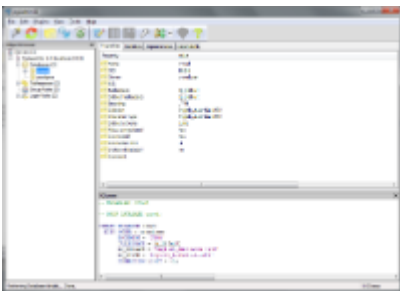
 We're using [pgAdmin III](#), the administration user interface supplied with PostgreSQL. If you used the one-click installer, pgAdmin III will be already installed on your computer.

1. Start pgAdmin III.
2. Double-click the name of the PostgreSQL server in the '**Object browser**' window and enter the password for the 'postgres' super user as prompted.
3. Add a new login role called 'crowduser':
 - Right-click '**Login Roles**' and select '**New Login Role**'.
 - Enter the '**Role name**': crowduser.
 - Enter a '**Password**' and enter it again to confirm it.
 - Click the '**Role privileges**' tab.
 - Select '**Can create database objects**'.
 - Select '**Can create roles**'.
 - Click '**OK**' to create the user.
4. 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/8.4/bin/, enter the following commands:

```
sudo -s -H -u postgres
# Create the Crowd user:
/opt/PostgreSQL/8.4/bin/createuser -S -d -r -P -E crowduser
# Create the Crowd database:
/opt/PostgreSQL/8.4/bin/createdb -O crowduser crowd
exit
```

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



Step 4. Install Crowd

Requirements: **Crowd 2.2.2**.

► For Windows: (click to expand)

1. Go to the Atlassian [download centre](#).
2. Download the '**Standalone (ZIP Archive)**' file for **Crowd 2.2.2**.
3. Unpack the zip archive into a directory of your choice, avoiding spaces in the directory name.
4. Tell Crowd where to put 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 2.2.2**.
3. Unpack the archive into a directory of your choice, avoiding spaces in the directory name.
4. Tell Crowd where to put 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 [raise a support ticket](#) for the product you're stuck on, or try the [Dragon Slayers' Forum](#).

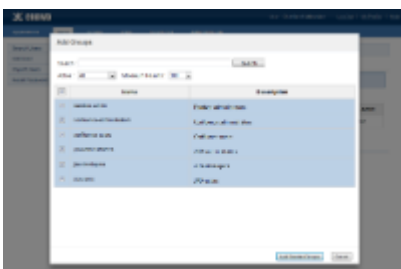
Victory? Please continue.

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.
 - Session Timeout – Leave this at the default value, i.e. 30
 - 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.fooobar.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'** in 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 top navigation bar and find **'Charlie of Atlassian'**.
 - Click the name to view Charlie's user information.
 - Click the **'Groups'** tab under **'View User'**, 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 raise a support ticket for the product you're stuck on, or try 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 raise a support ticket for the product you're stuck on, or try 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**.

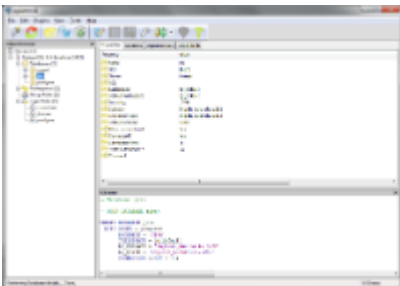
 We are using pgAdmin III, the administration user interface supplied with PostgreSQL. If you used the one-click installer when installing PostgreSQL, pgAdmin III will be already installed on your computer.

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.
 - Click the '**Role privileges**' tab.
 - 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/8.4/bin/, enter the following commands:

```
sudo -s -H -u postgres
# Create the JIRA user:
/opt/PostgreSQL/8.4/bin/createuser -S -d -r -P -E jirauser
# Create the JIRA database:
/opt/PostgreSQL/8.4/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 4.3**.

1. Check your version of JIRA.
2. If you do not have **JIRA 4.3** or later, follow the instructions on [upgrading to JIRA 4.3](#).

Problems? Please raise a [support ticket](#) for the product you're stuck on, or try 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. Turn on the public API and allow unassigned issues:
 - Click '**Administration**' in the top navigation bar.
 - Click '**General Configuration**' in the left-hand panel (in the 'Global Settings' section).
 - Enter your password as prompted, to confirm that you want administrator access. (Note that the Atlassian applications will request this confirmation at various steps in the process. This guide will not mention this step again.)
 - Click '**Edit Configuration**'.
 - Select the '**ON**' radio button next to '**Allow unassigned issues**'.
 - Select the '**ON**' radio button next to '**Accept remote API calls**'.
 - Click '**Update**'.

Problems? Please [raise a support ticket](#) for the product you're stuck on, or try 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 [raise a support ticket](#) for the product you're stuck on, or try the [Dragon Slayers' Forum](#).


Victory? Please continue.

Step 5. Hook JIRA up to Crowd


In this step you will configure JIRA to use Crowd for SSO and centralised user management. To do that, you will define the JIRA application in Crowd, define the Crowd user directory in JIRA, and configure the SSO property files.

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.foo.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. The first screen of the Crowd **'Add Application' wizard** will appear. 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.foo.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, excluding the `'http://www.'` prefix and the `':8080'` port number, e.g. `foo.com`.
 - If the following IP address is not already present, add it: `127.0.0.1`.
8. Connect JIRA to the Crowd user directory:
 - Go to your JIRA URL in your browser, e.g. <http://www.foo.com:8080>.
 - Select **'User Directories'** from the **'Users, Groups & Roles'** section of the 'Administration' menu.
 - Click **'Add Directory'**, select type **'Atlassian Crowd'** and click **'Next'**.
 - The Crowd server configuration screen will appear. Enter the following information:
 - Name – Accept the default value, `Crowd Server`.
 - Server URL – Enter the web address of your Crowd server, e.g. <http://www.foo.com:8095/crowd>
 - Application Name: `jira`. This is the application name that you used to define JIRA in the Crowd 'Add Application' wizard above.
 - Application Password – Enter the password that you defined for JIRA in the Crowd 'Add Application' wizard above.
 - Crowd Permissions – Select `Read/Write`.
 - Leave the other settings at their default values and click the **'Test Settings'** button to test the connection.
 - When you have a working connection, click **'Save'**.
 - The 'User Directories' screen will appear. Now you will move the Crowd directory to the top of the list of directories. Click the **blue upward arrow** in the **'Order'** column next to the **'Crowd Server'**, so that the Crowd directory moves to the top of the list

Here is a summary of how the directory order affects the processing:

- The order of the directories is the order in which they will be searched for users and groups.
 - Changes to users and groups will be made only in the first directory where the application has permission to make changes.
9. Log out of JIRA, but leave JIRA running. (Click the dropdown arrow next to the name **'Charlie of Atlassian'**, then select **'Log Out'**.)
 10. Log in to JIRA again, with the same username [charlie](#) and Charlie's password in Crowd.
 -  You are now authenticating via Crowd!
 11. 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).)
 12. Configure the JIRA property files for SSO:
 - Edit the `{JIRA_INSTALL}/atlassian-jira/WEB-INF/classes/seraph-config.xml` file.
 - Comment out the default authenticator node:


```
.....
.....
```
 - Uncomment the line that contains the new authenticator:


```
.....
.....
```
 - Save the `seraph-config.xml` file.
 - Copy the `crowd.properties` file from `{CROWD_INSTALL}/client/conf/` to `{JIRA_INSTALL}/atlassian-jira/WEB-INF/classes`.
 - 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.
 - Save the `crowd.properties` file.
 -  You now have SSO between JIRA and Crowd! Try it:
 - Start your JIRA server again, and go to your JIRA URL in your browser, e.g. <http://www.foo.com:8080>.
 - If you are already logged in to Crowd, you will not need to log in to JIRA. SSO ensures that you are already logged in as [charlie](#).
 - Log out of JIRA.
 - Go to Crowd and click an option. Crowd will prompt you to log in. When you logged out of JIRA, SSO ensured that you logged out of Crowd too.
 - Log in to either JIRA or Crowd. You will be logged in to both.

Screenshot 3: User directories in JIRA



Full details are in the [Crowd documentation](#) and the [JIRA administrator's guide](#).

i 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 [raise a support ticket](#) for the product you're stuck on, or try 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 right 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 [raise a support ticket](#) for the product you're stuck on, or try 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 [raise a support ticket](#) for the product you're stuck on, or try 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](#).