Space Details

<table>
<thead>
<tr>
<th>Key:</th>
<th>BAMBOO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name:</td>
<td>Bamboo</td>
</tr>
<tr>
<td>Description:</td>
<td></td>
</tr>
<tr>
<td>Creator (Creation Date):</td>
<td><a href="mailto:edwin@atlassian.com">edwin@atlassian.com</a> (Jul 31, 2006)</td>
</tr>
<tr>
<td>Last Modifier (Mod. Date):</td>
<td><a href="mailto:edwin@atlassian.com">edwin@atlassian.com</a> (Jul 31, 2006)</td>
</tr>
</tbody>
</table>

Available Pages

- Bamboo Documentation Home
  - Bamboo Administrator’s Guide
    - 01. Configuring Projects and Plans
      - 1.1 About Projects, Plans and Builds
      - 1.2 Creating a Plan
      - 1.3 Editing a Plan
      - 1.4 Specifying Labels for a Plan's Build Results
      - 1.5 Moving a Plan to a different Project
      - 1.6 Disabling or deleting a Plan
        - 1.6.1 Deleting a Build Result
    - 02. Configuring Build Resources
      - 2.1 Configuring a new Builder
      - 2.2 Configuring a new JDK
      - 2.3 Configuring Global Variables
    - 03. Triggering a Build
      - 3.1 About Build Triggering
      - 3.2 Triggering a Build when Code is Updated
        - 3.2.1 Polling the Repository for Code Changes
        - 3.2.2 Triggering a Build on Code Check-in
      - 3.3 Triggering a Build on Schedule
        - 3.3.1 Scheduling a Single Daily Build
        - 3.3.2 Specifying a Cron-based Schedule
      - 3.4 Triggering a Build when another Build finishes
      - 3.5 Triggering a Build Manually
    - 04. Managing Build Queues
      - 4.1 Creating a Build Queue
      - 4.2 Assigning a Plan to a Build Queue
      - 4.3 Removing a Plan from a Build Queue
      - 4.4 Stopping an Active Build
      - 4.5 Disabling or deleting a Build Queue
    - 05. Managing Users and Security
      - 5.01 Creating a User
      - 5.02 Deleting a User
• 5.03 Creating a Group
• 5.04 Deleting a Group
• 5.05 Adding Users to and removing them from Groups
• 5.06 Granting Administration Privileges to a User
• 5.07 Enabling or disabling Contact Details Display
• 5.08 Enabling or disabling Public Signup
• 5.09 Enabling or disabling Anonymous Access
• 5.10 Working with External User Repositories
  • 5.10.1 Integrating Bamboo with Crowd
  • 5.10.2 Integrating Bamboo with LDAP
• 06. Configuring Email and Instant Messaging Notifications
• 6.1 Enabling or disabling Notifications for a Plan
• 6.2 Configuring Bamboo to send SMTP Email
• 6.3 Configuring Bamboo to use Instant Messaging (IM)
  • 6.3.1 Configuring Bamboo to use Google Talk for Instant Messaging
• 07. Managing Data and Backups
• 7.1 Re-indexing Data
• 7.2 Exporting Data for Backup
• 7.3 Importing Data from Backup
• 7.4 Specifying the Expiry Date for Build Results
• 08. Configuring System Settings
• 8.1 Locating Important Directories and Files
  • 8.1.1 Specifying Bamboo's Working Directory
• 8.2 Viewing Bamboo's System Information
• 8.3 Updating your Bamboo License Details
• 8.4 Specifying Bamboo's URL
• 8.5 Enabling JIRA integration
• 8.6 Enabling GZIP Compression
• 8.7 Enabling Bamboo's Remote API
• 09. Configuring Plugins
• 9.1 About Bamboo Plugins
• 9.2 Enabling the 'Auto-Favourites' Plugin
• 9.3 Installing a new Plugin
• Appendix A. Embedding Bamboo into Other Applications
  • Javascript Widgets
    • All Plans & My Favourite Plans
    • Latest Builds
    • Latest Status of a Plan
    • My Latest Changes
    • Plan Summary Graphs
• Bamboo Installation Guide
  • Bamboo EAR-WAR Installation Guide
  • Bamboo Standalone Installation Guide (Linux)
  • Bamboo Standalone Installation Guide (Mac)
- Bamboo Standalone Installation Guide (Windows)
- Running the Setup Wizard
- Bamboo Knowledge Base
  - Installation FAQ
  - Integration FAQ
  - Usage FAQ
    - Can multiple plans share a common 3rd-party directory?
- Bamboo Release Notes
  - Bamboo 0.1 Release Notes
  - Bamboo 0.2 Release Notes
    - Bamboo 0.2 Upgrade Guide
  - Bamboo 0.3 Release Notes
    - Bamboo 0.3 Upgrade Guide
  - Bamboo 0.4 Release Notes
    - Bamboo 0.4 Upgrade Guide
  - Bamboo 0.5 Release Notes
    - Bamboo 0.5 Upgrade Guide
  - Bamboo 0.5.1 Release Notes
    - Bamboo 0.5.1 Upgrade Guide
  - Bamboo 0.6 Release Notes
    - Bamboo 0.6 Upgrade Guide
  - Bamboo 0.7 Release Notes
    - Bamboo 0.7 Upgrade Guide
  - Bamboo 0.8 Release Notes
    - Bamboo 0.8 Upgrade Guide
  - Bamboo 0.9 Release Notes
    - Bamboo 0.9 Upgrade Guide
  - Bamboo 1.0 Release Notes
    - Bamboo 1.0 Upgrade Guide
  - Bamboo 1.0-Beta Release Notes
    - Bamboo 1.0-Beta Upgrade Guide
  - Bamboo 1.0-RC1 Release Notes
    - Bamboo 1.0-RC1 Upgrade Guide
  - Bamboo 1.0-RC2 Release Notes
    - Bamboo 1.0-RC2 Upgrade Guide
  - Bamboo 1.0.1 Release Notes
    - Bamboo 1.0.1 Upgrade Guide
  - Bamboo 1.0.2 Release Notes
    - Bamboo 1.0.2 Upgrade Guide
  - Bamboo 1.0.3 Release Notes
    - Bamboo 1.0.3 Upgrade Guide
  - Bamboo 1.0.4 Release Notes
    - Bamboo 1.0.4 Upgrade Guide
  - Bamboo 1.0.5 Release Notes
    - Bamboo 1.0.5 Upgrade Guide
• Bamboo 1.1 Release Notes
  • Bamboo 1.1 Upgrade Guide
• Bamboo Upgrade Guide
• Bamboo User's Guide
  • 01. Getting Started
    • 1.1 Using the Bamboo Dashboard
    • 1.2 Viewing Bamboo's Current Activity
    • 1.3 Viewing your Latest Build Results
    • 1.4 Working with Favourites
      • 1.4.1 Adding a Plan to your Favourites
      • 1.4.2 Removing a Plan from your Favourites
  • 02. Working with Projects and Plans
    • 2.1 About Projects and Plans
    • 2.2 Viewing a Plan's Details
    • 2.3 Viewing a Plan's Activity Log
  • 03. Working with Build Results
    • 3.1 About Builds and Build Results
    • 3.2 Viewing a Build Result
    • 3.3 Viewing a Build Log
    • 3.4 Viewing the Code Changes that triggered a Build
    • 3.5 Viewing Clover Code-Coverage for a Build Result
    • 3.6 Viewing Metadata for a Build Result
    • 3.7 Viewing JIRA Issues for a Build Result
  • 04. Working with Tests
    • 4.1 Viewing Test Results for a Build
    • 4.2 Viewing a Test's History
    • 4.3 Viewing Test Statistics for a Plan
  • 05. Reporting on Plan Trends
    • 5.1 Viewing Build Statistics for a Plan
    • 5.2 Generating Reports across multiple Plans
      • 'Build Activity per Plan' Report
      • 'Build Duration per Plan' Report
      • 'Clover Code Coverage per Plan' Report
      • 'Clover Lines of Code per Plan' Report
      • 'Number of Build Failures per Plan' Report
      • 'Number of Tests per Plan' Report
      • 'Percentage of Successful Builds per Plan' Report
      • 'Time to Fix per Plan' Report
  • 06. Reporting on Author Trends
    • 6.1 Viewing Build Statistics for all Authors
    • 6.2 Viewing Build Results for an Author
    • 6.3 Generating Reports on selected Authors
      • 'Build Activity per Author' Report
      • 'Number of Build Failures per Author' Report
      • 'Number of Builds Broken per Author' Report
• 'Number of Builds Fixed per Author' Report
• 'Percentage of Successful Builds per Author' Report

• 07. Working with Comments
  • 7.1 About Comments
  • 7.2 Commenting about a Build Result
  • 7.3 Viewing Comments about a Build Result
  • 7.4 Viewing Code Check-in Comments

• 08. Working with Labels
  • 8.1 About Labels
  • 8.2 Labelling a Build Result
  • 8.3 Removing a Label from a Build Result
  • 8.4 Viewing Labelled Build Results
  • 8.5 Viewing Popular Labels

• 09. Subscribing to RSS Feeds
  • 9.1 Subscribing to an RSS Feed for All Build Results for All Plans
  • 9.2 Subscribing to an RSS Feed for Failed Builds for All Plans
  • 9.3 Subscribing to an RSS Feed for All Build Results for a Particular Plan
  • 9.4 Subscribing to an RSS Feed for Failed Builds for a Particular Plan
  • 9.5 Subscribing to an RSS Feed for Labelled Build Results

• 10. Working with Instant Messenger (IM) Notifications
  • 10.1 About Instant Messenger (IM) Notifications
  • 10.2 Labelling a Build Result via IM
  • 10.3 Commenting about a Build Result via IM

• 11. Editing your User Profile
  • 11.1 Changing your Password
  • 11.2 Changing your Notification Preferences
  • 11.3 Associating your Author Name with your User Profile

• Appendix A. Bamboo Glossary
  • activity log
  • artifact
  • author
  • build
  • build activity
  • build duration
  • build log
  • build plan
  • build queue
  • build result
  • build telemetry
  • builder
  • favourites
  • label
  • plan
  • project
- triggering
- Bamboo Development Hub
  - Bamboo API
  - Bamboo Plugin Developer's Guide
    - Build Complete Action Module
    - Build Processor Module
    - Builder Plugin Module
    - Getting Started
    - Index Reader Module
    - Notification Condition Module
    - Post Build Index Writer Module
    - Pre Build Action Module
    - Report Module
    - Web Item Module
    - Web Section Module
    - XWork Plugin Module
  - Bamboo Plugin Tutorial
    - Tutorial 1 - Getting Started with a Simple Post Build Labeller
    - Tutorial 2 - Configurable Regex Labeller
Welcome to the new Atlassian Bamboo! Bamboo is Atlassian’s new Continuous Integration Server.

Bamboo 1.1 has just been released (08/05/2007)!!! Check out Bamboo’s features and try it out while it’s hot!

<table>
<thead>
<tr>
<th>Installing/upgrading Bamboo 1.1</th>
<th>Using/administering Bamboo 1.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Installation Guide</td>
<td>User's Guide</td>
</tr>
<tr>
<td>Upgrade Guide</td>
<td>Administrator's Guide</td>
</tr>
<tr>
<td>Release Notes</td>
<td>Development Hub</td>
</tr>
<tr>
<td>Download Bamboo</td>
<td>Support</td>
</tr>
</tbody>
</table>

For previous versions please see:

- Bamboo 1.0 Documentation Home

Recently Updated

by Rosie Jameson (6 mins ago)
04. Managing Build Queues

by Rosie Jameson (9 mins ago)
4.2 Assigning a Plan to a Build Queue

by Rosie Jameson (14 mins ago)
Bamboo-queue-edit.PNG

by Bob Swift (2 hours ago)
Re: 5.10.2 Integrating Bamboo with LDAP
## Bamboo 1.1 Administrator's Guide

<table>
<thead>
<tr>
<th>Bamboo 1.1 Administrator's Guide</th>
<th>Download</th>
</tr>
</thead>
<tbody>
<tr>
<td>01. Configuring Projects and Plans</td>
<td>You can download the Bamboo Administrator's Guide in PDF, HTML or XML formats.</td>
</tr>
<tr>
<td>02. Configuring Build Resources</td>
<td></td>
</tr>
<tr>
<td>03. Triggering a Build</td>
<td></td>
</tr>
<tr>
<td>04. Managing Build Queues</td>
<td></td>
</tr>
<tr>
<td>05. Managing Users and Security</td>
<td></td>
</tr>
<tr>
<td>06. Configuring Email and Instant Messaging Notifications</td>
<td></td>
</tr>
<tr>
<td>07. Managing Data and Backups</td>
<td></td>
</tr>
<tr>
<td>08. Configuring System Settings</td>
<td></td>
</tr>
<tr>
<td>09. Configuring Plugins</td>
<td></td>
</tr>
<tr>
<td>Appendix A, Embedding Bamboo into Other Applications</td>
<td></td>
</tr>
<tr>
<td>Index</td>
<td></td>
</tr>
</tbody>
</table>

### Search

Type `CTRL-F` to search the full Table of Contents on this page.

### About

The Bamboo Administrator's Guide provides information for system administrators of Bamboo. If you have a question that hasn't been answered here, please let us know.

For more documentation please visit Bamboo Documentation Home.

For assistance with Bamboo, please log a support request.

---

## Table of Contents

### 01. Configuring Projects and Plans

- 1.1 About Projects, Plans and Builds
- 1.2 Creating a Plan
- 1.3 Editing a Plan
- 1.4 Specifying Labels for a Plan's Build Results
- 1.5 Moving a Plan to a different Project
- 1.6 Disabling or deleting a Plan
  - 1.6.1 Deleting a Build Result

### 02. Configuring Build Resources
03. Triggering a Build

- 3.1 About Build Triggering
- 3.2 Triggering a Build when Code is Updated
  - 3.2.1 Polling the Repository for Code Changes
  - 3.2.2 Triggering a Build on Code Check-in
- 3.3 Triggering a Build on Schedule
  - 3.3.1 Scheduling a Single Daily Build
  - 3.3.2 Specifying a Cron-based Schedule
- 3.4 Triggering a Build when another Build finishes
- 3.5 Triggering a Build Manually

04. Managing Build Queues

- 4.1 Creating a Build Queue
- 4.2 Assigning a Plan to a Build Queue
- 4.3 Removing a Plan from a Build Queue
- 4.4 Stopping an Active Build
- 4.5 Disabling or deleting a Build Queue

05. Managing Users and Security

- 5.01 Creating a User
- 5.02 Deleting a User
- 5.03 Creating a Group
- 5.04 Deleting a Group
- 5.05 Adding Users to and removing them from Groups
- 5.06 Granting Administration Privileges to a User
- 5.07 Enabling or disabling Contact Details Display
- 5.08 Enabling or disabling Public Signup
- 5.09 Enabling or disabling Anonymous Access
- 5.10 Working with External User Repositories
  - 5.10.1 Integrating Bamboo with Crowd
  - 5.10.2 Integrating Bamboo with LDAP

06. Configuring Email and Instant Messaging Notifications

- 6.1 Enabling or disabling Notifications for a Plan
- 6.2 Configuring Bamboo to send SMTP Email
- 6.3 Configuring Bamboo to use Instant Messaging (IM)
  - 6.3.1 Configuring Bamboo to use Google Talk for Instant Messaging

07. Managing Data and Backups
08. Configuring System Settings

- 8.1 Locating Important Directories and Files
  - 8.1.1 Specifying Bamboo's Working Directory
- 8.2 Viewing Bamboo's System Information
- 8.3 Updating your Bamboo License Details
- 8.4 Specifying Bamboo's URL
- 8.5 Enabling JIRA integration
- 8.6 Enabling GZIP Compression
- 8.7 Enabling Bamboo's Remote API

09. Configuring Plugins

- 9.1 About Bamboo Plugins
- 9.2 Enabling the 'Auto-Favourites' Plugin
- 9.3 Installing a new Plugin

Appendix A. Embedding Bamboo into Other Applications

- Javascript Widgets
  - All Plans & My Favourite Plans
  - Latest Builds
  - Latest Status of a Plan
  - My Latest Changes
  - Plan Summary Graphs

Index
01. Configuring Projects and Plans

1. Configuring Projects and Plans

- 1.1 About Projects, Plans and Builds
- 1.2 Creating a Plan
- 1.3 Editing a Plan
- 1.4 Specifying Labels for a Plan's Build Results
- 1.5 Moving a Plan to a different Project
- 1.6 Disabling or deleting a Plan
  - 1.6.1 Deleting a Build Result
1.1 About Projects, Plans and Builds

A Bamboo plan (or build plan) is the "recipe" for a build.

A plan defines: what gets built (i.e. the source-code repository); how the build is triggered; which builder to use; what tests to run; what artifacts the build will produce; who will be notified of the build result; and any labels with which the build result or build artifacts will be tagged.

Every plan belongs to a project.

A project enables easy identification of plans that are logically related to each other, which is useful for instance when generating reports across multiple plans. Each project has a Name (e.g. "CRM System") and a Key (e.g. "CRM"). The Project Key is prefixed to the relevant Plan Keys, e.g. the "CRM" project could have plans "CRM-TRUNK" and "CRM-BRANCH". Note that creating a new project only requires defining the Project Name and Project Key, which is (optionally) done as part of the process of creating a new plan.

A build is one execution of a plan.

Every build has a Build Number, which is appended to the relevant Plan Key to form the Build Key. For example, if a plan with the key "CRM-BRANCH" is executed for the seventeenth time, the build key will be "CRM-BRANCH-17".

RELATED TOPICS

- 1.1 About Projects, Plans and Builds
- 1.2 Creating a Plan
- 1.3 Editing a Plan
- 1.4 Specifying Labels for a Plan's Build Results
- 1.5 Moving a Plan to a different Project
- 1.6 Disabling or deleting a Plan
  - 1.6.1 Deleting a Build Result

Bamboo Documentation Home
1.2 Creating a Plan

To create a new plan,

1. Click the 'Create Plan' link in the top navigation bar.
2. Enter the required information in the 6 screens as described below. When you return to the Dashboard, your new plan (and new project, if applicable) will be displayed in the 'All Projects' list.

To copy an existing plan,

1. Click the 'Create Plan' link in the top navigation bar.
2. On the 'Plan Details' screen, select the check-box 'Clone an existing build plan?'
3. A list called 'Plan to clone' will be displayed. Select the plan you wish to copy.
4. Enter the required information in the 6 screens as described below. On screens 2-6, appropriate information will be copied from the plan you selected; but you will need to complete all fields on screen 1.

Screenshot 1. 'Plan Details'

Project Name:

How do you want to call the project within Bamboo? e.g. 'Issue Tracking Application'.

Project Key:

This is the unique project key to identify a project. The key must contain only alphanumeric characters. e.g. 'ITA'.

Build Plan Name:

How do you want to identify the plan within the project? Names must be unique within the project. e.g. 'Core Branch'.

Build Plan Key:

This is the key for the plan which must unique within a project. In conjunction with the project key, it is used to identify a build in URLs, trigger scripts and API calls. The key must contain only alphanumeric characters. e.g. 'CORE'.

Clone an existing build plan?

1. 'Plan Details':

- Project — When you create a new plan, you can either add it to an existing project or create a new project. Either:
  - Select the appropriate project from the drop-down list;
  - or:
  - Select 'New Project' and complete the following two fields:
    1. Project Name — Type a descriptive name (e.g. 'Issue Tracking Application') that will identify your project on the Dashboard and in reports.
    2. Project Key — Type a logical contraction of the Project Name (e.g. 'ITA'). The Project Key
will be included in the plan's Build Results keys (e.g. 'ITA-MAIN-179'), so you may want to make it no longer than 3 or 4 characters. The Project Key must be unique within your Bamboo system.

- **Build Plan Name** — Type a name that will identify the plan within its project (e.g. 'Main Build', 'Branch', 'Unit Tests', 'Acceptance Tests'). Note that the Build Plan Name, which is displayed throughout Bamboo, is always accompanied by its Project Name.

- **Build Plan Key** — Type a logical contraction of the Build Plan Name. The Build Plan Key (e.g. 'MAIN') will be included in the plan's Build Results keys (e.g. 'ITA-MAIN-179'), so you may want to make it no longer than 3 or 4 characters. Note that the Build Plan Key only has to be unique within the project, that is, you could have a 'MAIN' plan in lots of different projects.

Tip: Project Name and Build Plan Name can be edited after the plan is created. A plan's Project Key and Build Plan Key are not editable, but can be changed as described in [1.5 Moving a Plan to a different Project](#).

### Screenshot 2. 'Source Repository'

|----------------|----------------------|--------------------------|-------------------|-----------------------|----------------|

#### Source Repository

- **Repository:**
  - **CVS:**
    - **CVS Root:** `server:atlassian.com:/svnroot/atlassian`
      - The full path to your CVS repository root. Bamboo supports per-user, per-repository, and global repository access methods.
    - **Authentication Type:** Password
    - **Password:**
      - (Optional) The password required to access the CVS repository.
    - **Quiet Period:** 2
      - CVS checkouts are not atomic. How many seconds should Bamboo wait between checkouts to determine if the checkout is complete?
    - **Module:**
      - The repository module containing the source code.
    - **Version of module:** HEAD
      - Which version of the module should Bamboo build?
    - **Include / Exclude Files:** Exclude all changes that matches the following pattern
    - **File Pattern:** `.\documentation.*`
      - A [regular expression](#) to match the file to be included / excluded.
    - **Web Repository URL:** `http://myserver.atlassian.com/mysvn/AtlassianCVS`
      - (Optional) The URL to the build's viewable repository.
    - **Web Repository Module:** myrepo
      - (Optional) The build's repository name if the above Web Repository URL points to multiple repositories.
    - **Build Strategy:** Polling the Repository for changes
      - How should Bamboo detect that the source repository has changed?
    - **Polling Frequency:** 180
      - How often (in seconds) should Bamboo check the repository for changes?

[Previous] [Next] [Cancel]
• ‘Repository’ — Select the type of repository from which Bamboo will check-out and build this plan's source-code. The following fields will vary depending on what type of repository you select:
  o **CVS:**
    - 'CVS Root' — Type the full path to your CVS repository root (e.g. `pserver:me@cvs.atlassian.com:/cvsroot/atlassian`). Bamboo supports pserver, ext (ssh) and local repository access methods.
    - 'Authentication Type' — Select either 'Password' or 'SSH'.
      - If you select 'Password', the following fields will appear:
        - 'Password' — (Optional) Type the password for your CVS repository.
        - 'Change Password' --- (Will only appear after you have saved the plan) Select this check-box if you want to change the password that is used to access the CVS repository.
      - If you select 'SSH', the following fields will appear:
        - 'Private Key' — Type the absolute path of your SSH private key.
        - 'Passphrase' — Type the passphrase for your SSH private key.
        - 'Change Passphrase' --- (Will only appear after you have saved the plan) Select this check-box if you want to change the password for your SSH private key.
    - 'Quiet Period' — This setting is used to avoid starting a build while someone is in mid-checkin. Bamboo will only initiate a build for this plan when no more changes are detected within the Quiet Period following the last known change. Type the number of seconds Bamboo should wait.
    - 'Module' — Type the name of the CVS module that contains the source-code.
    - 'Version of Module' — Select either 'HEAD' or 'Branch/Tag'. If you select 'Branch/Tag', the following field will appear:
      - 'Branch name' — Type the relevant branch name or tag.
  o **Subversion:**
    - 'Repository URL' — The location of subversion repository (e.g. [http://svn.collab.net/repos/svn/trunk](http://svn.collab.net/repos/svn/trunk)).
    - 'Username' — (Optional) The Subversion username (if any) required to access the repository.
    - 'Authentication Type' — Select either 'Password' or 'SSH'.
      - If you select 'Password', the following field will appear:
        - 'Password' — (Optional) Type the password required by the Subversion username (if applicable).
        - 'Change Password' --- (Will only appear after you have saved the plan) Select this check-box if you want to change the password that is used to access the Subversion repository.
      - If you select 'SSH', the following fields will appear:
        - 'Private Key' — Type the absolute path of your SSH private key.
        - 'Passphrase' — Type the passphrase for your SSH private key.
        - 'Change Passphrase' --- (Will only appear after you have saved the plan) Select this check-box if you want to change the password for your SSH private key.
    - 'Use Externals?' — (Optional) Select this check-box if your Subversion repository uses `svn:externals` to link to other repositories. Please note that you only need to select this check box if you require Bamboo to detect changes in the externals. If your externals references a particular (static) revision, you do not need to check this box.
  o **Perforce:**
    - If you wish to have several plans pointing to the same Perforce depot, you will need to set up a different client for each.
      - 'Perforce P4 client' — The location (on the Bamboo server) of the Perforce P4 client application. On UNIX systems this is typically `/usr/local/bin/p4`.
      - 'Port' — Type either the port to which the Perforce client will connect, or the Perforce server itself. This is the Perforce P4PORT environment variable that tells Bamboo which p4d (Perforce server) to use.
      - 'Client' — The name of the Perforce Client Workspace which Bamboo will use. The Client
Workspace determines which portions of the depot are visible in your Workspace Tree and where local copies of depot files are stored in your workspace.

- 'Depot' — The client view of the depot that contains the source-code files for this plan. This is typically in the form //<clientname>/depot/... For details please see the Perforce User's Guide.
- 'Username': (Optional) The Perforce username that Bamboo will use when it accesses the server ('Port'). Leave this field blank if you want Bamboo to use the default Perforce user (i.e. the OS username).
- 'Web Repository URL' — (Optional) You can specify the URL of the plan's browsable repository. If you specify a Web Repository URL, then links to relevant files will be displayed in the 'Code Changes' section of a build result.

- 'Include/Exclude Files' — (Optional) You can specify a particular inclusion or exclusion pattern for file changes to be detected.
- 'File Pattern' — (Optional) The regular expression for file changes for which you wish to include/exclude.
- 'Web Repository Module' — (Optional) The plan's repository name, if the above Web Repository URL points to multiple repositories.
- 'Build Strategy' — The default value, 'Poll the repository for changes', is a convenient option that requires no additional configuration. A number of other options are available; for details, please see 03. Triggering a Build. You can change the Build Strategy over time as required. The rest of the fields on this tab will vary depending on which Build Strategy you select.

Screenshot 3. 'Builder Configuration'
3. 'Builder Configuration':

To define a new Builder, please see 2.1 Configuring a new Builder.

- 'Builder' — From the list of available Builders, select the one which Bamboo will use to build this plan. The following fields will vary depending on what type of Builder you select:
  - Ant:
    - 'Build File' — (Optional) Type the relevant filename (e.g. build.xml)
    - 'Target' — Specify the Ant target you want Bamboo to execute each time the source code changes. For example: test (this will run the Ant target 'test'). You can also use '-D' to define one or more system properties, e.g.: -Djava.awt.headless=true (this will pass the parameter 'java.awt.headless' to with a value of 'true').
  - Maven:
    - 'Goal' — Specify the Maven goal you want Bamboo to execute each time the source code changes. For example: clean (this will run the Maven goal 'clean' followed by the Maven goal 'test'). You can also use '-D' to define one or more system properties, e.g.: -Djava.awt.headless=true (this will pass the parameter 'java.awt.headless' to with a value of 'true').
  - Custom command:
    - 'Argument' — Specify the relevant argument to pass to the command. Note that
arguments which contain spaces must be quoted.

- **Script:**
  - 'Script' — Specify the location of the script file. This can be either relative to the repository root of the plan, or absolute.
  - 'Argument' — Specify the relevant argument to pass to the script. Note that arguments which contain spaces must be quoted.

- 'Build JDK' — If you selected an Ant or Maven builder above, you will need to choose a JDK from the list.
  To define a new JDK in your Bamboo system, please see [2.2 Configuring a new JDK](#).

- 'System Environment Variables' — (Optional) Specify any operating system environment variables you want to pass to your build. These variables will be evaluated during build time. Variables must be in the format `${bamboo.<your_variable>}`.
  To configure global variable values that can be used across all plans, please see [2.3 Configuring Global Variables](#).

- 'Working Sub Directory' — (Optional) If you leave this field blank, Bamboo will look for the build files in the build root directory (which is assumed to be the build's Working Directory). You can override this option by specifying an alternative working directory (which must be a subdirectory of the root directory). For example, if your plan has a build script in a subdirectory, and the script needs to be run from within that subdirectory, you would type the name of that subdirectory in the 'Working Sub Directory' field.

- 'The build has tests' — Select this check-box if you want Bamboo to gather test results data for each build result. Choose one of the following:
  - 'Test Results Directory' — Select this option if Bamboo should look in the Builder's standard test results directory.
  - 'Specify custom results directories' — Select this option if the Builder will place generated test results in an alternative directory:
    - 'Specify custom results directories' — Type the name of the test results directory (or multiple directories, separated by commas). You can also use Ant-style patterns such as */test-reports/*.xml.

- 'Clover output will be produced' — Select this check-box if you are running Cenqua Clover and want to view its code-coverage data from within Bamboo. The following field will be displayed:
  - 'Clover XML Directory' — Type the name of the directory where Bamboo will look for the XML report output file from Clover. Please note that the Clover coverage analysis will only run for successful builds.

### Screenshot 4. 'Build Artifacts'

<table>
<thead>
<tr>
<th>Build Artifacts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Artifact Label</td>
</tr>
<tr>
<td>e.g. Clover</td>
</tr>
<tr>
<td>e.g. JAR files</td>
</tr>
</tbody>
</table>

### 4. 'Build Artifacts':

- Here you can specify the plan's artifacts, e.g. JAR files which you wish to keep after each build. Using the Artifact Copy Pattern, you can specify exactly which artifact(s) you want to keep. These
can be any reports, websites, or JAR files that were created by the build process. Build artifacts are copied to a subdirectory (/PROJECT_NAME/download_data) under your 'Projects Data' folder, which you specified when installing Bamboo (see 8.1 Locating Important Directories and Files).

### Screenshot 5. 'Notifications'

<table>
<thead>
<tr>
<th>Add Build Notification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Notification Trigger:</td>
</tr>
<tr>
<td>All Completed Builds</td>
</tr>
<tr>
<td>When should Bamboo send a notification?</td>
</tr>
<tr>
<td>Roles:</td>
</tr>
<tr>
<td>Committer</td>
</tr>
<tr>
<td>Groups:</td>
</tr>
<tr>
<td>Comma separated list of Groups</td>
</tr>
<tr>
<td>Users:</td>
</tr>
<tr>
<td>Comma separated list of Users</td>
</tr>
<tr>
<td>Email Addresses:</td>
</tr>
<tr>
<td>Comma separated list of email addresses (e.g. <a href="mailto:bob@work.com">bob@work.com</a>, <a href="mailto:bill@work.com">bill@work.com</a>)</td>
</tr>
<tr>
<td>Instant Messaging Addresses:</td>
</tr>
<tr>
<td>Comma separated list of m addresses (e.g. <a href="mailto:bob@mvnt.com">bob@mvnt.com</a>, <a href="mailto:bill@mvnt.com">bill@mvnt.com</a>)</td>
</tr>
<tr>
<td>Add</td>
</tr>
</tbody>
</table>

#### 5. 'Notifications':

- Here you can specify who will receive notifications about this plan's build results, and under what circumstances. For details please see 6.1 Enabling or disabling Notifications for a Plan.

### Screenshot 6. 'Post Actions'

<table>
<thead>
<tr>
<th>Post Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Please specify what you would like Bamboo to do once the build has been built.</td>
</tr>
</tbody>
</table>

**Pattern matching labelling.**

- **Regex Pattern:**
  - The regular expression for which to match the log files on. Leave this blank to label every build.

- **Label(s):**
  - The label(s) for the build if it matches the specified regex pattern.
6. 'Post Actions':

- 'Regex Pattern' — The regular expression for which to match the log files on. Labels will be applied to the build if the regular expression finds a match. You can leave this blank to label every build.
- 'Labels' — Type the labels (if any) which you want to automatically apply to the plan's build results. See 1.4 Specifying Labels for a Plan's Build Results.

RELATED TOPICS

- 1.1 About Projects, Plans and Builds
- 1.2 Creating a Plan
- 1.3 Editing a Plan
- 1.4 Specifying Labels for a Plan's Build Results
- 1.5 Moving a Plan to a different Project
- 1.6 Disabling or deleting a Plan
  - 1.6.1 Deleting a Build Result

Bamboo Documentation Home
1.3 Editing a Plan

To edit an existing plan,

1. Click 'Home' to go to the Dashboard.
2. Click the 'All Plans' tab.
3. Locate the plan in the list and click the 'Edit' icon:
4. The 'Configuration' tab will displayed. You can now edit the 'Plan Details' and other sub-tabs as described in 1.2 Creating a Plan.

RELATED TOPICS

- 1.1 About Projects, Plans and Builds
- 1.2 Creating a Plan
- 1.3 Editing a Plan
- 1.4 Specifying Labels for a Plan's Build Results
- 1.5 Moving a Plan to a different Project
- 1.6 Disabling or deleting a Plan
  - 1.6.1 Deleting a Build Result

Bamboo Documentation Home
1.4 Specifying Labels for a Plan's Build Results

A label is a convenient way to tag and group build results that are logically related to each other. Labels can also be used to define RSS feeds.

Labels can be applied to build results automatically, by specifying the label(s) in a build plan (note that only Bamboo administrators can do this). Labels can also be applied ad-hoc to build results by Bamboo users.

To specify labels for a plan's build results,

1. Click 'Home' to go to the Dashboard.
2. Click the 'All Plans' tab.
3. Locate the plan in the list and click this icon: 
4. The 'Configuration' tab will be displayed. Click the 'Post Actions' sub-tab.
5. In the 'Labels' field, type the word (or multiple words, separated by commas) with which the plan's build results are to be labelled.
6. Click the 'Save' button.

RELATED TOPICS

- 1.1 About Projects, Plans and Builds
- 1.2 Creating a Plan
- 1.3 Editing a Plan
- 1.4 Specifying Labels for a Plan's Build Results
- 1.5 Moving a Plan to a different Project
- 1.6 Disabling or deleting a Plan
  - 1.6.1 Deleting a Build Result

Bamboo Documentation Home
1.5 Moving a Plan to a different Project

A project enables easy identification of plans that are logically related to each other, which is useful for instance when generating reports across multiple plans. Each project has a Name (e.g. "CRM System") and a Key (e.g. "CRM"). The Project Key is prefixed to the relevant Plan Keys, e.g. the "CRM" project could have plans "CRM-TRUNK" and "CRM-BRANCH". Moving a plan to a different project will therefore involve changing the plan’s Project Key (as well as possibly the Plan Name and/or Plan Key), which will also change the build key for all of the plan’s build results.

Moving a plan does not affect the plan’s configuration, nor any comments or labels that have been applied to the plan's build results.

⚠️ Before you begin
Note that moving a plan will require Bamboo to re-index all its data, so your Bamboo system may run slowly for a few minutes. It is recommended that you backup your Bamboo build results before you move a plan — see 7.2 Exporting Data for Backup.

To move a plan to a different project,

1. Click the 'Administration' link in the top navigation bar.  
2. Click the 'Move Plans' link in the left navigation column.  
3. This will display the 'Move Build Plan Wizard'. Plans are listed under their project name, e.g. in Screenshot 1 below, the 'Clover Build' plan is listed under the 'Atlassian Config' project. Select the check-box for the plan you wish to move.  
4. Select the 'Destination Project' from the drop-down box at the bottom of the list.  
5. Click the 'Move' button at the bottom of the list.  
6. This will display the 'Configure New Plan Details' screen (see Screenshot 2 below).  
7. If the destination project already includes a plan with the same Plan Name, or if you want to change the Plan Name for some other reason, overtype the 'New Plan Name' field.  
8. If the destination project already includes a plan with the same Plan Key, or if you want to change the Plan Key for some other reason, overtype the 'New Plan Key' field.  
9. Click the 'Move' button.

Screenshot 1: 'Move Plans-Select Plans'
Move Build Plan Wizard

Move builds

It is strongly recommended that you ensure that all build queues are disabled before you perform the move.

You can move a plan to another project with this wizard. Simply select the plans you want to move and the destination project. As names and keys may conflict, you'll then be asked to enter new names and keys for the plans. Note that because we are changing plan keys, this operation requires some slow operations (e.g. indexing of all builds) and may take a few minutes.

Plan:

- Atlassian Bucket (BUCKET)
  - Main Build (MAIN)
- Atlassian Config (CONFIGS)
  - Main Build (MAIN)
  - Clover Build (CLOVER)
- Atlassian Core (CORE)
  - Main Build (MAIN)
- Atlassian Event (EVENT)
  - Main Build (MAIN)
- Atlassian Extras (EXTRA)
  - Main Build (MAIN)

Destination Project: Atlassian Bucket

The project you want to move your plans to

Move

Screenshot 2: 'Move Plans-Configure New Plan Details'

Configure New Plan Details

Choose new build keys and build names

The following plans will be moved to project Atlassian Bucket. You can update the plan names and keys below.

The existing plans for the destination project include:

- Main Build (MAIN)

<table>
<thead>
<tr>
<th>Original Project</th>
<th>Original Name</th>
<th>New Name</th>
<th>Original Key</th>
<th>New Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>Atlassian Config (CONFIGS)</td>
<td>Clover Build</td>
<td>Clover Build</td>
<td>CLOVER</td>
<td>CLOVER</td>
</tr>
</tbody>
</table>

Move | Cancel
1.1 About Projects, Plans and Builds
1.2 Creating a Plan
1.3 Editing a Plan
1.4 Specifying Labels for a Plan's Build Results
1.5 Moving a Plan to a different Project
1.6 Disabling or deleting a Plan
  1.6.1 Deleting a Build Result

Bamboo Documentation Home
1.6 Disabling or deleting a Plan

Sometimes, for example if a plan's latest build is broken and cannot be fixed quickly, you might need to temporarily stop the plan from being built. You can achieve this by disabling the plan, which will prevent it from submitting builds to queues under any circumstances.

If a plan is no longer relevant, you have the option to completely delete it from your Bamboo system. Note that deleting a plan will also delete all of the plan's build results, labels and comments.

To delete a build that is currently in progress, see 4.4 Stopping an Active Build.

To disable a plan,

1. Click 'Home' to go to the Dashboard.
2. Click the 'All Plans' tab.
3. Locate the plan in the list and click the plan name.
4. This will display the 'Plan Summary'. Click the 'Build Actions' link (at the right of the page) and select 'Disable Plan':

   ![Build Actions](image)

To delete a plan,

⚠️ Before you begin
If you need to keep a permanent record of the plan's build results, see 7.2 Exporting Data for Backup.

1. Click the 'Administration' link in the top navigation bar.
2. Click the 'Remove Plans' link in the left navigation column.
3. This will display a list of all plans in your Bamboo system. Select the check-box for the plan you wish to delete.
4. Click the 'Delete' button at the bottom of the list.
5. You will be prompted to confirm your deletion.

Screenshot: 'Remove Plans'
## Remove Plans

### Check the plans to be removed

When deleting plans from Bamboo, build history is unrecoverable for the deleted plan.

Check one or more of the following plans to be deleted:

- [ ] Abtazian Bucket (BUCKET)
- [ ] Main Build (MAIN)
- [ ] Abtazian CONFIG (CONFIG)
- [ ] Main Build (MAIN)
- [ ] COVER BUILD (COVER)
- [ ] Abtazian Core (CORE)

[Delete]

---

### RELATED TOPICS

- 1.1 About Projects, Plans and Builds
- 1.2 Creating a Plan
- 1.3 Editing a Plan
- 1.4 Specifying Labels for a Plan's Build Results
- 1.5 Moving a Plan to a different Project
- 1.6 Disabling or deleting a Plan
  - 1.6.1 Deleting a Build Result

[Bamboo Documentation Home]
1.6.1 Deleting a Build Result

This page last changed on Feb 19, 2007 by rosie@atlassian.com.

If a build result is no longer relevant, you have the option to completely delete it from your Bamboo system. Note that you can also automatically delete build results that reach a particular age — see 7.4 Specifying the Expiry Date for Build Results.

To delete a build that is currently in progress, see 4.4 Stopping an Active Build.

To delete a build result,

1. Go to the build result's plan. There are two ways to do this:
   a. Click 'Home' to go to the Dashboard, then click the 'All Plans' tab. Locate the plan in the list and click the plan name.
      OR:
   b. From the build result, click the plan name.
2. This will display the 'Plan Summary'. Click the 'Completed Builds' tab.
3. This will display a list of all completed build results for this plan (see screenshot below). Locate the relevant build result and click the 'Delete' icon: 

Screenshot: Build Results for a Plan

Project: Test  Plan: AntEater (ext)

**Test - AntEater (ext): Completed Build Results**

<table>
<thead>
<tr>
<th>Build Number</th>
<th>Reason</th>
<th>Date</th>
<th>Duration</th>
<th>Test Results</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>TEST-AFT-1</td>
<td>Initial clean build</td>
<td>1 month ago</td>
<td>4 seconds</td>
<td>No tests found!</td>
<td></td>
</tr>
</tbody>
</table>

- Successful builds are green, failed builds are red.
- The plan has been built 1 time.
- The average build time for recent builds is approximately 4 seconds.
- Bamboo builds everything whenever the CVS source code repository changes.
- Feed for all builds or just the failed builds

RELATED TOPICS

- 1.1 About Projects, Plans and Builds
- 1.2 Creating a Plan
- 1.3 Editing a Plan
- 1.4 Specifying Labels for a Plan's Build Results
- 1.5 Moving a Plan to a different Project
- 1.6 Disabling or deleting a Plan
  - 1.6.1 Deleting a Build Result
02. Configuring Build Resources

This page last changed on Feb 07, 2007 by rosie@atlassian.com.

2. Configuring Build Resources

- 2.1 Configuring a new Builder
- 2.2 Configuring a new JDK
- 2.3 Configuring Global Variables
2.1 Configuring a new Builder

A builder is a software compiler program external to Bamboo. Bamboo supports multiple builders. Once a builder is defined in the Bamboo system, it can then be specified in build plans by a Bamboo administrator.

One builder (Maven) is automatically configured when you install Bamboo. You can configure more builders as required.

Bamboo supports the following types of builders:

- Ant
- Maven
- Maven 2
- Custom command (e.g. ‘make’)

Once you have configured a new builder, it will appear in the ‘Builder’ drop-down list on the ‘Build Resources’ tab when you configure a build plan (see 1.2 Creating a Plan).

To configure a new builder,

1. Click the 'Administration' link in the top navigation bar.
2. Click the ‘Builders’ link in the left navigation column.
3. This will display a list of Builders that have been configured in Bamboo, with an empty field at the bottom of each column for adding a new Builder.
4. In the 'Label' field, type the name that will appear in the 'Builders' drop-down list when a plan is configured.
5. In the 'JDK Home Directory' field, enter the location of the JDK Home Directory on your Bamboo server.
6. Select the appropriate 'Type' from the drop-down list.
7. In the 'Path' field, enter the appropriate path. This will vary depending on the 'Type' you selected in step 6; relevant instructions will be shown below the 'Type'.
8. Click the 'Save' button.
9. Verify that 'OK' is displayed in the 'Is Valid?' column for your new Builder. If not, click the 'Edit' link in the 'Operations' column to reconfigure the new Builder’s details.

Screenshot: Configuring a Builder
### Available Builders

You can use this page to view, add and delete the Builders you want made available to projects. The table below shows the Builders currently configured for this server. The label will appear in the build forms Build/Tool dropdown list.

<table>
<thead>
<tr>
<th>Label</th>
<th>Type</th>
<th>Path</th>
<th>Is Valid</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bash</td>
<td>Custom Command</td>
<td>Abinbash</td>
<td>OK</td>
<td>Edit</td>
</tr>
<tr>
<td>Make</td>
<td>Custom Command</td>
<td>/usr/bin/make</td>
<td>OK</td>
<td>Edit</td>
</tr>
<tr>
<td>Maven</td>
<td>Maven 2.x</td>
<td>/opt/javaninja/maven2</td>
<td>OK</td>
<td>Edit</td>
</tr>
<tr>
<td>Ant</td>
<td>Ant</td>
<td>/opt/javaninja/maven</td>
<td>OK</td>
<td>Edit</td>
</tr>
<tr>
<td>MAVEN_HOME</td>
<td>Maven</td>
<td>/opt/javaninja/maven</td>
<td>OK</td>
<td>Edit</td>
</tr>
</tbody>
</table>

Please enter the ANT_HOME value as your path, e.g. /opt/javaninja/1.0.5

### RELATED TOPICS

- [2.1 Configuring a new Builder](#)
- [2.2 Configuring a new JDK](#)
- [2.3 Configuring Global Variables](#)

**Bamboo Documentation Home**
2.2 Configuring a new JDK

When you configure a plan, you need to specify which JDK should be used for the plan's builds. One JDK is automatically configured when you install Bamboo. You can configure more as required.

Once you have configured a new JDK, it will appear in the 'Build JDK' drop-down list on the 'Build Resources' tab when you configure a build plan (see 1.2 Creating a Plan).

To configure a new JDK,

1. Click the 'Administration' link in the top navigation bar.
2. Click the 'JDKs' link in the left navigation column.
3. This will display a list of JDKs that have been configured in Bamboo, with an empty field at the bottom of each column for adding a new JDK.
4. In the 'JDK Label' field, type the name that will appear in the drop-down list when a plan is configured.
5. In the 'JDK Home Directory' field, type the location of the JDK Home Directory on your Bamboo server.
6. Click the 'Save' button.
7. Verify that 'OK' is displayed in the 'Is Valid JDK?' column for your new JDK. If not, click the 'Edit' link in the 'Operations' column to reconfigure the new JDK's details.

### Screenshot: JDKs

You can use this page to view, add and delete System JDKs. The table below shows the JDKs currently configured for this server. The label will appear when configuring build plans.

Bamboo will automatically add the default JAVA_HOME entry if there are no JDKs defined.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1.5 JDK</td>
<td>/usr/java/jdk1.5.0</td>
<td>OK</td>
<td>Edit</td>
</tr>
<tr>
<td>1.4 JDK</td>
<td>/usr/java/jdk1.4.2_05</td>
<td>OK</td>
<td>Edit</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Save</td>
</tr>
</tbody>
</table>

### RELATED TOPICS

- 2.1 Configuring a new Builder
- 2.2 Configuring a new JDK
- 2.3 Configuring Global Variables

[Back to Bamboo Documentation Home]
2.3 Configuring Global Variables

When creating a plan, you can specify variables to be used in the build process. You can specify your variables globally across your Bamboo instance. The values of these variables will then be available on every plan that is built by Bamboo.

To add a new global variable,

1. Click the 'Administration' link in the top navigation bar.
2. Click the 'Global Variables' link in the left navigation column.
3. This will display a list of variables that have been configured in Bamboo, with an empty field at the bottom of each column for adding a new variable.
4. In the 'Key' field, type the key that you will use to identify the variable.
5. In the 'Value' field, type the value of the variable.
6. Click the 'Save' button.

**Screenshot: Global Variables**

<table>
<thead>
<tr>
<th>Builder</th>
<th>Build Queues</th>
<th>JDKs</th>
<th>Build Expire</th>
<th>Remove Plans</th>
<th>Move Plans</th>
<th>Global Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>Builders</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Global Variables</td>
<td>Value</td>
<td>Operations</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bambooVersion</td>
<td>1.1</td>
<td>Save</td>
<td>Delete</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**RELATED TOPICS**

- [2.1 Configuring a new Builder](#)
- [2.2 Configuring a new JDK](#)
- [2.3 Configuring Global Variables](#)

Bamboo Documentation Home
03. Triggering a Build

3. Triggering a Build

- 3.1 About Build Triggering
- 3.2 Triggering a Build when Code is Updated
  - 3.2.1 Polling the Repository for Code Changes
  - 3.2.2 Triggering a Build on Code Check-in
- 3.3 Triggering a Build on Schedule
  - 3.3.1 Scheduling a Single Daily Build
  - 3.3.2 Specifying a Cron-based Schedule
- 3.4 Triggering a Build when another Build finishes
- 3.5 Triggering a Build Manually
3.1 About Build Triggering

There are a variety of ways in which a build can be triggered for a plan:

- Code updated — a build can be triggered whenever one or more authors checks-in code.
- Scheduled build — a build can be scheduled to occur at regular intervals.
- Dependency — a build can be triggered whenever a successful build occurs for another plan.
- Manual build — a build can be triggered manually.
- Initial clean build — a build will be triggered when a new plan is created.

The way in which each build was triggered is listed in the ‘Reason’ column on the Dashboard.

⚠ Considerations for choosing a Build Strategy

- Code updated:
  Triggering a build when code is updated ensures that a build only occurs when necessary.
  There are two ways to trigger a build when code is updated:
  - "Pull strategy" —
    Polling the repository for code changes means that Bamboo will check-out the source-code on a regular basis, and examine it for changes. If Bamboo detects a change, it will trigger a build. See 3.2.1 Polling the Repository for Code Changes.
  - "Push strategy" —
    Triggering a build on code check-in has the advantage of placing minimal load on your Bamboo server, but requires that the repository is configured to fire an event to the Bamboo server. See 3.2.2 Triggering a Build on Code Check-in.

- Scheduled build:
  Triggering a build on schedule can allow a team to structure the day according to a predictable schedule. Note that scheduled builds are run regardless of whether or not any code changes have occurred. There are two ways to schedule a build:
  - Single Daily Build —
    A single daily build runs at a time of your choice. This is particularly suitable for builds that take a long time to complete.
    See 3.3.1 Scheduling a Single Daily Build.
  - Cron-Based Scheduling —
    A cron-based schedule allows you to schedule builds according to a flexible cron expression. For example, "0 0/30 9-19 * MON-FRI" would trigger a build every half-an-hour from 9am to 7pm, Monday to Friday. See 3.3.2 Specifying a Cron-based Schedule.

⚠ Also see 4.4 Stopping an Active Build.

RELATED TOPICS

- 3.1 About Build Triggering
- 3.2 Triggering a Build when Code is Updated
  - 3.2.1 Polling the Repository for Code Changes
3.2.2 Triggering a Build on Code Check-in

3.3 Triggering a Build on Schedule
  3.3.1 Scheduling a Single Daily Build
  3.3.2 Specifying a Cron-based Schedule

3.4 Triggering a Build when another Build finishes

3.5 Triggering a Build Manually

Bamboo Documentation Home
3.2 Triggering a Build when Code is Updated

Triggering a build when code is updated ensures that a build only occurs when necessary. There are two ways to trigger a build when code is updated:

- "Pull strategy" —
  Polling the repository for code changes means that Bamboo will check-out the source-code on a regular basis, and examine it for changes. If Bamboo detects a change, it will trigger a build. See 3.2.1 Polling the Repository for Code Changes.
- "Push strategy" —
  Triggering a build on code check-in has the advantage of placing minimal load on your Bamboo server, but requires that the repository is configured to fire an event to the Bamboo server. See 3.2.2 Triggering a Build on Code Check-in.
3.2.1 Polling the Repository for Code Changes

Polling the repository for code changes means that Bamboo will check-out the source-code on a regular basis, and examine it for changes. If Bamboo detects a change, it will trigger a build. To poll the repository for changes:

1. Click 'Home' to go to the Dashboard.
2. Click the 'All Plans' tab.
3. Locate the plan in the list and click this icon:
4. The 'Configuration' tab will be displayed. Click the 'Source Repository' sub-tab.
5. In the 'Build Strategy' field, select 'Poll the repository for changes' (see screenshot below).
6. In the 'Polling Frequency' field, specify how often (in seconds) Bamboo should check-out the source-code and examine it for changes.
7. Click the 'Save' button.

Screenshot: 'Plan Configuration--Source Repository--Build Strategy: Poll the repository for changes'

RELATED TOPICS
• 3.1 About Build Triggering
• 3.2 Triggering a Build when Code is Updated
  ° 3.2.1 Polling the Repository for Code Changes
  ° 3.2.2 Triggering a Build on Code Check-in
• 3.3 Triggering a Build on Schedule
  ° 3.3.1 Scheduling a Single Daily Build
  ° 3.3.2 Specifying a Cron-based Schedule
• 3.4 Triggering a Build when another Build finishes
• 3.5 Triggering a Build Manually

Bamboo Documentation Home
3.2.2 Triggering a Build on Code Check-in

Triggering a build on code check-in has the advantage of placing minimal load on your Bamboo server, but requires that the repository is configured to fire an event to the Bamboo server.

⚠️ Before you begin:
- If you choose to trigger a build on code check-in, you will need to configure your source-code repository to tell Bamboo whenever a code commit has occurred.
  - For CVS, this is done by editing some files in the CVSROOT module.
  - For Subversion, this is done by editing the Subversion repository's post-commit trigger file.
  - Unfortunately, Perforce repository triggers are currently not supported by Bamboo.

You can access Bamboo's data from an external program by using Bamboo's REST-style remote API. You will first need to [enable the remote API](https://confluence.atlassian.com/display/BAM/Remote+API).

To trigger a build on code check-in,

1. Click 'Home' to go to the Dashboard.
2. Click the 'All Plans' tab.
3. Locate the plan in the list and click this icon: ![edit](https://guide.atlassian.net/u/cre.png)
4. The 'Configuration' tab will be displayed. Click the 'Source Repository' sub-tab.
5. In the 'Build Strategy' field, select 'Repository triggers the build when changes are committed' (see screenshot below).
6. This will display the 'Trigger IP Address' field. If you want Bamboo to receive post-commit notifications from the repository's primary IP address, leave the 'Trigger IP Address' field blank. If you want Bamboo to receive post-commit notifications from a different IP address, type the IP address in the 'Trigger IP Address' field.
7. Click the 'Save' button.

**Screenshot:** 'Plan Configuration--Source Repository-Build Strategy: Repository triggers the build when changes are committed'
## Source Repository

**Repository:**
- CVS

**CVS Root:**
- `@10.15.229.122@` (Unspecified)

**CIMT Path:**
- `file:///home/bruce/bruce` (Unspecified)

**CIMT Directory:**
- `file:///home/bruce/bruce` (Unspecified)

**CIMT File:**
- `file:///home/bruce/bruce` (Unspecified)

**Build Strategy:**
- Repository triggers the build when changes are committed

**Trigger IP Address:**
- `192.168.1.1` (Unspecified)

### Related Topics

- **3.1 About Build Triggering**
- **3.2 Triggering a Build when Code is Updated**
  - 3.2.1 Polling the Repository for Code Changes
  - 3.2.2 Triggering a Build on Code Check-in
- **3.3 Triggering a Build on Schedule**
  - 3.3.1 Scheduling a Single Daily Build
  - 3.3.2 Specifying a Cron-based Schedule
- **3.4 Triggering a Build when another Build finishes**
- **3.5 Triggering a Build Manually**

**Bamboo Documentation Home**
3.3 Triggering a Build on Schedule

Triggering a build on schedule can allow a team to structure the day according to a predictable schedule. Note that scheduled builds are run regardless of whether or not any code changes have occurred. There are two ways to schedule a build:

- **Single Daily Build** — A single daily build runs at a time of your choice. This is particularly suitable for builds that take a long time to complete. See [3.3.1 Scheduling a Single Daily Build](#).
- **Cron-Based Scheduling** — A cron-based schedule allows you to schedule builds according to a flexible cron expression. For example, "0 0/30 9-19 ? * MON-FRI" would trigger a build every half-an-hour from 9am to 7pm, Monday to Friday. See [3.3.2 Specifying a Cron-based Schedule](#).

**RELATED TOPICS**

- [3.1 About Build Triggering](#)
- [3.2 Triggering a Build when Code is Updated](#)
  - [3.2.1 Polling the Repository for Code Changes](#)
  - [3.2.2 Triggering a Build on Code Check-in](#)
- [3.3 Triggering a Build on Schedule](#)
  - [3.3.1 Scheduling a Single Daily Build](#)
  - [3.3.2 Specifying a Cron-based Schedule](#)
- [3.4 Triggering a Build when another Build finishes](#)
- [3.5 Triggering a Build Manually](#)

[Bamboo Documentation Home](#)
3.3.1 Scheduling a Single Daily Build

A single daily build runs at a time of your choice. This is particularly suitable for builds that take a long time to complete.

To schedule a single daily build,

1. Click 'Home' to go to the Dashboard.
2. Click the 'All Plans' tab.
3. Locate the plan in the list and click this icon:
4. The 'Configuration' tab will be displayed. Click the 'Source Repository' sub-tab.
5. In the 'Build Strategy' field, select 'Single Daily Build' (see screenshot below).
6. In the 'Build Time' field, specify the time of day at which the build should run. Specify the time in "hh:mm" format, using a 24-hour clock.
7. Click the 'Save' button.

Screenshot: 'Plan Configuration--Source Repository-Build Strategy: Single daily build'

- Repository: CVS
- CVS Root: 
- Quiet Period: 4
- Module: busket
- Version of Module: HEAD
- Web Repository URL: http://repos.sydex.atlassian.com/0101/manifests/manifests/CVS
- Web Repository Module: busket
- Build Strategy: Single daily build
- Build Time: 21:00

RELATED TOPICS

- 3.1 About Build Triggering
• **3.2 Triggering a Build when Code is Updated**
  o 3.2.1 Polling the Repository for Code Changes
  o 3.2.2 Triggering a Build on Code Check-in
• **3.3 Triggering a Build on Schedule**
  o 3.3.1 Scheduling a Single Daily Build
  o 3.3.2 Specifying a Cron-based Schedule
• **3.4 Triggering a Build when another Build finishes**
• **3.5 Triggering a Build Manually**

Bamboo Documentation Home
3.3.2 Specifying a Cron-based Schedule

This page last changed on Feb 18, 2007 by rosie@atlassian.com.

A cron-based schedule allows you to schedule builds according to a flexible cron expression. For example, "0 0/30 9-19 ? * MON-FRI" would trigger a build every half-an-hour from 9am to 7pm, Monday to Friday.

To specify a cron-based schedule,

1. Click 'Home' to go to the Dashboard.
2. Click the 'All Plans' tab.
3. Locate the plan in the list and click this icon:
4. The 'Configuration' tab will be displayed. Click the 'Source Repository' sub-tab.
5. In the 'Build Strategy' field, select 'Cron Based Schedule' (see screenshot below).
6. In the 'Cron Expression' field, type your cron expression. A cron expression consists of 6 mandatory and one optional field. The fields in sequential order are: seconds, minutes, hours, day-of-month, month, day-of-week and (optional) year. For more information about cron expressions, please see http://www.opensymphony.com/quartz/wikidocs/TutorialLesson6.html.
7. Click the 'Save' button.

Screenshot: 'Plan Configuration--Source Repository-Build Strategy: Cron-based schedule'

Source Repository

Repository:
CVS

CVS Root: *.previuimos@atlassian.com/sourcerepository
The full path to your CVS repository root. Bamboo supports per-user, subdirectory and local repository access methods.

Change password?

Queue Period:

Cron expr. does not apply: How many seconds should Bamboo wait between checks to determine if the changes are complete?

Module:
based

The repository module containing the source code.

Version of module:
HEAD

Which version of the module should Bamboo build?

Web Repository URL:
http://source.repository.atlassian.com/8/1/BambooRepositoryCVS
(Optional) The URL to the build's sourcecode repository.

Web Repository Module:
based

(Optional) The build's repository name if the above Web Repository URL points to multiple repositories.

Build Strategy:
Cron Based Scheduling

How should Bamboo detect that the source repository has changed?

Cron Expression:
0 0/30 9-19 ? * MON-FRI

Specify the details of the schedule using a cron expression.

Save Cancel

RELATED TOPICS
3.1 About Build Triggering
3.2 Triggering a Build when Code is Updated
   3.2.1 Polling the Repository for Code Changes
   3.2.2 Triggering a Build on Code Check-in
3.3 Triggering a Build on Schedule
   3.3.1 Scheduling a Single Daily Build
   3.3.2 Specifying a Cron-based Schedule
3.4 Triggering a Build when another Build finishes
3.5 Triggering a Build Manually

Bamboo Documentation Home
3.4 Triggering a Build when another Build finishes

This page last changed on Feb 18, 2007 by rosie@atlassian.com.

Sometimes you may want to trigger a build when another plan's build has successfully completed. This ensures that changes to one plan's code do not break a dependent build.

For example, there could be two plans in Bamboo:

1. 'Atlassian CORE' — which contains the core code for an application.
2. 'Atlassian PLUGIN' — which contains code for a plugin to the application.

In this scenario, Application-PLUGIN is a dependency of Application-CORE. Any changes to the Atlassian-CORE code should trigger a build of Atlassian-PLUGIN.

If you specify that a build should run when another build successfully finishes, you may want to prevent it from running at other times. You can achieve this by specifying 'manual builds only'.

See 3.5 Triggering a Build Manually.

To trigger a build when another build finishes,

1. Click 'Home' to go to the Dashboard.
2. Click the 'All Plans' tab.
3. Locate the plan whose build should trigger another plan's build (e.g. 'Atlassian-CORE').
4. Click this icon: 
5. The 'Configuration' tab will be displayed. Click the 'Dependencies' sub-tab. This will display a list of every plan in your Bamboo system, as shown in the screenshot below.
6. Click the 'Edit' link.
7. Select the dependent plan (e.g. 'Atlassian-PLUGIN') — that is, the plan for which a build will be triggered when the first plan's build finishes.
8. Click the 'Save' button.

Screenshot: 'Plan Configuration--Source Repository-Build Strategy: Poll the repository for changes'
Atlassian Core - Main Build: Edit Build Configuration

Please select the dependent builds that will be triggered after a successful Main Build build.

- Atlassian Basket - Main Build
- Atlassian Config - Clover Build
- Atlassian Config - Main Build
- Atlassian Event - Main Build
- Atlassian Extras - Main Build
- Atlassian Log Analysis - Default
- Atlassian Maven2 Base Project - Main Build
- Atlassian Plugins - Main Build
- Atlassian Saph - Main Build
- Atlassian Spring - Default
- Atlassian Spring - Main Build

RELATED TOPICS

- 3.1 About Build Triggering
- 3.2 Triggering a Build when Code is Updated
  - 3.2.1 Polling the Repository for Code Changes
  - 3.2.2 Triggering a Build on Code Check-in
- 3.3 Triggering a Build on Schedule
  - 3.3.1 Scheduling a Single Daily Build
  - 3.3.2 Specifying a Cron-based Schedule
- 3.4 Triggering a Build when another Build finishes
- 3.5 Triggering a Build Manually

Bamboo Documentation Home
3.5 Triggering a Build Manually

To start a manual build,

1. Click 'Home' to go to the [Dashboard](#).
2. Locate the relevant [plan](#) and click the 'Check Out and Build' icon:
   - See also 4.4 Stopping an Active Build.

You can specify that a plan should only ever be built manually. This is useful for:

- **Broken builds** — If a build is broken, you may want to temporarily specify 'manual builds only'. This means that a failing build will not be triggered frequently and hence will not take up time and processing power when other builds could be running.

- **Dependent builds** — If you specify that a build should run when another build successfully finishes, you may want to prevent it from running at other times. You can achieve this by specifying 'manual builds only'.

To specify manual builds only,

1. Click 'Home' to go to the Dashboard.
2. Click the 'All Plans' tab.
3. Locate the plan in the list and click this icon: 
4. The 'Configuration' tab will be displayed. Click the 'Source Repository' sub-tab.
5. In the 'Build Strategy' field, select 'Manual builds only' (see screenshot below).
6. Click the 'Save' button.

**Screenshot:** 'Plan Configuration--Source Repository--Build Strategy: Manual builds only'
3.1 About Build Triggering
3.2 Triggering a Build when Code is Updated
  3.2.1 Polling the Repository for Code Changes
  3.2.2 Triggering a Build on Code Check-in
3.3 Triggering a Build on Schedule
  3.3.1 Scheduling a Single Daily Build
  3.3.2 Specifying a Cron-based Schedule
3.4 Triggering a Build when another Build finishes
3.5 Triggering a Build Manually

Bamboo Documentation Home
04. Managing Build Queues

This page last changed on May 08, 2007 by rosie@atlassian.com.

4. Configuring Build Queues

- 4.1 Creating a Build Queue
- 4.2 Assigning a Plan to a Build Queue
- 4.3 Removing a Plan from a Build Queue
- 4.4 Stopping an Active Build
- 4.5 Disabling or deleting a Build Queue
4.1 Creating a Build Queue

This page last changed on Feb 18, 2007 by rosie@atlassian.com.

A build queue controls the sequence of builds. Bamboo administrators can specify how many build queues there are in the system. For example, if there are two build queues, two builds can occur in parallel while subsequent builds will wait in a queue.

One default queue is created when you install Bamboo.

To create a new queue,

1. Click the 'Administration' link in the top navigation bar.
2. Click the 'Build Queues' link in the left navigation column.
3. This will display a list of all queues that currently exist in your Bamboo system. Below the list, in the blank box in the 'Name' column, type the name of the new queue. (Note that this name will be displayed on the Dashboard.)
4. Click the 'Save' button.
5. This will return you to the list of queues. Your new queue will appear in the list (note: new queues are enabled by default).

By default, a new queue will accept builds from all plans. To control which plans may submit builds to each queue, see 4.2 Assigning a Plan to a Build Queue.

Screenshot: 'Build Queue--Add'

Build Queues

You can use this page to view, add and delete build queues.

The table below shows the Build Queues configured for this server. The label is used to identify the build queue on the home page.

<table>
<thead>
<tr>
<th>Name</th>
<th>Allowed builds</th>
<th>Queued Builds</th>
<th>Status</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default Queue</td>
<td>Accepts all builds</td>
<td>0</td>
<td>Enabled</td>
<td>Edit</td>
</tr>
<tr>
<td>Fast Build</td>
<td></td>
<td></td>
<td></td>
<td>Save</td>
</tr>
</tbody>
</table>

RELATED TOPICS

- 4.1 Creating a Build Queue
- 4.2 Assigning a Plan to a Build Queue
- 4.3 Removing a Plan from a Build Queue
- 4.4 Stopping an Active Build
- 4.5 Disabling or deleting a Build Queue

Bamboo Documentation Home
4.2 Assigning a Plan to a Build Queue

A build queue controls the sequence of builds. Bamboo administrators can specify how many build queues there are in the system. For example, if there are two build queues, two builds can occur in parallel while subsequent builds will wait in a queue.

You can assign different plans to different queues. For example, you could assign fast plans (i.e. plans whose builds take relatively little time) to a special 'Fast Plans' queue so that they are not held up by slower plans.

By default, a build queue accept builds from all plans. You can change this by assigning particular builds to a particular queue. If you do this, the queue will only accept builds from plans that are assigned to it.

Note
Make sure that either you have one queue configured to accept 'All builds', or that every plan is assigned to a queue. Otherwise some plans may be unable to submit builds to any queue, which means that those plans would never be built.

To assign a plan to a build queue,

1. Click the 'Administration' link in the top navigation bar.
2. Click the 'Build Queues' link in the left navigation column.
3. This will display a list of all build queues in your Bamboo system. Locate the relevant queue and click the corresponding 'Edit' link in the 'Operations' column.
4. This will display a list of all plans in your Bamboo system, as shown in the screenshot below. Select the check-boxes corresponding to the relevant plan(s).
5. Click the 'Save' button.

After you click the 'Edit' button, you can also change the queue's name if required.

Screenshot: 'Build Queue--Edit'
## Build Queues

You can use this page to view, add and delete build queues.

The table below shows the Build Queues configured for this server. The label is used to identify the build queue on the home page.

<table>
<thead>
<tr>
<th>Name</th>
<th>Allowed builds</th>
<th>Queued Builds</th>
<th>Status</th>
<th>Operations</th>
</tr>
</thead>
</table>
| Fast Builds | Bamboo - HEAD  
Crowd - Crowd JDK 1.5  
JIRA MAIN  
Crowd - Main Build  
JIRA BRANCH MAIN  
Crowd - Crowd JDK 1.4 | 0             | Disabled | Edit | Delete | Enable |
| Queue 2  | Atlassian Config - Clover Build  
Atlassian Config - Main Build  
Atlassian Core - Main Build  
Atlassian Event - Main Build  
Atlassian Extension - Main Build  
Atlassian Log Analysis - Default  
Atlassian Maven2 Base Project - Main Build  
Atlassian Plugin - Main Build | 0             | Disabled | Save | Cancel |

### RELATED TOPICS

- [4.1 Creating a Build Queue](#)
- [4.2 Assigning a Plan to a Build Queue](#)
- [4.3 Removing a Plan from a Build Queue](#)
- [4.4 Stopping an Active Build](#)
- [4.5 Disabling or deleting a Build Queue](#)

Bamboo Documentation Home
4.3 Removing a Plan from a Build Queue

Removing a plan from a build queue means that the plan can no longer submit future builds to that queue. Note, however, that if you remove every individual plan from the queue, then the queue will revert to the default, that is, it will accept builds from all plans.

⚠️ If a plan is currently being built, see also 4.4 Stopping an Active Build.

To remove a plan from a build queue,

1. Click the 'Administration' link in the top navigation bar.
2. Click the 'Build Queues' link in the left navigation column.
3. This will display a list of all build queues in your Bamboo system. For each queue, a list of the plans that are currently assigned to the queue is shown.
4. Locate the relevant queue and click the corresponding 'Edit' link in the 'Operations' column.
5. This will display a list of all plans in your Bamboo system. Plans that are currently assigned to the queue are shown in blue. Hold the <Ctrl> key while you deselect the relevant plan(s).
6. Click the 'Save' button.

Screenshot: 'Build Queue--Edit'

Build Queues

You can use this page to view, add and delete build queues.

The table below shows the Build Queues configured for this server. The label is used to identify the build queue on the home page.

<table>
<thead>
<tr>
<th>Name</th>
<th>Allowed builds</th>
<th>Queued Builds</th>
<th>Status</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fast Builds</td>
<td>Bamboo - HEAD&lt;br&gt;Crowd - Crowd JDK 1.6&lt;br&gt;JIRA MAIN&lt;br&gt;Crowd - Main Build&lt;br&gt;JIRASRANCH MAIN&lt;br&gt;Crowd - Crowd JDK 1.4</td>
<td>0</td>
<td>Disabled</td>
<td>Edit</td>
</tr>
<tr>
<td>Queue 2</td>
<td>Atlassian Config - Clover Build&lt;br&gt;Atlassian Config - Main Build&lt;br&gt;Atlassian Core - Main Build&lt;br&gt;Atlassian Event - Main Build&lt;br&gt;Atlassian Extras - Main Build&lt;br&gt;Atlassian Log Analysis - Default&lt;br&gt;Atlassian Maven2 Base Project - Main Build&lt;br&gt;Atlassian Plugin - Main Build</td>
<td>0</td>
<td>Disabled</td>
<td>Save</td>
</tr>
</tbody>
</table>

RELATED TOPICS

- 4.1 Creating a Build Queue
- 4.2 Assigning a Plan to a Build Queue
- 4.3 Removing a Plan from a Build Queue
• 4.4 Stopping an Active Build
• 4.5 Disabling or deleting a Build Queue

Bamboo Documentation Home
4.4 Stopping an Active Build

This page last changed on Feb 18, 2007 by rosie@atlassian.com.

⚠️ If you are running Bamboo on Windows, it may only be possible to stop an active build by going to Windows Task Manager and ending the relevant processes.

To stop an active build,

1. Click 'Home' to go to the Dashboard.
2. Locate the relevant plan on the 'All Plans' tab
   OR:
   Locate the relevant plan on the 'Current Activity' tab.
3. Click the 'Stop Build' icon:

⚠️ To start a build on demand, see 3.5 Triggering a Build Manually.

⚠️ To stop a plan from submitting builds to a particular queue, see 4.3 Removing a Plan from a Build Queue. Also see 1.6 Disabling or deleting a Plan.

RELATED TOPICS

- 4.1 Creating a Build Queue
- 4.2 Assigning a Plan to a Build Queue
- 4.3 Removing a Plan from a Build Queue
- 4.4 Stopping an Active Build
- 4.5 Disabling or deleting a Build Queue

Bamboo Documentation Home
4.5 Disabling or deleting a Build Queue

This page last changed on Feb 11, 2007 by rosie@atlassian.com.

A build queue controls the sequence of builds. Bamboo administrators can specify how many build queues there are in the system. For example, if there are two build queues, two builds can occur in parallel while subsequent builds will wait in a queue.

You can add new queues as required (see 4.1 Creating a Build Queue), and delete queues which are no longer necessary (see below).

Sometimes you might need to stop Bamboo from building any plans. You could achieve this by disabling all queues. Note that you can also delete/disable individual plans — see 1.6 Disabling or deleting a Plan.

To disable a build queue,

1. Click the 'Administration' link in the top navigation bar.
2. Click the 'Build Queues' link in the left navigation column.
3. This will display a list of all build queues in your Bamboo system. The 'Status' column indicates which plans are currently enabled/disabled.
4. Locate the relevant queue and click the corresponding 'Disable' link in the 'Operations' column.

To delete a build queue,

1. Click the 'Administration' link in the top navigation bar.
2. Click the 'Build Queues' link in the left navigation column.
3. This will display a list of all build queues in your Bamboo system. Locate the relevant queue and click the corresponding 'Delete' link in the 'Operations' column.

Screenshot: 'Build Queue--Delete or Disable'

**Build Queues**

You can use this page to view, add and delete build queues.

The table below shows the Build Queues configured for this server. The label is used to identify the build queue on the home page.

<table>
<thead>
<tr>
<th>Name</th>
<th>Allowed builds</th>
<th>Queued Builds</th>
<th>Status</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fast Builds</td>
<td>JIRA - Main Build</td>
<td>0</td>
<td>Enabled</td>
<td>Edit</td>
</tr>
<tr>
<td></td>
<td>Crowd - Main Build</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>JIRA Branch - Main Build</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Queue 2</td>
<td>Accepts all builds</td>
<td>0</td>
<td>Enabled</td>
<td>Edit</td>
</tr>
<tr>
<td></td>
<td>Atlassian Bucket - Main Build</td>
<td></td>
<td></td>
<td>Save</td>
</tr>
<tr>
<td></td>
<td>Atlassian Config - Clever Build</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Atlassian Config - Main Build</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**RELATED TOPICS**

- [4.1 Creating a Build Queue](#)
• 4.2 Assigning a Plan to a Build Queue
• 4.3 Removing a Plan from a Build Queue
• 4.4 Stopping an Active Build
• 4.5 Disabling or deleting a Build Queue

Bamboo Documentation Home
5. Managing Users and Security

- 5.01 Creating a User
- 5.02 Deleting a User
- 5.03 Creating a Group
- 5.04 Deleting a Group
- 5.05 Adding Users to and removing them from Groups
- 5.06 Granting Administration Privileges to a User
- 5.07 Enabling or disabling Contact Details Display
- 5.08 Enabling or disabling Public Signup
- 5.09 Enabling or disabling Anonymous Access
- 5.10 Working with External User Repositories
  - 5.10.1 Integrating Bamboo with Crowd
  - 5.10.2 Integrating Bamboo with LDAP
5.01 Creating a User

An author is any person who checks-in code to a repository that is associated with a Bamboo plan. An author need not be a Bamboo user.

Depending on your organisation's needs, you can Bamboo configure to grant access to non-users. However, only Bamboo users can:

- view the 'My Bamboo' tab on the Dashboard.
- add comments or labels to a build result.
- belong to a group.

To create a Bamboo user,

1. Click the 'Administration' link in the top navigation bar.
2. Click the 'Users' link in the left navigation column.
3. The 'Manage Users' screen will be displayed. The 'Create User' section (as shown below) will be displayed at the bottom of the 'Manage Users' screen.
4. In the 'Username' field, type the user's login name.
   - Note that the Username cannot be changed after the user is created.
5. In the 'Password' and 'Confirm Password' fields, type the user's password.
   - The user can easily change their password later.
6. In the 'Full Name' field, type the user's display-name.
7. In the 'Email' field, type the user's email address. This address is where the user will receive password notifications.
8. (optional) In the 'Jabber address' field, type the user's Instant Messaging (IM) address. This address is where the user will receive any group-based notifications about build results.
   - If no IM address is specified, Bamboo will not be able to recognise the user's context when interacting via IM.
9. Select at least one group from the 'Groups' list. (To select multiple groups, press the <Ctrl> key.)
10. If the user is a Bamboo author, select 'Add Alias' (instead of 'None') in the 'Source Repository Alias' field. This will display the 'New alias' field. Type the user's login name for their source-code repository.
    - If you don't know the user's login name for their source-code repository, they can specify it themselves later.
11. Click the 'Save' button.

If you have configured SMTP email on your Bamboo server, the new user will automatically receive an email containing their Bamboo login name and password.

Screenshot: Create User
Add User

User Details

Enter the details of the user to add to Bamboo, then click Save.

Username: * jsmith
The name to use to login to Bamboo.

Password: * 

Confirm Password: * 

Full Name: * Jill Smith

Email: * jsmith@mycompany.com

Jabber Address: 
The address to which instant messages will be sent if they are enabled for a build.

Groups: 
- bamboo-admin
- project

Each user must be in at least one group.

Source Repository Alias: Add Alias 
The repository name the user commits with in the repositories.

New alias: jills

Save

RELATED TOPICS

- 5.01 Creating a User
- 5.02 Deleting a User
- 5.03 Creating a Group
- 5.04 Deleting a Group
- 5.05 Adding Users to and removing them from Groups
- 5.06 Granting Administration Privileges to a User
- 5.07 Enabling or disabling Contact Details Display
- 5.08 Enabling or disabling Public Signup
- 5.09 Enabling or disabling Anonymous Access
- 5.10 Working with External User Repositories
  - 5.10.1 Integrating Bamboo with Crowd
  - 5.10.2 Integrating Bamboo with LDAP

Bamboo Documentation Home
5.02 Deleting a User

This page last changed on May 08, 2007 by rosie@atlassian.com.

Note that deleting a Bamboo user will not delete their author data — that is, their author statistics and code check-in comments will still exist in Bamboo.

Also note that:

- You cannot delete a user who has created labels or comments about build results.
- You cannot delete a user who is the last member of the bamboo-admin group.
- You cannot delete the user account with which you are currently logged in to Bamboo.

To delete a Bamboo user,

1. Click the 'Administration' link in the top navigation bar.
2. Click the 'Users' link in the left navigation column.
3. The 'Manage Users' screen will be displayed. Locate the relevant user in the list, and click the corresponding 'Delete' link in the 'Operations' column.

Screenshot: Delete User

**Manage Users**

You can use this page to view, update and delete users.

<table>
<thead>
<tr>
<th>Username</th>
<th>Email</th>
<th>Full Name</th>
<th>Groups</th>
<th>Repository Alias</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>admin</td>
<td><a href="mailto:admin@mycompany.com">admin@mycompany.com</a></td>
<td>admin</td>
<td>bamboo-admin</td>
<td></td>
<td>Edit</td>
</tr>
<tr>
<td>jsmith</td>
<td><a href="mailto:jsmith@mycompany.com">jsmith@mycompany.com</a></td>
<td>Jill Smith</td>
<td>project-x</td>
<td>jills</td>
<td>Edit, Delete</td>
</tr>
</tbody>
</table>
5.03 Creating a Group

This page last changed on May 08, 2007 by rosie@atlassian.com.

Bamboo groups are used to specify which users will receive notifications about a plan's build results. You can create and delete as many groups as you need. You will typically create at least one group per project.

Additionally, a special group called bamboo-admin is automatically created when you install Bamboo. Members of this group have the ability to perform Bamboo administration. To create a group,

1. Click the 'Administration' link in the top navigation bar.
2. Click the 'Groups' link in the left navigation column.
3. The 'Manage Groups' screen will be displayed. The 'Create Group' section (as shown below) will be displayed at the bottom of the 'Manage Groups' screen.
4. In the 'Group Name' field, type a name for your new group.
   - Note that the Group Name cannot be changed after the group is created.
5. Select relevant users from the 'Users in Group' list. Hold the <Ctrl> to select multiple users.
   - You can also add or remove users from the group later if required.
6. Click the 'Save' button.

Screenshot: Create Group

Create Group

Group Details

Enter the details of the group to add in Bamboo, then click Save.

Group Name: `project-x`

Users in Group:

- admin
- andreas
- sin
- ben.kuo
- bmoody
- booyah
- christopher
- cmiller
- cavo
- don.willis
- dushan
- erwin
- ernest
- jason
- jad
- jans
- jstepka
- justin
- mark
- matt

Save
• 5.01 Creating a User
• 5.02 Deleting a User
• 5.03 Creating a Group
• 5.04 Deleting a Group
• 5.05 Adding Users to and removing them from Groups
• 5.06 Granting Administration Privileges to a User
• 5.07 Enabling or disabling Contact Details Display
• 5.08 Enabling or disabling Public Signup
• 5.09 Enabling or disabling Anonymous Access
• 5.10 Working with External User Repositories
  • 5.10.1 Integrating Bamboo with Crowd
  • 5.10.2 Integrating Bamboo with LDAP

Bamboo Documentation Home
5.04 Deleting a Group

This page last changed on May 08, 2007 by rosie@atlassian.com.

Bamboo groups are used to specify which users will receive notifications about a plan's build results. You can create and delete as many groups as you need. You will typically create at least one group per project.

Additionally, a special group called bamboo-admin is automatically created when you install Bamboo. Members of this group have the ability to perform Bamboo administration.

The bamboo-admin group cannot be deleted.

To delete a group,

1. Click the 'Administration' link in the top navigation bar.
2. Click the 'Groups' link in the left navigation column.
3. The 'Manage Groups' screen will be displayed. Locate the relevant group in the list, and click the corresponding 'Delete' link in the 'Operations' column.

Screenshot: Delete Group

Manage Group

You can use this page to view and delete groups.

<table>
<thead>
<tr>
<th>Group</th>
<th>Number of Users</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>bamboo-admin</td>
<td>29</td>
<td>Edit</td>
</tr>
<tr>
<td>crowd-developers</td>
<td>3</td>
<td>Edit/Delete</td>
</tr>
</tbody>
</table>

RELATED TOPICS

- 5.01 Creating a User
- 5.02 Deleting a User
- 5.03 Creating a Group
- 5.04 Deleting a Group
- 5.05 Adding Users to and removing them from Groups
- 5.06 Granting Administration Privileges to a User
- 5.07 Enabling or disabling Contact Details Display
- 5.08 Enabling or disabling Public Signup
- 5.09 Enabling or disabling Anonymous Access
- 5.10 Working with External User Repositories
- 5.10.1 Integrating Bamboo with Crowd
- 5.10.2 Integrating Bamboo with LDAP

Bamboo Documentation Home
5.05 Adding Users to and removing them from Groups

Bamboo groups are used to specify which users will receive notifications about a plan's build results. You can create and delete as many groups as you need. You will typically create at least one group per project.

Additionally, a special group called bamboo-admin is automatically created when you install Bamboo. Members of this group have the ability to perform Bamboo administration.

To add users to a group,

1. Click the 'Administration' link in the top navigation bar.
2. Click the 'Groups' link in the left navigation column.
3. The 'Manage Groups' screen will be displayed. Locate the relevant group in the list, and click the corresponding 'Edit' link in the 'Operations' column.
4. The 'Edit Group Details' screen will be displayed. Users who already belong to the group are shown in blue; users who do not currently belong to the group are shown in white. Press the <Ctrl> key and hold it while you select the user(s) whom you want to add to the group.
5. Click the 'Save' button.

Screenshot: Edit Group Details

<table>
<thead>
<tr>
<th>Group Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Update the details of the group in Bamboo, then click Save.</td>
</tr>
<tr>
<td>Group Name: bamboo-admin</td>
</tr>
<tr>
<td>Users in Group:</td>
</tr>
<tr>
<td>admin, andrew, ben, booyah, christopher, dave, dusan, ernest, jason, jens, jstokes, justin, mark, mat</td>
</tr>
</tbody>
</table>
To remove users from a group,

1. Click the 'Administration' link in the top navigation bar.
2. Click the 'Groups' link in the left navigation column.
3. The 'Manage Groups' screen will be displayed. Locate the relevant group in the list, and click the corresponding 'Edit' link in the 'Operations' column.
4. The 'Edit Group Details' screen will be displayed. Users who belong to the group are shown in blue. Press the <Ctrl> key and hold it while you deselect the user(s) whom you want to remove from the group.
5. Click the 'Save' button.

⚠️ You cannot remove a user from the bamboo-admin group if they are the only member.

RELATED TOPICS

- 5.01 Creating a User
- 5.02 Deleting a User
- 5.03 Creating a Group
- 5.04 Deleting a Group
- 5.05 Adding Users to and removing them from Groups
- 5.06 Granting Administration Privileges to a User
- 5.07 Enabling or disabling Contact Details Display
- 5.08 Enabling or disabling Public Signup
- 5.09 Enabling or disabling Anonymous Access
- 5.10 Working with External User Repositories
  - 5.10.1 Integrating Bamboo with Crowd
  - 5.10.2 Integrating Bamboo with LDAP

Bamboo Documentation Home
5.06 Granting Administration Privileges to a User

Members of the group 'bamboo-admin' have administration privileges — that is, the ability to perform the functions described in the Bamboo Administrator's Guide.

To grant administration privileges to a user,

1. Click the 'Administration' link in the top navigation bar.
2. Click the 'Users' link in the left navigation column.
3. The 'Manage Users' screen will be displayed. Locate the user in the list of groups and click the corresponding 'Edit' link in the 'Operations' column.
4. The 'Edit Users' screen will be displayed. Groups to which the user belongs are shown in blue; groups to which the user does not belong are shown in white. If 'bamboo-admin' is white, press the <Ctrl> key and hold it while you click 'bamboo-admin'.
5. Click the 'Save' button.
6. The 'Manage Users' screen will be displayed. Verify that 'bamboo-admin' is now included in the list of Groups to which the user belongs.

RELATED TOPICS

- 5.01 Creating a User
- 5.02 Deleting a User
- 5.03 Creating a Group
- 5.04 Deleting a Group
- 5.05 Adding Users to and removing them from Groups
- 5.06 Granting Administration Privileges to a User
- 5.07 Enabling or disabling Contact Details Display
- 5.08 Enabling or disabling Public Signup
- 5.09 Enabling or disabling Anonymous Access
- 5.10 Working with External User Repositories
  - 5.10.1 Integrating Bamboo with Crowd
  - 5.10.2 Integrating Bamboo with LDAP

Bamboo Documentation Home
5.07 Enabling or disabling Contact Details Display

If you enable contact details display to your Bamboo system, the full contact details for a user, including email address, IM address, and group, will be visible to any visitors of Bamboo. The email addresses of administrators in the 'Contact Administrators' page will also be visible.

To enable contact details display,

1. Click the 'Administration' link in the top navigation bar.
2. Click the 'Security Settings' link in the left navigation column.
3. Select the 'Enable contact details to be displayed?' check-box.
4. Click the 'Save' button.

To disable contact details display,

1. Click the 'Administration' link in the top navigation bar.
2. Click the 'Security Settings' link in the left navigation column.
3. Deselect the 'Enable contact details to be displayed?' check-box.
4. Click the 'Save' button.

RELATED TOPICS

- 5.01 Creating a User
- 5.02 Deleting a User
- 5.03 Creating a Group
- 5.04 Deleting a Group
- 5.05 Adding Users to and removing them from Groups
- 5.06 Granting Administration Privileges to a User
- 5.07 Enabling or disabling Contact Details Display
- 5.08 Enabling or disabling Public Signup
- 5.09 Enabling or disabling Anonymous Access
- 5.10 Working with External User Repositories
  - 5.10.1 Integrating Bamboo with Crowd
  - 5.10.2 Integrating Bamboo with LDAP

Bamboo Documentation Home
5.08 Enabling or disabling Public Signup

This page last changed on May 08, 2007 by rosie@atlassian.com.

If you enable signup for your Bamboo system, visitors can create their own Bamboo user accounts. Note that they cannot grant themselves administration rights.

To enable signup,

1. Click the 'Administration' link in the top navigation bar.
2. Click the 'Security Settings' link in the left navigation column.
3. Select the 'Enable Signup?' check-box.
4. Click the 'Save' button.
5. Log out of Bamboo and verify that the top navigation bar now contains a 'Signup' link (see screenshot below).

To disable signup,

1. Click the 'Administration' link in the top navigation bar.
2. Click the 'Security Settings' link in the left navigation column.
3. Deselect the 'Enable Signup?' check-box.
4. Click the 'Save' button.

Screenshot: 'Signup'

<table>
<thead>
<tr>
<th>Username:</th>
<th></th>
<th>The name to use to login to Bamboo.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Password:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Confirm Password:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Full Name:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Email:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Jabber Address</td>
<td></td>
<td>The address to which instant messages will be sent to if they are enabled for a build.</td>
</tr>
</tbody>
</table>

Create an account and login to Bamboo

RELATED TOPICS
- 5.01 Creating a User
- 5.02 Deleting a User
- 5.03 Creating a Group
- 5.04 Deleting a Group
- 5.05 Adding Users to and removing them from Groups
- 5.06 Granting Administration Privileges to a User
- 5.07 Enabling or disabling Contact Details Display
- 5.08 Enabling or disabling Public Signup
- 5.09 Enabling or disabling Anonymous Access
- 5.10 Working with External User Repositories
  - 5.10.1 Integrating Bamboo with Crowd
  - 5.10.2 Integrating Bamboo with LDAP

Bamboo Documentation Home
5.09 Enabling or disabling Anonymous Access

This page last changed on May 08, 2007 by rosie@atlassian.com.

If you enable anonymous access to your Bamboo system, visitors who aren't logged in to Bamboo will be able to perform most of the functions described in the Bamboo User's Guide (e.g. generating reports; viewing plans and build results).

Note that people who access Bamboo without logging in will not have a 'My Bamboo' tab on the Dashboard.

To enable anonymous access,

1. Click the 'Administration' link in the top navigation bar.
2. Click the 'Security Settings' link in the left navigation column.
4. Click the 'Save' button.

To disable anonymous access,

1. Click the 'Administration' link in the top navigation bar.
2. Click the 'Security Settings' link in the left navigation column.
3. Deselect the 'Enable Anonymous Access?' check-box.
4. Click the 'Save' button.

RELATED TOPICS

- 5.01 Creating a User
- 5.02 Deleting a User
- 5.03 Creating a Group
- 5.04 Deleting a Group
- 5.05 Adding Users to and removing them from Groups
- 5.06 Granting Administration Privileges to a User
- 5.07 Enabling or disabling Contact Details Display
- 5.08 Enabling or disabling Public Signup
- 5.09 Enabling or disabling Anonymous Access
- 5.10 Working with External User Repositories
  - 5.10.1 Integrating Bamboo with Crowd
  - 5.10.2 Integrating Bamboo with LDAP

Bamboo Documentation Home
5.10 Working with External User Repositories

You can integrate external user repositories with Bamboo:

- 5.10.1 Integrating Bamboo with Crowd
- 5.10.2 Integrating Bamboo with LDAP

RELATED TOPICS

- 5.01 Creating a User
- 5.02 Deleting a User
- 5.03 Creating a Group
- 5.04 Deleting a Group
- 5.05 Adding Users to and removing them from Groups
- 5.06 Granting Administration Privileges to a User
- 5.07 Enabling or disabling Contact Details Display
- 5.08 Enabling or disabling Public Signup
- 5.09 Enabling or disabling Anonymous Access
- 5.10 Working with External User Repositories
  - 5.10.1 Integrating Bamboo with Crowd
  - 5.10.2 Integrating Bamboo with LDAP

Bamboo Documentation Home
5.10.1 Integrating Bamboo with Crowd

Atlassian's Crowd identity management system can be integrated with Bamboo. Please see the document Integrating Crowd with Bamboo in the Crowd Administrator's Guide.

RELATED TOPICS

- 5.01 Creating a User
- 5.02 Deleting a User
- 5.03 Creating a Group
- 5.04 Deleting a Group
- 5.05 Adding Users to and removing them from Groups
- 5.06 Granting Administration Privileges to a User
- 5.07 Enabling or disabling Contact Details Display
- 5.08 Enabling or disabling Public Signup
- 5.09 Enabling or disabling Anonymous Access
- 5.10 Working with External User Repositories
  - 5.10.1 Integrating Bamboo with Crowd
  - 5.10.2 Integrating Bamboo with LDAP

Bamboo Documentation Home
5.10.2 Integrating Bamboo with LDAP

This page last changed on May 08, 2007 by asridhar.

⚠️ Before you begin:
Warning: Before proceeding with LDAP integration, please back up your data.

Bamboo at this stage can be integrated with LDAP. This functionality is limited to authenticating and authorisation of users. Bamboo does not let users manage/administer LDAP accounts or user groups.

However, Bamboo will continue to use locally created users, even when LDAP is enabled. This means that locally created users and groups can still be managed via Bamboo.

Also, note that Bamboo does not support multiple LDAP servers. Once LDAP has been enabled on Bamboo, you cannot revert back to a local user management in Bamboo.

Stage 1 - Configure Connection Details

The LDAP server connection is specified by manually editing the file atlassian-user.xml.

1. Edit the file .../webapp/WEB-INF/classes/atlassian-user.xml and configure the connection AD or LDAP.
2. Check your configuration against the example connection details shown below.

```xml
<ldap key="ldapRepository" name="LDAP Repository@hecate.atlassian.com" cache="true">
  <host>hecate.atlassian.com</host>
  <port>389</port>
  <securityPrincipal>cn=admin,dc=atlassian,dc=private</securityPrincipal>
  <securityCredential>secret</securityCredential>
  <securityProtocol>plain</securityProtocol>
  <securityAuthentication>simple</securityAuthentication>
  <baseContext>dc=atlassian,dc=private</baseContext>
...
```

3. Please ensure that the following line is also active in your atlassian-user.xml (it should be there by default):

```xml
<hibernate name="Hibernate Repository" key="hibernateRepository" description="Hibernate Repository" />
```

Stage 2 - Map LDAP Data Tree

1. To configure the mappings in atlassian-user.xml for either AD or LDAP, please see:
   - Mapping Active Directory
   - Mapping other LDAP servers
2. Check your configuration against the example connection details shown below.

```xml
...<baseUserNamespace>dc=staff,dc=perftest,dc=atlassian,dc=private</baseUserNamespace>
<baseGroupNamespace>dc=groups,dc=perftest,dc=atlassian,dc=private</baseGroupNamespace>
```
Stage 3 - Optional LDAP Settings

The following settings do not appear in the default atlassian-user.xml file:

```xml
<poolingOn>true</poolingOn>
<maxSize>0</maxSize>
<initSize>10</initSize>
<prefSize>10</prefSize>
<debugLevel>none</debugLevel>
<securityProtocol>plain, ssl</securityProtocol>
<authentication>simple</authentication>
<initialContextFactory>com.sun.jndi.ldap.LdapCtxFactory</initialContextFactory>
<batchSize>100</batchSize>
<timeToLive>0</timeToLive>
<userSearchAllDepths>true</userSearchAllDepths>
<groupSearchAllDepths>true</groupSearchAllDepths>
</ldap>
```

However, if you want to override the default values listed above, you can add the value onto the end like so:

```xml
...
<groupnameAttribute>cn</groupnameAttribute>
<groupSearchFilter>(objectClass=groupOfNames)</groupSearchFilter>
<membershipAttribute>member</membershipAttribute>
<initSize>20</initSize>
</ldap>
```

It is important that the connection pool timeout value be set to 0, as this will force Atlassian User (via the JNDI layer) to clean up lingering connections that have lived past one request. For more information about LDAP pools please see [http://java.sun.com/products/jndi/tutorial/ldap/connect/config.html](http://java.sun.com/products/jndi/tutorial/ldap/connect/config.html).

Stage 4 - Assigning LDAP Users to Bamboo Groups

Once Bamboo is started with LDAP enabled, you can assign LDAP users to Bamboo groups. Please see Adding Users to and removing them from Groups.

RELATED TOPICS

- 5.01 Creating a User
- 5.02 Deleting a User
• 5.03 Creating a Group
• 5.04 Deleting a Group
• 5.05 Adding Users to and removing them from Groups
• 5.06 Granting Administration Privileges to a User
• 5.07 Enabling or disabling Contact Details Display
• 5.08 Enabling or disabling Public Signup
• 5.09 Enabling or disabling Anonymous Access
• 5.10 Working with External User Repositories
  • 5.10.1 Integrating Bamboo with Crowd
  • 5.10.2 Integrating Bamboo with LDAP

Bamboo Documentation Home
6. Configuring Email and Instant Messaging Notifications

- 6.1 Enabling or disabling Notifications for a Plan
- 6.2 Configuring Bamboo to send SMTP Email
- 6.3 Configuring Bamboo to use Instant Messaging (IM)
  - 6.3.1 Configuring Bamboo to use Google Talk for Instant Messaging
6.1 Enabling or disabling Notifications for a Plan

You can specify which people will receive notifications about build results for a particular plan, and under what circumstances (known as 'Notification Triggers'), i.e.:

<table>
<thead>
<tr>
<th>Notification Trigger</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>'All Completed Builds'</td>
<td>Bamboo will send a notification whenever a build finishes for this plan, regardless of the build result. This trigger is recommended for any plans for which it is critical that people are always informed about the latest build activity. Many organisations start with this trigger, then change it later as they get more confidence in the continuous build process.</td>
</tr>
<tr>
<td>'Failed Builds And First Successful'</td>
<td>Bamboo will send a notification whenever:</td>
</tr>
<tr>
<td></td>
<td>• a build fails for this plan.</td>
</tr>
<tr>
<td></td>
<td>• the plan is 'fixed' (that is, the plan's latest build is successful and the previous build failed). This trigger is generally suitable for the majority of plans.</td>
</tr>
<tr>
<td>After X Failed Builds'</td>
<td>This trigger enables you to specify the 'Number Of Failed Builds' after which Bamboo will send a notification. This is a useful way of limiting the number of notifications, if you are concerned about people receiving too many.</td>
</tr>
</tbody>
</table>

For each plan, you can specify different recipients for each Notification Trigger. Note also that recipients need not be people with Bamboo user accounts.

⚠️ Before you begin
You need to configure Bamboo's system email and/or instant messaging capabilities before Bamboo can send notifications.

To enable notifications for a plan,

1. Click 'Home' to go to the Dashboard.
2. Click the 'All Plans' tab.
3. Locate the plan in the list and click this icon: 🖋
4. The 'Configuration' tab will be displayed. Click the 'Post Actions' tab.
5. A list of 'Notification Triggers' will be displayed. Select the 'Notification Trigger' you wish to enable, then specify any or all of the following recipients:
   • 'Roles' — Select from the following:
     ○ 'Committer' — The Bamboo user(s) who have committed to this build (i.e. users who committed code after the previous build was checked out by Bamboo).
     ○ 'Watcher' — All Bamboo users who have marked this plan as one of their favourites.
   • 'Groups' — Type the names of the appropriate Bamboo group(s), separated by commas.
• 'Users' — Type the usernames of the appropriate Bamboo users, separated by commas; or click the following icon to select from a list of users:

• 'Email Addresses' — This is useful if you need to send email notifications to people who are not Bamboo users. Type the appropriate email addresses, separated by commas.

• 'Instant Messaging Addresses' — This is useful if you need to send IM notifications to people who are not Bamboo users. Type the appropriate IM addresses, separated by commas. Note that if you specify a broadcast address (e.g., 'project-x@broadcast.chat.mycompany.com'), Bamboo will not know the context of related IM responses.

6. Click the 'Add' button.
7. Repeat steps 5 and 6 until you have added all the Notification Triggers that you wish to enable for this plan.
8. Click the 'Done' button.

Each Bamboo user can choose whether to receive their notifications via email, IM, both or neither.

Screenshot: ‘Plan Configuration-Post Actions’

Add Build Notification

Notification Trigger:  All Completed Builds

Roles:  Committer

Groups:

Users:

Email Addresses:

Instant Messaging Addresses:

Operations

To disable notifications for a plan,

1. Click 'Home' to go to the Dashboard.
2. Click the 'All Plans' tab.
3. Locate the plan in the list and click this icon:
4. The 'Configuration' tab will be displayed. Click the 'Build Notifications' tab.
5. For each 'Notification Trigger', click the corresponding 'Remove' link in the 'Operations' column.
6. Click the 'Done' button.

Bamboo will no longer send notifications about this plan's build results.

RELATED TOPICS

- 6.1 Enabling or disabling Notifications for a Plan
- 6.2 Configuring Bamboo to send SMTP Email
- 6.3 Configuring Bamboo to use Instant Messaging (IM)
  - 6.3.1 Configuring Bamboo to use Google Talk for Instant Messaging

Bamboo Documentation Home
6.2 Configuring Bamboo to send SMTP Email

This page last changed on Apr 23, 2007 by rosie@atlassian.com.

Bamboo can send email notifications about build results. There are two steps to setting this up:

1. Configure Bamboo to send SMTP email (see below).
2. Configure a plan to send SMTP email notifications about build results (see 6.1 Enabling or disabling Notifications for a Plan).

To configure Bamboo to send SMTP email,

1. Click the 'Administration' link in the top navigation bar.
2. Click the 'Mail Server' link in the left navigation column (under 'Communication').
3. This will display the 'Mail Server Details' page (see screenshot below). Click the 'Edit' button.
4. In the 'Name' field, type a display-name for the email address in step 5 (below).
5. In the 'From Address' field, type the email address from which Bamboo notifications will be sent.
6. In the 'Subject Prefix' field, type the text (if any) with which you would like the email Subject line to begin. E.g. if you type '[Bamboo]', then people will receive emails with Subjects like this:
   - '[Bamboo] TEST build 1,001 has FAILED (77 tests failed, no failures were new) : Change made by jsmith'
   - '[Bamboo] TEST build 1,002 was SUCCESSFUL (with 77 tests) : Change made by jsmith'
7. If you are not using JNDI1:
   a. In the 'SMTP Server' field, type the name of the email server via which Bamboo notifications will be sent. E.g. 'mail.myserver.com'.
   b. In the 'Username' field, type the login name of the account which Bamboo will use to login to the SMTP server.
   c. In the 'Password' field, type the password for the account specified in step 7 (if any).
   d. Go to step 8.
8. If you are using JNDI1, type the JNDI name in the 'JNDI Location' field. The JNDI Location will depend on your application server, and on the location of the 'mail' resource within the JNDI tree you specify. E.g. 'java:comp/env/mail/BambooMailServer'.
9. Type a test email address in the 'Test Recipient Address' box.
10. Click the 'Test' button, and verify that a test email is received.
11. Click the 'Save' button.

1 Note re JNDI: As an alternative to specifying mail details directly in Bamboo, you can configure them in your application server (e.g. in the server.xml file — see 8.1 Locating Important Directories and Files), and then use JNDI to look up a preconfigured mail session. JNDI has the following advantages:

   - Centralised management - mail details are configured in the same place as database details, and may be configured through your application server administration tools.
   - Better security - mail details are not available to Bamboo administrators through the Bamboo interface, and aren't stored in Bamboo backup files.
   - More SMTP options - e.g. SSL. If you want to use SMTP over SSL you will need to use JNDI.

Screenshot: 'Email Server Details'
Configure Mail Server Details

What email settings should Bamboo use? Please enter either a SMTP Server hostname, or the JNDI location of a javax.mail.Session object to use.

Name: * Bamboo
Provide a name for the email server (this will be displayed on notification emails).

From Address: * bamboo@mycompany.com
This is the email address Bamboo-generated emails are sent from.

Subject Prefix: [Bamboo]
This is the tag added to the start of the subject line to identify Bamboo-generated email, e.g. [Bamboo]

SMTP Server: * localhost
Specify the email server used to send out the build emails. For example, "mail.mydomain.com".

Username:

Password:

JNDI Location:

Test Email Configuration
You can enter a test recipient email address below and click Test. Bamboo will test whether this email setting is valid by sending a test email to the specified address(es).

Test Recipient Address (optional):
You can enter one or more, comma separated email address to which Bamboo will send a test email.

Save Test Cancel

Next step
Now that you have configured Bamboo's SMTP email capability, you can specify notifications for a plan.

RELATED TOPICS

- 6.1 Enabling or disabling Notifications for a Plan
- 6.2 Configuring Bamboo to send SMTP Email
- 6.3 Configuring Bamboo to use Instant Messaging (IM)
  - 6.3.1 Configuring Bamboo to use Google Talk for Instant Messaging
6.3 Configuring Bamboo to use Instant Messaging (IM)

This page last changed on Feb 18, 2007 by rosie@atlassian.com.

Bamboo can send Instant Messaging (IM) notifications about build results. There are two steps to setting this up:

1. Configure Bamboo to use Instant Messaging (see below).
2. Configure a plan to send IM notifications about build results (see 6.1 Enabling or disabling Notifications for a Plan).

To configure Bamboo to use Instant Messaging,

1. Click the 'Administration' link in the top navigation bar.
2. Click the 'IM Server' link in the left navigation column (under 'Communication').
3. This will display the 'Instant Messaging Server Details' page. Click the 'Edit' button.
4. In the 'Host' field, type the address of your IM server (for example, 'chat.atlassian.com').
5. In the 'Port' field, type the TCP port that your organisation uses for IM traffic (or leave this field blank to have Bamboo either perform a DNS lookup or use the default port).
6. In the 'Username' field, type the login name of the IM account from which Bamboo notifications will be sent.
7. In the 'Password' field, type the password for the account specified in step 6.
8. If your IM server uses SSL, select the 'Requires an SSL Connection' check-box.
9. Type a test IM user's address in the 'Test Recipient Address' box.
10. Click the 'Test' button, and verify that a test IM message is received.
11. Click the 'Save' button.

Screenshot: 'Instant Messaging Server Details'
Add an Instant Messaging Server

Add an Instant Messaging Server

Enter the details of the instant messaging server to add in Bamboo, then click Save. Currently only XMPP (such as Jabber, LiveMessenger) is supported.

Host

For example "chat.myserver.com".

Port

If no port is specified, Bamboo will first perform a DNS SRV lookup or use the default port.

Username:

Password:

Required an SSL connection

Test Instant Messaging Server Configuration

Enter recipient address below. Bamboo will test whether this instant messaging server setting is valid by sending a test message to the specified recipient(s).

Test Recipient Address

You can enter one or more, comma separated instant messaging address to which Bamboo will send a test instant message.

Save  Test  Cancel

Next step

Now that you have configured Bamboo's IM capability, you can specify notifications for a plan.

RELATED TOPICS

- 6.1 Enabling or disabling Notifications for a Plan
- 6.2 Configuring Bamboo to send SMTP Email
- 6.3 Configuring Bamboo to use Instant Messaging (IM)
  - 6.3.1 Configuring Bamboo to use Google Talk for Instant Messaging

Bamboo Documentation Home
6.3.1 Configuring Bamboo to use Google Talk for Instant Messaging

This page last changed on Apr 02, 2007 by asridhar.

If your Bamboo server has access to the internet, it can use Google Talk to send IM notifications about build results.

To configure Bamboo to use Google Talk for Instant Messaging,

1. Click the 'Administration' link in the top navigation bar.
2. Click the 'IM Server' link in the left navigation column (under 'Communication').
3. This will display the 'Instant Messaging Server Details' page. Click the 'Edit' button.
4. In the 'Host' field, type 'gmail.com'.
5. In the 'Port' field blank, type the TCP port that your organisation uses for IM traffic (or leave this field blank to have Bamboo either perform a DNS lookup or use the default port).
6. In the 'Username' field, type the login name of the Google account from which IM notifications will be sent. Only the account name needs to be included e.g. atlassianbamboo NOT atlassianbamboo@gmail.com.
7. Do not select "Requires an SSL connection"
8. In the 'Password' field, type the password for the account specified in step 6.
9. Type a test IM user's address (e.g. atlassianbamboo@gmail.com NOT atlassianbamboo) in the 'Test Recipient Address' box. (Note: use a different user to the one you specified in step 6.)
10. Click the 'Test' button, and verify that the message is successfully received.
11. Click the 'Save' button.

Google Talk does not allow IM messages to be received unless the receiver has approved the sender. Please ensure that the Gmail user specified in step 6 is approved by each Google Talk recipient. That is, ensure that the 'Host' and 'Username' have previously sent messages to each other via Google Talk.

Additional notes about using Google Talk:

- The Google Talk service is hosted at talk.google.com. The default port is 5222. (Note: be aware that your firewall might be blocking traffic to this port.)
- TLS is required.
- The only supported authentication mechanism is SASL PLAIN. For additional information, please see: http://code.google.com/apis/talk/open_communications.html

RELATED TOPICS

- 6.1 Enabling or disabling Notifications for a Plan
- 6.2 Configuring Bamboo to send SMTP Email
- 6.3 Configuring Bamboo to use Instant Messaging (IM)
6.3.1 Configuring Bamboo to use Google Talk for Instant Messaging

Bamboo Documentation Home
7. Managing Data and Backups

- 7.1 Re-indexing Data
- 7.2 Exporting Data for Backup
- 7.3 Importing Data from Backup
- 7.4 Specifying the Expiry Date for Build Results
7.1 Re-indexing Data

You will need to re-index your Bamboo data whenever you perform a data import. It is also good practice to re-index data on a regular basis.

⚠️ Before you begin
While indexing, Bamboo will not be accessible. This may take a few minutes to complete.

To re-index Bamboo's build results data,

1. Click the 'Administration' link in the top navigation bar.
2. Click the 'Indexing' link in the left navigation column.
3. Click the 'Index' button.

RELATED TOPICS

- 7.1 Re-indexing Data
- 7.2 Exporting Data for Backup
- 7.3 Importing Data from Backup
- 7.4 Specifying the Expiry Date for Build Results

Bamboo Documentation Home
7.2 Exporting Data for Backup

Before you begin

Bamboo will be unavailable until the export process completes. Depending on the number of builds and tests, the export may take a long time to complete and may require large amounts of disk space. Please make sure you have enough disk space before proceeding.

To export data for backup,

1. Click the 'Administration' link in the top navigation bar.
2. Click the 'Export' link in the left navigation column.
3. Type the absolute 'File Path' to which Bamboo is to export data. For example, "/opt/bamboo/bamboohome/export.zip".
4. Click the 'Export' button.

Screenshot: Export

Export

Specify Export Paths

File Path: 

/opt/bamboo/bamboohome/export.zip

Specify the absolute path on the server where Bamboo should export. For example, "/opt/bamboo/bamboohome/export.zip".
RELATE TOPICS

- 7.1 Re-indexing Data
- 7.2 Exporting Data for Backup
- 7.3 Importing Data from Backup
- 7.4 Specifying the Expiry Date for Build Results

Bamboo Documentation Home
7.3 Importing Data from Backup

This page last changed on Feb 18, 2007 by rosie@atlassian.com.

⚠️ Before you begin

The import process will DELETE this instance and restore data from a previous export of Bamboo. This includes login data, hence you will need an administration login that is contained in the Bamboo data to be imported. Bamboo will be unavailable until the import process is complete, which may take some time. Please check the paths of your builders and JDK after importing.

You will also need to index the data after the import is complete.

To import data from backup,

1. Click the 'Administration' link in the top navigation bar.
2. Click the 'Import' link in the left navigation column.
3. Type the absolute 'File Path' from which Bamboo is to import data. For example, "/opt/bamboo/bamboohome/export.zip".
4. Select the 'Backup Data' check-box (HIGHLY RECOMMENDED).
5. Specify the absolute 'File path of backup' to which Bamboo should backup data (note that this must be different from the 'File Path' above). For example, "/opt/bamboo/bamboohome/backup.zip".
6. Click the 'Import' button.
Import

Specify Import Paths

You need an administration login in the import file to be able to make changes after import.

The import process will DELETE this instance and restore data from a previous export of Bamboo. This includes login data hence you will need an administration login that is contained in the Bamboo data to be imported. Bamboo will be unavailable until the import process is complete which may take some time. Please check the paths of your builders and JDK after importing.

File Path:

Specify the absolute path to the file on the server from which Bamboo is to import from. For example: `ftp://bamboo/bamboo-home/export.zip`.

Backup data?

Although this will make the import process longer, backup is strongly recommended. Import will not proceed unless Bamboo successfully backups.

File path of backup:

Specify the absolute path on the server where Bamboo should backup to. For example: `ftp://bamboo/bamboo-home/backup.zip`.

Import

RELATED TOPICS

- 7.1 Re-indexing Data
- 7.2 Exporting Data for Backup
- 7.3 Importing Data from Backup
- 7.4 Specifying the Expiry Date for Build Results

Bamboo Documentation Home
7.4 Specifying the Expiry Date for Build Results

The expiry date determines how long your build results data will be retained in your Bamboo system (e.g. for reporting purposes) before being automatically deleted. If no expiry date is specified, build results will never be automatically deleted from Bamboo.

Note that you can also delete plans (and their build results) manually — see 1.6 Disabling or Deleting a Plan and 1.6.1 Deleting a Build Result.

⚠️ If you specify an expiry date, ensure that you back up your build results data before its expiry date is reached.

To specify an expiry date for build results data,

1. Click the 'Administration' link in the top navigation bar.
2. Click the 'Build Expiry' link in the left navigation column.
3. Select the 'Enable Build Expiry' check-box.
4. The 'Remove builds that are more than' fields will appear. Select 'days', 'weeks' or 'months', and type the appropriate number.
5. Click the 'Save' button.

Screenshot: 'Build Expiry'.

**Build Expiry**

**Build Expiry Settings**

You can use this to tell Bamboo when to remove expired data that is no longer useful. Bamboo will check for and remove expired data at midnight.

- Enable build expiry.
- Remove builds that are more than 24 months old.

Save

RELATED TOPICS

- 7.1 Re-indexing Data
- 7.2 Exporting Data for Backup
- 7.3 Importing Data from Backup
- 7.4 Specifying the Expiry Date for Build Results
8. Configuring System Settings

- 8.1 Locating Important Directories and Files
  - 8.1.1 Specifying Bamboo's Working Directory
- 8.2 Viewing Bamboo's System Information
- 8.3 Updating your Bamboo License Details
- 8.4 Specifying Bamboo's URL
- 8.5 Enabling JIRA integration
- 8.6 Enabling GZIP Compression
- 8.7 Enabling Bamboo's Remote API
8.1 Locating Important Directories and Files

This page last changed on May 08, 2007 by edwin@atlassian.com.

When you installed Bamboo, you specified two directories:

- **Bamboo installation directory** — This is the directory where the Bamboo application files are installed. (The default location depends on your operating system: Windows, Unix/Linux, Solaris or Mac OS.)
- **Bamboo home directory** — This is the directory where Bamboo's configuration data and build results are stored. (The default location depends on your operating system: Windows, Unix/Linux, Solaris or Mac OS.) This directory can grow quite large when managing large quantities of plans and builds.

The most important contents of these two directories are described below.

### Bamboo home directory

- **bamboo.cfg.xml** — This is Bamboo's core configuration file. It includes the configuration information for connecting to Bamboo's database.
- **database/** — This directory contains Bamboo's embedded HSQL database. The database contains plan configurations and some build results data.
- **index/** — This directory contains the build results index. Removing or modifying files in this directory may corrupt build history. Rebuilding the search index from Bamboo's global administration screen (see 7.1 Re-indexing Data) will completely regenerate the contents of this directory.
- **xml-data/** — This directory contains all files relating to source repositories and build results.
  - **xml-data/build-dir/** — This is known as the Working Directory. This is where Bamboo temporarily puts the checked-out files it is building. The location of this directory was specified via the Setup Wizard, can be viewed as described in 8.2 Viewing Bamboo's System Information, and can be changed as described in 8.1.1 Specifying Bamboo's Working Directory.
  - **xml-data/builds/** — This is known as the Build Directory. This is where Bamboo stores build results and artifacts (note that they will be deleted as described in 7.4 Specifying the Expiry Date for Build Results). The location of this directory was specified via the Setup Wizard, and can be viewed as described in 8.2 Viewing Bamboo's System Information. Its contents can be backed up as per 7.2 Exporting Data for Backup.
    - **xml-data/builds/PLAN_KEY/results** — Contains the build results for all the builds belonging to the 'PLAN_KEY' plan. Each build result is an individual XML file. Do not edit these files or the corresponding information in the database may become corrupt.
    - **xml-data/builds/PLAN_KEY/download-data/rss** — Contains the content for various RSS feeds related to the 'PLAN_KEY' plan.
  - **xml-data/configuration/** — This is known as the Configuration Directory. It contains server-wide configuration information. The location of this directory was specified via the Setup Wizard, and can be viewed as described in 8.2 Viewing Bamboo's System Information. Its contents can be backed up as per 7.2 Exporting Data for Backup.

### Bamboo installation directory

- **webapp/WEB-INF/classes/bamboo-init.properties** — This file tells Bamboo where to find the Bamboo home directory. The location of this directory is specified by the Bamboo administrator as described in the Bamboo Installation Guide, and can be viewed as described in 8.2 Viewing...
**Bamboo's System Information.**
- `bamboo.sh` * — This is the startup file for Bamboo Standalone under Unix/Linux, Solaris and Mac OS.
- `bamboo.bat` * — This is the startup file for Bamboo Standalone under Windows.
- `bamboo.pid` * — This file, under Linux, contains the Process ID for the running instance of Bamboo.
- `conf/wrapper.conf` * — This file provides the means to configure Bamboo on startup, when using the Java Service wrapper under Linux or Windows.
- `scripts/` — This directory contains operational scripts, including scripts for CVS and SVN triggers.
- `wrapper/` — This directory contains the necessary files to start Bamboo using the Java Service wrapper (see the Mac and Linux installation guides).
- `logs/` — This directory contains logs written by the Java Service wrapper. (Note: The Bamboo server logs are written to the root of the installation directory. Build logs are stored in the xml-data/builds/ sub-directories.)
- `webapp/` — This directory contains all the Bamboo server application files.
- `webapp/WEB-INF/lib/` — This directory is used when deploying Bamboo plugins. It also contains other libraries required by Bamboo.
- `webapp/WEB-INF/classes/log4j.properties` — This is Bamboo's logging configuration file.
- `webapp/WEB-INF/classes/jetty.xml` * — This is the configuration file for Jetty, the application server that is bundled with Bamboo Standalone. The configuration format is simply a mapping from XML to Java. With this format you can call the methods defined in the Jetty javadoc to configure the server.

* This applies to the Bamboo Standalone distribution. The configuration may differ for the Bamboo EAR-WAR distribution.

### RELATED TOPICS

- **8.1 Locating Important Directories and Files**
  - 8.1.1 Specifying Bamboo's Working Directory
- **8.2 Viewing Bamboo's System Information**
- **8.3 Updating your Bamboo License Details**
- **8.4 Specifying Bamboo's URL**
- **8.5 Enabling JIRA integration**
- **8.6 Enabling GZIP Compression**
- **8.7 Enabling Bamboo's Remote API**

**Bamboo Documentation Home**
8.1.1 Specifying Bamboo's Working Directory

This page last changed on Apr 11, 2007 by rosie@atlassian.com.

The working directory is where Bamboo temporarily puts the checked-out files it is building. By default, this directory is located under the xml-data directory in the Bamboo home directory.

To change the location of Bamboo's working directory,

1. Click the 'Administration' link in the top navigation bar.
2. Click the 'General Configuration' link in the left navigation column.
3. This will display the 'General Configuration' page. In the 'Working Directory' field, type the absolute path of your Bamboo server (for example, "C:/bamboo-home/xml-data/build-dir").
4. Click the 'Save' button.

RELATED TOPICS

- 8.1 Locating Important Directories and Files
  - 8.1.1 Specifying Bamboo's Working Directory
- 8.2 Viewing Bamboo's System Information
- 8.3 Updating your Bamboo License Details
- 8.4 Specifying Bamboo's URL
- 8.5 Enabling JIRA integration
- 8.6 Enabling GZIP Compression
- 8.7 Enabling Bamboo's Remote API

Bamboo Documentation Home
8.2 Viewing Bamboo's System Information

This page last changed on Apr 11, 2007 by rosie@atlassian.com.

When you installed Bamboo, you provided information about how the system should be configured.

To view your Bamboo system information,

1. Click the 'Administration' link in the top navigation bar.
2. Click the 'System Info' link in the left navigation column.

Screenshot: 'System Information'
**System Information**

### System Properties

- **System Date:** Wednesday, 11 Apr 2007
- **System Time:** 11:11:43
- **Username:** asridhar
- **User Timezone:** Australia/ACT
- **User Locale:** English (United States)
- **System Encoding:** UTF-8
- **Operating System:** Linux 2.6.17-11-generic
- **Operating System Architecture:** i386
- **Available Processors:** 2
- **Application Server Container:** Jetty/6.1.4

### Java Runtime Information

- **Java Version:** 1.5.0.08
- **Java Vendor:** Sun Microsystems Inc.
- **JVM Version:** 1.0
- **JVM Vendor:** Sun Microsystems Inc.
- **JVM Implementation Version:** 1.5.0.08-b03
- **Java Runtime:** Java(TM) 2 Runtime Environment, Standard Edition
- **Java VM:** Java HotSpot(TM) Client VM

### Java VM Memory Statistics

- **Total Memory:** 32 MB
- **Free Memory:** 13 MB
- **Used Memory:** 19 MB

### Bamboo Paths

- **Configuration Path:** /home/asridhar/Bamboo/data/bamboo-home/xml-data/configuration
- **Build Path:** /home/asridhar/Bamboo/data/bamboo-home/xml-data/builds
- **Bamboo Home:** /home/asridhar/Bamboo/data/bamboo-home

⚠️ For information about the 'Bamboo Paths' please see [8.1 Locating Important Directories and Files](#).
8.1 Locating Important Directories and Files
  8.1.1 Specifying Bamboo's Working Directory
8.2 Viewing Bamboo's System Information
8.3 Updating your Bamboo License Details
8.4 Specifying Bamboo's URL
8.5 Enabling JIRA integration
8.6 Enabling GZIP Compression
8.7 Enabling Bamboo's Remote API

Bamboo Documentation Home
8.3 Updating your Bamboo License Details

When you upgrade or renew your Bamboo license, you will receive a new license key. You will need to update your Bamboo server with the new license key.

To update your Bamboo license key,

1. Click the 'Administration' link in the top navigation bar.
2. Click the 'License Details' link in the left navigation column.
3. This will display your existing Bamboo license details, and an empty box called 'License Key'. Paste your new license into this box.
4. Click the 'Save New License' button.

⚠ Licensing Questions?
Please see the Licensing FAQ.

RELATED TOPICS

- 8.1 Locating Important Directories and Files
  - 8.1.1 Specifying Bamboo's Working Directory
- 8.2 Viewing Bamboo's System Information
- 8.3 Updating your Bamboo License Details
- 8.4 Specifying Bamboo's URL
- 8.5 Enabling JIRA integration
- 8.6 Enabling GZIP Compression
- 8.7 Enabling Bamboo's Remote API

Bamboo Documentation Home
8.4 Specifying Bamboo's URL

This is the base URL of this installation of Bamboo. All links created (for emails etc) will be prefixed by this URL.

To specify Bamboo's URL,

1. Click the 'Administration' link in the top navigation bar.
2. Click the 'General Configuration' link in the left navigation column.
3. This will display the 'General Configuration' page. In the 'Base URL' field, type the URL address of your Bamboo server (for example, "http://keg:8080/bamboo").
4. Click the 'Save' button.

RELATED TOPICS

- 8.1 Locating Important Directories and Files
  - 8.1.1 Specifying Bamboo's Working Directory
- 8.2 Viewing Bamboo's System Information
- 8.3 Updating your Bamboo License Details
- 8.4 Specifying Bamboo's URL
- 8.5 Enabling JIRA integration
- 8.6 Enabling GZIP Compression
- 8.7 Enabling Bamboo's Remote API

Bamboo Documentation Home
8.5 Enabling JIRA integration

When Bamboo's JIRA integration plugin is enabled, Bamboo can provide greater visibility of the issue tracking cycle by automatically linking JIRA issues (in commit messages) to Bamboo builds.

Note that this will require your Bamboo server to login to your JIRA server, regardless of whether your JIRA server is running in 'public' or 'private' mode.

⚠️ Before you begin:
Ensure that the 'Remote API' option is enabled on your JIRA server.

To enable the JIRA integration plugin,

1. Click the 'Administration' link in the top navigation bar.
2. Click the 'JIRA Server' link in the left navigation column.
3. In the 'Host URL' field, type the URL address of your JIRA server (e.g. 'http://jira.atlassian.com').
4. In the 'Username' field, type the name of the JIRA account which your Bamboo server will use to login to your JIRA server.
   - This JIRA account does not require JIRA administration permission.
5. In the 'Password' field, type the corresponding password for the JIRA account you specified in step 4.
6. In the 'Test' section, type a JIRA issue key in the 'Issue Key' field (e.g. 'BAM-738').
7. Click the 'Test' button. This should display the following message: 'Successfully retrieved JIRA issue from remote server'. If not, check that you can login to your JIRA server using the JIRA account and password you specified in steps 4 and 5.
8. When the test is successful, click the 'Save' button.

💡 Two-way integration is available

JIRA’s Bamboo plugin allows JIRA users to view the relevant Bamboo builds for a JIRA issue, from within JIRA.

RELATED TOPICS

- [8.1 Locating Important Directories and Files](#)
  - [8.1.1 Specifying Bamboo's Working Directory](#)
- [8.2 Viewing Bamboo's System Information](#)
- [8.3 Updating your Bamboo License Details](#)
• 8.4 Specifying Bamboo's URL
• 8.5 Enabling JIRA integration
• 8.6 Enabling GZIP Compression
• 8.7 Enabling Bamboo's Remote API

Bamboo Documentation Home
8.6 Enabling GZIP Compression

You can enable GZIP compression in order to reduce the size of Bamboo's web pages.

This is useful if Bamboo is being run over slow networks. There is a slight performance penalty, and note that GZIP may not work for languages other than English.

To enable GZIP Compression,

1. Click the 'Administration' link in the top navigation bar.
2. Click the 'General Configuration' link in the left navigation column.
3. Select the 'Apply gzip compression to reduce the size of Bamboo's web pages?' check-box.
4. Click the 'Save' button.

RELATED TOPICS

- 8.1 Locating Important Directories and Files
  - 8.1.1 Specifying Bamboo's Working Directory
- 8.2 Viewing Bamboo's System Information
- 8.3 Updating your Bamboo License Details
- 8.4 Specifying Bamboo's URL
- 8.5 Enabling JIRA integration
- 8.6 Enabling GZIP Compression
- 8.7 Enabling Bamboo's Remote API

Bamboo Documentation Home
8.7 Enabling Bamboo's Remote API

You can access Bamboo's data from an external program by using Bamboo's REST-style remote API.

The remote API is disabled by default. Bamboo will return an error if people try to use the remote API when it is disabled.

To enable the remote API,

1. Click the 'Administration' link in the top navigation bar.
2. Click the 'General Configuration' link in the left navigation column.
3. Select the 'Accept remote API calls?' check-box.
4. Click the 'Save' button.
5. Bamboo will now accept remote calls. You do not have to restart the Bamboo server.

⚠️ Looking for Remote API documentation? See Bamboo API

RELATED TOPICS

- 8.1 Locating Important Directories and Files
  - 8.1.1 Specifying Bamboo's Working Directory
- 8.2 Viewing Bamboo's System Information
- 8.3 Updating your Bamboo License Details
- 8.4 Specifying Bamboo's URL
- 8.5 Enabling JIRA integration
- 8.6 Enabling GZIP Compression
- 8.7 Enabling Bamboo's Remote API

Bamboo Documentation Home
9. Configuring Plugins

- 9.1 About Bamboo Plugins
- 9.2 Enabling the 'Auto-Favourites' Plugin
- 9.3 Installing a new Plugin
9.1 About Bamboo Plugins

A Bamboo plugin is a program that provides a piece of Bamboo functionality. Bamboo comes with some preinstalled plugins, including:

- 'AutoFavourites' plugin
- 'JIRA' plugin

Additional plugins can be created (see the Bamboo Plugin Developer's Guide) and installed into your Bamboo system.

RELATED TOPICS

- 9.1 About Bamboo Plugins
- 9.2 Enabling the 'Auto-Favourites' Plugin
- 9.3 Installing a new Plugin

Bamboo Documentation Home
9.2 Enabling the 'Auto-Favourites' Plugin

If the 'Auto-Favourites' plugin is enabled, then a plan will be automatically added to a user's 'My Bamboo' tab when the user checks-in code to the plan's source-code repository.

To enable the 'Auto-Favourites' plugin,

1. Click the 'Administration' link in the top navigation bar.
2. Click the link 'Configure Auto-Favourites Plugin' in the left-hand column.
3. Tick the box 'Enable Auto-Favourite Plugin'.
4. Click the 'Save' button.

A plan will now be automatically added to a user's favourites the first time the user checks-in code to the plan's source-code repository. Note that, if the user removes the plan from their favourites, it will not be automatically added again.

RELATED TOPICS

- 9.1 About Bamboo Plugins
- 9.2 Enabling the 'Auto-Favourites' Plugin
- 9.3 Installing a new Plugin

Bamboo Documentation Home
9.3 Installing a new Plugin

To install a new plugin,

1. Copy the new plugin (ie. JAR file) into the appropriate directory as described in 8.1 Locating Important Directories and Files.
2. Restart Bamboo.

RELATED TOPICS

- 9.1 About Bamboo Plugins
- 9.2 Enabling the 'Auto-Favourites' Plugin
- 9.3 Installing a new Plugin

Bamboo Documentation Home
Appendix A. Embedding Bamboo into Other Applications

- Javascript Widgets
  - All Plans & My Favourite Plans
  - Latest Builds
  - Latest Status of a Plan
  - My Latest Changes
  - Plan Summary Graphs
Javascript Widgets

Bamboo has a number of widgets which can be used by external applications:

- All Plans & My Favourite Plans
- Latest Builds
- Latest Status of a Plan
- My Latest Changes
- Plan Summary Graphs
All Plans & My Favourite Plans

This page last changed on Feb 18, 2007 by bmccoy.

These widgets retrieve a summary of plans for a particular Bamboo instance. The summary is shown as a list and depicts the current status, the last completed builds and the reason for the last build. You can either show all the plans or just those that are in your favourites list (username and password required).

To use this widget

1. Include the style sheet in your html document

```
<link rel="stylesheet" type="text/css" href="<bamboo-base-url>/styles/bamboo-widget.css" >
```

2. Place the following script tag in your html

- For all plans

```
<script type="text/javascript"
src="<bamboo-base-url>/js/jsBuildSummaryAll.action" >
</script>
```

- For your favourite plans

```
<script type="text/javascript"
src="<bamboo_base_url>/js/jsBuildSummaryFavourites.action?os_username=<your-user-name>&os_password=<your_password>
</script>
```

3. Replace `<bamboo-base-url>` with the base url for your bamboo instance.

4. Replace `<your-user-name>` and `<your-password>` with the appropriate values.

5. Style! - The style sheet provided just gives some basic style definitions. You can override these definitions to customise the widgets to suit your needs.

Example

Live example from http://opensource.bamboo.atlassian.com/

```
<link rel="stylesheet" type="text/css"
href="http://opensource.bamboo.atlassian.com/styles/bamboo-widget.css">
<script type="text/javascript"
src="http://opensource.bamboo.atlassian.com/js/jsBuildSummaryAll.action" ></script>
```
Latest Builds

This page last changed on Feb 18, 2007 by bmccoy.

This widget produces a list of the last 15 completed builds. A summary is provided for each outlining the build number, reason for the build, date, duration and test results.

To use this widget

1. Include the style sheet in your html document

```html
<link rel="stylesheet" type="text/css" href="<bamboo-base-url>/styles/bamboo-widget.css" />
```

2. Place the following script tag in your html

```html
<script type="text/javascript" src="<bamboo-base-url>/js/showRecentlyCompleted.action"></script>
```

3. Replace `<bamboo-base-url>` with the base url for your bamboo instance.

4. **Style!** - The style sheet provided just gives some basic style definitions. You can override these definitions to customise the widgets to suit your needs.

Example

Live example from [http://opensource.bamboo.atlassian.com/](http://opensource.bamboo.atlassian.com/)

```html
<link rel="stylesheet" type="text/css" href="http://opensource.bamboo.atlassian.com/styles/bamboo-widget.css" />

<script type="text/javascript" src="http://opensource.bamboo.atlassian.com/js/showRecentlyCompleted.action" ></script>
```
Latest Status of a Plan

This page last changed on Feb 21, 2007 by bmccoy.

This widget allows you to view the current status of a particular plan.

To use this widget

1. Include the style sheet in your html document

   <link rel="stylesheet" type="text/css" href="<bamboo-base-url>/styles/bamboo-status.css" >

2. Place the following script tag in your html

   <script type="text/javascript"
   src="<bamboo-base-url>/js/showLatestBuildStatus.action?buildKey=<plan-key>">
   </script>

3. Replace <bamboo-base-url> with the base url for your bamboo instance.

4. Replace <plan-key> with the key of the plan you want to summarise.
   eg. TEST-DEF

5. Style - The style sheet provided just gives some basic style definitions. You can override these definitions to customise the widgets to suit your needs.

Example

Live example from http://opensource.bamboo.atlassian.com/

   <link rel="stylesheet" type="text/css" href="http://opensource.bamboo.atlassian.com/styles/bamboo-status.css">
   <script type="text/javascript" src="http://opensource.bamboo.atlassian.com/js/showLatestBuildStatus.action?buildKey=STRUTS-MAIN">
   </script>
My Latest Changes

This page last changed on Feb 21, 2007 by bmccoy.

This widget allows you to view a list of your 10 most recent changes. It provides details of the changes you made (including the commit comments and links to related JIRA issues) as well as details of the build the change was included in (success or failure, how long ago and test results).

To use this widget

1. Include the style sheet in your html document

   `<link rel="stylesheet" type="text/css" href="<bamboo-base-url>/styles/bamboo-widget.css" />

2. Place the following script tag in your html

   `<script type="text/javascript" src="<bamboo-base-url>/js/myChanges.action?os_username=<your-user-name>&os_password=<your-password">"></script>`

3. Replace `<bamboo-base-url>` with the base url for your bamboo instance.

4. Replace `<your-use-name>` and `<your-password>` with the appropriate values.

5. Style!! - The style sheet provided just gives some basic style definitions. You can override these definitions to customise the widgets to suit your needs.

Example

   `<link rel="stylesheet" type="text/css" href="http://localhost:8085/styles/bamboo-widget.css">
   <script type="text/javascript" src="http://localhost:8085/js/myChanges.action?os_username=admin&os_password=admin" /></script>`
<table>
<thead>
<tr>
<th>Build</th>
<th>When</th>
<th>Comments</th>
<th>Test Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUCBUILD-DEF-88</td>
<td>8 minutes ago</td>
<td>20th time lucky!</td>
<td>2 passed</td>
</tr>
<tr>
<td>SUCBUILD-DEF-83</td>
<td>14 minutes ago</td>
<td>Trying to stop build from failing (TST-11)</td>
<td>4 out of 6 failed!</td>
</tr>
<tr>
<td>SUCBUILD-DEF-82</td>
<td>17 minutes ago</td>
<td>Altered test files (TST-11)</td>
<td>4 out of 6 failed!</td>
</tr>
<tr>
<td>SUCBUILD-DEF-81</td>
<td>32 minutes ago</td>
<td>Added failing test to original test suite</td>
<td>4 out of 6 failed!</td>
</tr>
<tr>
<td>SUCBUILD-DEF-75</td>
<td>1 hour ago</td>
<td>*** empty log message ***</td>
<td>4 out of 6 failed!</td>
</tr>
<tr>
<td>SUCBUILD-DEF-74</td>
<td>3 days ago</td>
<td>Brycie's change is related to TST-1</td>
<td>4 out of 6 failed!</td>
</tr>
<tr>
<td>SUCBUILD-DEF-73</td>
<td>3 days ago</td>
<td>Trying to break build</td>
<td>4 out of 6 failed!</td>
</tr>
<tr>
<td>AMP-DEF-10</td>
<td>1 week ago</td>
<td>*** empty log message ***</td>
<td>3 passed</td>
</tr>
<tr>
<td>TEST-DEF-80</td>
<td>1 week ago</td>
<td>*** empty log message ***</td>
<td>2 out of 4 failed!</td>
</tr>
<tr>
<td>SUCBUILD-DEF-2-14</td>
<td>1 week ago</td>
<td>*** empty log message ***</td>
<td>2 passed</td>
</tr>
</tbody>
</table>
Plan Summary Graphs

This page last changed on Feb 18, 2007 by bmccoy.

These widgets allows you to view either of the two summary graphs displayed on the plan summary page. The two graphs are:

- Build Duration & Number of Failures per Build
- Successful Builds & Average Duration Per Time Period

To use this widget

1. Place the following script tag in your html

   - For Build Duration & Number of Failures per Build
     
     `<script type="text/javascript" src="<bamboo-base-url>/js/viewCombinedByBuildNumberChart.action?buildKey=<Plan-Key>&filterController.selectedFilterKey=<filter-key>"></script>`

   - For Successful Builds & Average Duration Per Time Period
     
     `<script type="text/javascript" src="<bamboo_base_url>/js/jsViewCombinedByTimePeriodChart.action?buildKey=<Plan-Key>&filterController.selectedFilterKey=<filter-key>"></script>`

2. Replace <plan-key> with the key of the plan you want to summarise.
   eg. TEST-DEF

3. Replace <filter-key> with one of the following options:
   * LAST_25_BUILDS
   * LAST_7_DAYS
   * LAST_30_DAYS
   * LAST_90_DAYS
   * ALL_BUILDS

Example

Live example from http://opensource.bamboo.atlassian.com/

```
<script type="text/javascript" src="http://opensource.bamboo.atlassian.com/js/viewCombinedByBuildNumberChart.action?buildKey=STRUTS-MAIN&filterController.selectedFilterKey=LAST_25_BUILDS"></script>
<script type="text/javascript" src="http://opensource.bamboo.atlassian.com/js/jsViewCombinedByTimePeriodChart.action?buildKey=STRUTS-MAIN&filterController.selectedFilterKey=LAST_25_BUILDS"></script>
```
Bamboo Installation Guide

This page last changed on Apr 10, 2007 by rosie@atlassian.com.

Requirements

Bamboo requires JDK/JRE1.4+ installed. However, we recommend that you use Sun JDK 1.5 (Java 5) and above for best performance.

In addition, if you are running Bamboo as a WAR, Bamboo requires a servlet container that supports Servlet 2.4 spec. Most modern containers should comply to this.

⚠️ Warning
Bamboo has incompatibilities when not running under Sun JDK. Please make sure you are using Sun JDK to run your Bamboo installation.

Choose your Bamboo 'Distribution':

Bamboo is available in two 'distributions':

<table>
<thead>
<tr>
<th>Standalone distribution</th>
<th>EAR-WAR distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Pre-packaged with the Jetty application server</td>
<td>• Deploys into an existing application server</td>
</tr>
<tr>
<td>• Requires virtually no setup</td>
<td>• Requires manual configuration</td>
</tr>
<tr>
<td>• Recommended for all users</td>
<td>• Suitable only for system administrators</td>
</tr>
</tbody>
</table>

The Standalone distribution is recommended even for organisations with an existing application server environment.

NEXT

- [Standalone Installation Guide — Windows](#)
- [Standalone Installation Guide — Linux](#)
- [Standalone Installation Guide — Mac](#)

or

- [EAR-WAR Installation Guide](#)

RELATED TOPICS

Document generated by Confluence on May 08, 2007 23:50
Running the Setup Wizard

Upgrade Guide
Release Notes

Bamboo Documentation Home
Bamboo EAR-WAR Installation Guide

This page last changed on Feb 25, 2007 by mark@atlassian.com.

The EAR/WAR edition of Bamboo is intended for deployment into an existing J2EE application server. It is assumed that you already know how to deploy a webapp on the application server of choice. If not, it is recommended to install the standalone distribution.

The following instructions are only indicative of the process and examples are based on installing the Bamboo WAR file on Apache Tomcat. Deployment and Configuration will differ based on your webserver.

⚠️ Before you begin

Please review the System Requirements.

Step 1. Download the WAR file

1. The Bamboo WAR file is available for download here.

Step 2. Deploy onto your Application Server

In Tomcat you can do one of two things

1. Place the WAR file directly into the webapps folder of Tomcat. When Tomcat starts it will perform all the necessary extractions.
2. Extract the WAR file to your chosen directory in the webapps folder
   - Windows users avoid Win XP's built in unzip as it doesn't extract all the files. Use a 3rd party zip extractor like WinZip.
   - Solaris users will need to use GNU tar to handle the long filenames.

By default the WAR file will extract to a folder called Bamboo-<version>. Note: The name of the directory in the webapps folder will form the url required to access Bamboo (eg. Tomcat/webapps/bamboo-1.0/ will become http://host:port/bamboo-1.0/). You may wish to change the directory name for a more concise access url.

Step 3. Setting Bamboo Properties

You will need to set your bamboo home directory. You can do this in one of three ways:

1. set the bamboo.home property in the file /WEB-INF/classes/bamboo-init.properties to your chosen bamboo home directory
2. pass the bamboo home directory to the application server as a java opt. (eg. -Dbamboo.home=C:/bamboo/bamboo-home)
3. set a system property bamboo.home to your bamboo home directory
Bamboo also recommends setting the following java opts on your Application Server.

- server - Ensures that the jvm starts up in server mode. (This will perform various optimisation tasks, this is beneficial for long running applications.)
- -Xmx512m - Sets the maximum memory recommended for bamboo.
- -XX:MaxPermSize=256m - Sets the maximum permgen memory recommended for Bamboo.
- -Djava.awt.headless=true - For Unix systems. This allows AWT to run in headless mode and is required if running Bamboo in non-graphical environments. For more details visit the Sun Developer Network.

In Tomcat you can set the above java opts as follows

- Windows
  1. Find the setenv.bat file.
  2. Assign the desired properties to the JAVA_OPTS variable:

```bash
set JAVA_OPTS=-server -XX:MaxPermSize=256m -Dbamboo.home=/opt/bamboo/bamboohome -Xmx512m -Djava.awt.headless=true $JAVA_OPTS
```

- Linux Bases System
  1. Find the setenv.sh file
  2. Assign the desired properties to the JAVA_OPTS variable:

```bash
JAVA_OPTS="-server -XX:MaxPermSize=256m -Dbamboo.home=/opt/bamboo/bamboohome -Xmx512m -Djava.awt.headless=true $JAVA_OPTS"
export JAVA_OPTS
```

Step 3. Restart Server

1. Shut down, and then restart your application server
2. Bamboo should now be accessible on http://host:port/bamboo

Step 4. Configure Bamboo

See Running the Setup Wizard.

 RELATED TOPICS

Bamboo Installation Guide
Bamboo Documentation Home
Bamboo Standalone Installation Guide (Linux)

To install Bamboo Standalone on Linux,

**Step 1. Downloading & Installing Bamboo Standalone**

Bamboo Standalone is available for download [here](#).

**Linux Archive (.tar.gz)**

To install Bamboo using the Linux archive version (atlassian-bamboo-x.x-standalone.tar.gz), you need to extract the files to a Bamboo installation directory of your choice. By default, the root directory of the tar file is "Bamboo".

You will also need to setup your Bamboo home directory — this is the directory where Bamboo will store its configuration data. To do this, open the file named `bamboo-init.properties` in the `<Bamboo installation directory>/webapp/WEB-INF/classes` directory. In this file, insert the property "bamboo.home", with an absolute path to your Bamboo home directory. Your file should look something like this:

```
bamboo.home=/test/bamboo-home
```

⚠️ **You must use forward-slashes in your directory path. Backslashes are not recognised by Bamboo.**

**Step 2. Launching Bamboo Standalone on Linux**

There are two ways you can launch Bamboo on Linux:

1. **Launch via `bamboo.sh` startup script**

You can start Bamboo with the default `bamboo.sh` file in your installation root directory. The `bamboo.sh` command accepts the following options (e.g. `./bamboo.sh start`):

   - `start` — this starts Bamboo.
   - `stop` — this stops Bamboo.
   - `restart` — this restarts Bamboo
   - `status` — this provides the current status of Bamboo.

2. **Launch via Java Service Wrapper**
Alternatively, you can start Bamboo via a Java Service Wrapper, which provides services such as automatic restarting. To do this, you will need to use the `start-bamboo` command available in the `/wrapper` folder of the Bamboo installation. You will need to fire the command with one of the following options (e.g. `/start-bamboo start`):

- **console** — this starts Bamboo in a console. The logs will scroll to standard out.
- **start** — this starts Bamboo.
- **stop** — this stops Bamboo.
- **restart** — this restarts Bamboo
- **status** — this provides the current status of Bamboo.
- **dump** — stops Bamboo abruptly by killing the process

Once Bamboo has started, you can access it by going to your web browser and entering the address: [http://localhost:8085/](http://localhost:8085/).

### Step 3. Configuring Bamboo

See [Running the Setup Wizard](#).
Bamboo Standalone Installation Guide (Mac)

This page last changed on Apr 11, 2007 by rosie@atlassian.com.

To install Bamboo Standalone on Mac OS,

Step 1. Downloading & Installing Bamboo Standalone

Bamboo Standalone is available for download here. You can choose an Installer (.dmg) or an Archive (.tgz).

Mac OS Installer (.dmg)

Launching the Bamboo Mac OS installer (atlassian-bamboo-x.x-standalone.dmg) will mount the Atlassian Bamboo installation volume. Launch the Bamboo Continuous Integration Server Installer.app to begin the installation wizard.

The installer requires you to specify two directories:

- Bamboo installation directory — This is the directory where Bamboo's application files will be installed. The default is:

  
  
  Applications/Bamboo

- Bamboo home directory — This is the directory where Bamboo will store its configuration data. If the directory you specify doesn't exist, Bamboo will create the directory when it launches. The default is:

  
  
  /Users/<current-user>/Bamboo-home

⚠️ You must use forward-slashes in your directory path. Backslashes are not recognised by Bamboo.

Mac OS Archive (.tgz)

To install Bamboo using the Mac OS archive version (atlassian-bamboo-x.x-standalone.tgz), you need to extract the files to a Bamboo installation directory of your choice. By default, the root directory of your tgz file is "Bamboo".

You will also need to setup your Bamboo home directory — this is the directory where Bamboo will store its root configuration data. To do this, open the file named bamboo-init.properties in the <Bamboo installation directory>/webapp/WEB-INF/classes directory. In this file, insert the property "bamboo.home", with an absolute path to your Bamboo home directory. Your file should look something like this:

  
  
  bamboo.home=/test/bamboo-home
Step 2. Launching Bamboo on Mac OS

There are two ways you can launch Bamboo on Mac OS:

1. Launch via `bamboo.sh` startup script

You can start Bamboo with the default `bamboo.sh` file in your installation root directory. The `bamboo.sh` command accepts the following options (e.g. `./bamboo.sh start`):

- `console` — this starts Bamboo in a console. The logs will scroll to standard out.
- `start` — this starts Bamboo.
- `stop` — this stops Bamboo.
- `status` — this provides the current status of Bamboo.

2. Launch via Java Service Wrapper

Alternatively, you can start Bamboo via a Java Service Wrapper, which provides services such as automatic restarting. To do this, you will need to use the `run-bamboo` command available in the `/wrapper` folder of the Bamboo installation. You will need to fire the command with one of the following options (e.g. `./run-bamboo start`):

- `console` — this starts Bamboo in a console. The logs will scroll to standard out.
- `start` — this starts Bamboo.
- `stop` — this stops Bamboo.
- `status` — this provides the current status of Bamboo.

Once Bamboo has started, you can access it by going to your web browser and entering the address: [http://localhost:8085/](http://localhost:8085/).

Step 3. Configuring Bamboo

See [Running the Setup Wizard](#).
Bamboo Standalone Installation Guide (Windows)

This page last changed on Apr 11, 2007 by rosie@atlassian.com.

To install Bamboo Standalone on Windows,

**Step 1. Downloading & Installing Bamboo**

Bamboo Standalone is available for download [here](#). You can choose the Windows Installer (.exe) or the Windows Archive (.zip).

### Windows Installer (.exe)

Launch the Bamboo Windows installer (atlassian-bamboo-x.x-standalone.exe) to begin the installation wizard.

The installer requires you to specify two directories:

- **Bamboo installation directory** — This is the directory where Bamboo's application files will be installed. The default is: 

  ```
  C:/Program Files/Bamboo
  ```

- **Bamboo home directory** — This is the directory where Bamboo will store its configuration data. If the directory you specify doesn't exist, Bamboo will create the directory when it launches. The default is:

  ```
  C:/Documents and Settings/<current-user>/Bamboo-home
  ```

⚠️ You must use forward-slashes in your directory path. Backslashes are not recognised by Bamboo.

### Windows Archive (.zip)

To install Bamboo using the Windows archive version (atlassian-bamboo-x.x-standalone.zip), you need to extract the files to a Bamboo installation directory of your choice. By default, the root directory in your zip file is named "Bamboo".

You will also need to setup your Bamboo home directory — this is the directory where Bamboo will store its root configuration data. To do this, edit the file named `bamboo-init.properties` in the `Bamboo/webapp/WEB-INF/classes` directory. In this file, insert the property "bamboo.home", with an absolute path to your Bamboo home directory. Your file should look something like this:

```properties
bamboo.home=C:/test/bamboo-home
```
Step 2. Launching Bamboo

Once Bamboo is installed on your machine, you can launch the application either via the Start Menu (if you have used the self installer), or by running the batch files available in the root of the Bamboo installation directory. You can run Bamboo in two modes: either in the console, or as a Windows service. Bamboo comes with the following batch files:

- BambooConsole.bat — this starts Bamboo in a Windows console.
- InstallAsService.bat — this installs Bamboo as a Windows service. Note that this will not start Bamboo.
- StartBamboo.bat — this starts your installed Bamboo Windows service.
- StopBamboo.bat — this stops your installed Bamboo Windows service
- UninstallService.bat — this un-installs the Bamboo Windows service from your machine. Note that your Bamboo installation still remains.

Once Bamboo has started, you can access it by going to your web browser and entering the default address: http://localhost:8085/.

Step 3. Configuring Bamboo

See Running the Setup Wizard.
Running the Setup Wizard

This page last changed on Apr 11, 2007 by rosie@atlassian.com.

Step 1. Installation Settings

When you launch Bamboo for the first time, you will need to provide some configuration information before you can start using it.

Welcome to Atlassian Bamboo!

Welcome to Bamboo Continuous Integration Server. Bamboo needs some information before it is fully installed.

Install Bamboo with default settings and an embedded database.

Standard Installation Settings

<table>
<thead>
<tr>
<th>Configuration Directory:</th>
<th>C:\Documents and Settings\Bamboo-home\xml-data\configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Build Data Directory:</td>
<td>C:\Documents and Settings\Bamboo-home\xml-data\builds</td>
</tr>
<tr>
<td>Build Working Directory:</td>
<td>C:\Documents and Settings\Bamboo-home\xml-data\build-dir</td>
</tr>
<tr>
<td>Server ID:</td>
<td>AX11-BE22-CC33-DD44</td>
</tr>
<tr>
<td>License Key:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Please enter your Bamboo license key above - either commercial or evaluation. Please contact Atlassian if you require a license key.

- 'Configuration Directory' — This is where Bamboo will store its configuration files.
- 'Build Data Directory' — This is where Bamboo will store its project data files.
- 'Build Working Directory' — This is where Bamboo will check out project files from source control.
  
  You may want to keep the default settings for the above three directories. Bamboo will default these directories to bamboo-home/xml-data/projects and bamboo-home/xml-data/configuration respectively. For more information please see 8.1 Locating Important Directories and Files.
- 'Server ID' — This is generated automatically by Bamboo.
• 'License Key' — You are required to enter a valid license key before you can use Bamboo. You will be emailed a 30-day free evaluation license when downloading an evaluation version of Bamboo. You can copy and paste the license key from the email into the text box.

Step 2. Bamboo Administrator

Here, you will enter the details of the first registered user to the Bamboo system. This user will have administrative privileges over the entire installation of Bamboo and should not be removed.

Setup Administrator User

Please enter the details of the administrator user for this installation of Bamboo.

<table>
<thead>
<tr>
<th>Administrator User Details</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Username:</strong></td>
</tr>
<tr>
<td>The name to use to login to Bamboo.</td>
</tr>
<tr>
<td><strong>Password:</strong></td>
</tr>
<tr>
<td><strong>Confirm Password:</strong></td>
</tr>
<tr>
<td><strong>Full Name:</strong></td>
</tr>
<tr>
<td><strong>Email:</strong></td>
</tr>
</tbody>
</table>

Step 3. Server Configuration

The final page of the setup wizard allows you to enter some final configuration data for Bamboo.
Bamboo Configuration

Please specify the following settings to complete the installation of your Bamboo Server.

<table>
<thead>
<tr>
<th>Base URL</th>
<th><a href="http://localhost:8085">http://localhost:8085</a></th>
</tr>
</thead>
<tbody>
<tr>
<td>This is the base URL of this installation of Bamboo. All links created (for emails etc) will be prefixed by this URL. For example &quot;<a href="http://localhost:8085">http://localhost:8085</a>&quot;</td>
<td></td>
</tr>
</tbody>
</table>

What configuration options would you like?

- [ ] Apply gzip compression to reduce the size of Bamboo web pages?
  This is useful if Bamboo is being run over slow networks. There is a slight performance penalty, and may not work for non-English languages.

- [ ] Accept remote API calls?
  Do you want to allow remote client access (via REST) for authenticated users?

Complete Installation

Once you have clicked "Complete Installation", the setup process is done and you are now at the Bamboo dashboard.

Next...

1.1 About Projects, Plans and Builds
1.2 Creating a Plan
Bamboo Knowledge Base

This page last changed on Apr 03, 2007 by rosie@atlassian.com.

<table>
<thead>
<tr>
<th>Bamboo Knowledge Base</th>
</tr>
</thead>
<tbody>
<tr>
<td>Answers to commonly raised questions about configuring and using Bamboo:</td>
</tr>
<tr>
<td><strong>Installation FAQ</strong></td>
</tr>
<tr>
<td><strong>Integration FAQ</strong></td>
</tr>
<tr>
<td><strong>Usage FAQ</strong></td>
</tr>
<tr>
<td>- Can multiple plans share a common 3rd-party directory? — For example, you might have three repository directories, say, A, B, and C, where A is specific to one project, and B and C are common across many projects.</td>
</tr>
</tbody>
</table>

Do you have a question, or need help with Bamboo? Please [create a support request](#).
Installation FAQ

This page last changed on Apr 03, 2007 by rosie@atlassian.com.
Integration FAQ

This page last changed on Apr 03, 2007 by rosie@atlassian.com.
Usage FAQ

This page last changed on Apr 03, 2007 by rosie@atlassian.com.
Can multiple plans share a common 3rd-party directory?

For example, you might have three repository directories, say, A, B, and C, where A is specific to one project, and B and C are common across many projects. At this stage, Bamboo doesn't support having multiple checkout directories per build plan. However, you can work around this by setting these three directories up as separate Bamboo build plans, with B and C both being dependant on A (see 3.4 Triggering a Build when another Build finishes). This ensures that B and C will both build if you check-in source changes against A.

To make this work, you will also need to specify as an argument to your build scripts for B and C the location of A, which will be something like this:

```
../A/
```

Using a set up like this, your library module (A) should only be checked out once across the Bamboo instance.
Bamboo Release Notes

This page last changed on May 07, 2007 by edwin@atlassian.com.

Latest Version

The latest release of Bamboo is 1.1

Read the full Bamboo 1.1 Release Notes and Upgrade Guide.

Previous Releases

- Bamboo 1.1 Release Notes
  - Bamboo 1.1 Upgrade Guide
- Bamboo 1.0.5 Release Notes
  - Bamboo 1.0.5 Upgrade Guide
- Bamboo 1.0.4 Release Notes
  - Bamboo 1.0.4 Upgrade Guide
- Bamboo 1.0.3 Release Notes
  - Bamboo 1.0.3 Upgrade Guide
- Bamboo 1.0.2 Release Notes
  - Bamboo 1.0.2 Upgrade Guide
- Bamboo 1.0.1 Release Notes
  - Bamboo 1.0.1 Upgrade Guide
- Bamboo 1.0 Release Notes
  - Bamboo 1.0 Upgrade Guide
- Bamboo 1.0-RC2 Release Notes
  - Bamboo 1.0-RC2 Upgrade Guide
- Bamboo 1.0-RC1 Release Notes
  - Bamboo 1.0-RC1 Upgrade Guide
- Bamboo 1.0-Beta Release Notes
  - Bamboo 1.0-Beta Upgrade Guide
- Bamboo 0.9 Release Notes
  - Bamboo 0.9 Upgrade Guide
- Bamboo 0.8 Release Notes
  - Bamboo 0.8 Upgrade Guide
- Bamboo 0.7 Release Notes
  - Bamboo 0.7 Upgrade Guide
- Bamboo 0.6 Release Notes
  - Bamboo 0.6 Upgrade Guide
- Bamboo 0.5.1 Release Notes
  - Bamboo 0.5.1 Upgrade Guide
- Bamboo 0.5 Release Notes
  - Bamboo 0.5 Upgrade Guide
- Bamboo 0.4 Release Notes
  - Bamboo 0.4 Upgrade Guide
- Bamboo 0.3 Release Notes
  - Bamboo 0.3 Upgrade Guide
- **Bamboo 0.2 Release Notes**
  - **Bamboo 0.2 Upgrade Guide**
- **Bamboo 0.1 Release Notes**
Bamboo 0.1 Release Notes

This page last changed on Mar 06, 2007 by rosie@atlassian.com.

⚠️ Bamboo 1.1 has now been released – see the Bamboo 1.1 Release Notes

The pre-release of Atlassian's Bamboo Continuous Integration Server is here! There're plenty of features here already even if the feature set is far from complete. We'd love to hear any feedback you have on it. Feel free to raise bug reports or feature requests. Please keep in mind that it's still an early release and could be a little rough around the edges!

Simple User Interface

- Easy installation - be up and running in 5 minutes!
- Auto detection of your settings - if you have Maven, Ant or Java setup on your server, Bamboo will autodetect it
- Continuous logs - Monitor colour coded logs of your build through the activity monitor
- Convenient summary list of all projects and color coded results of their last builds.
- Build failures are highlighted and accessible - Test results are parsed and stored inside Bamboo
- Shows you the source code changes that triggered the build.

Managing Projects

- Support for multiple projects
- Rebuilds a project whenever the source code changes
- Ability to manually initiate a build
- Support for multiple JDKs

Building Projects

- Parallel Build Queues - Build multiple projects simultaneously, see progress on each build
- Reporting - Historical graphs for build time an test failures.
- Can build using Shell scripts, Ant and Maven 1 and Maven 2 projects (support for multiple versions).
- Ability to specify custom JVM options like -Xms and -Xmx

Version Control Systems Integration

- Can monitor CVS, Subversion and Perforce repositories
- Detects whenever the source code changes.
- Checks out and syncs the project source code.
- Generates a log of revisions.

Rapid Build Result Feedback

- Sends out emails informing interested parties of build results
• Sends out IM messages containing the build results
• Can access build results from an RSS feed
• Get notified about all builds or only failed builds
• Results are easily embedable within other application

Stay tuned, this is an early pre-release version of Bamboo and we'll adding many features in the weeks to come!
Bamboo 0.2 Release Notes

This page last changed on Mar 06, 2007 by rosie@atlassian.com.

⚠️ Bamboo 1.1 has now been released – see the Bamboo 1.1 Release Notes

⚠️ Upgrading? Please see the Bamboo 0.2 Upgrade Guide.

The 0.2 release saw major changes underneath the hood of Bamboo. See our Bamboo 0.1 Release Notes for more details. Some of the new additions include:

### New Features for 0.2

- New edit project configuration interface - easily accessible menus.
- Improved security, user group based permission controls for administrative functions.
- New email configuration, including authenticated SMTP server connections
- User signup - new users can now sign up the system. System admins can add users to the bamboo-admin group for admin access

### Improvements

- Image based charts
Bamboo 0.2 Upgrade Guide

Upgrading from Bamboo 0.1 to 0.2

Please follow the Bamboo Upgrade Guide.
Bamboo 0.3 Release Notes

This page last changed on Mar 06, 2007 by rosie@atlassian.com.

⚠️ Bamboo 1.1 has now been released – see the Bamboo 1.1 Release Notes.

⚠️ Upgrading? Please see the Bamboo 0.3 Upgrade Guide.

New Features for 0.3

- Improved build results screens with the capability to drill down into test classes and individual test cases.
- Find out which build originally broke failing tests.
- Find out which tests were fixed by a build.
- New instant messaging configuration which can be used across builds.
- JDK 1.4 support (BAM-113)
- Trigger one build to fire when another builds successfully with build dependencies (BAM-93)

<table>
<thead>
<tr>
<th>Key</th>
<th>Summary</th>
<th>Status</th>
<th>Res</th>
</tr>
</thead>
<tbody>
<tr>
<td>BAM-171</td>
<td>View Screens for Mail Server and IM Server</td>
<td>Resolved</td>
<td>FIXED</td>
</tr>
<tr>
<td>BAM-136</td>
<td>Can't add builder when there are no builders</td>
<td>Resolved</td>
<td>FIXED</td>
</tr>
<tr>
<td>BAM-97</td>
<td>IM Server details should be shareable</td>
<td>Resolved</td>
<td>FIXED</td>
</tr>
<tr>
<td>BAM-19</td>
<td>Artifacts should respect MIME types</td>
<td>Resolved</td>
<td>FIXED</td>
</tr>
<tr>
<td>BAM-22</td>
<td>Bamboo creating phantom directories</td>
<td>Resolved</td>
<td>FIXED</td>
</tr>
<tr>
<td>BAM-150</td>
<td>Installation fails to locate bambo-init.properties file for bambo home directory</td>
<td>Resolved</td>
<td>ANSWERED</td>
</tr>
<tr>
<td>BAM-113</td>
<td>UnsupportedClassVersionError deploying Bamboo on BEA WLS 8.1 SP 4</td>
<td>Resolved</td>
<td>FIXED</td>
</tr>
<tr>
<td>BAM-166</td>
<td>Add new screens and improve existing screen to improve overall information delivery</td>
<td>Resolved</td>
<td>FIXED</td>
</tr>
<tr>
<td>Ticket</td>
<td>Description</td>
<td>Status</td>
<td>Resolution</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
<td>---------</td>
<td>------------</td>
</tr>
<tr>
<td>BAM-96</td>
<td>New build failures count in Jabber message is different to web site</td>
<td>Resolved</td>
<td>FIXED</td>
</tr>
<tr>
<td>BAM-93</td>
<td>Project dependencies don't work</td>
<td>Resolved</td>
<td>FIXED</td>
</tr>
<tr>
<td>BAM-130</td>
<td>Validation for groups and users nonfunctional</td>
<td>Resolved</td>
<td>FIXED</td>
</tr>
<tr>
<td>BAM-131</td>
<td>Clicking on a project name for queued project cause 500 error</td>
<td>Resolved</td>
<td>FIXED</td>
</tr>
<tr>
<td>BAM-137</td>
<td>Context Path duplicated when submitting add/edit row in builders</td>
<td>Resolved</td>
<td>FIXED</td>
</tr>
<tr>
<td>BAM-134</td>
<td>attempts to email even when email not selected</td>
<td>Resolved</td>
<td>FIXED</td>
</tr>
<tr>
<td>BAM-178</td>
<td>Fix up footer to include the date that the build was made as well as the build number</td>
<td>Resolved</td>
<td>FIXED</td>
</tr>
<tr>
<td>BAM-180</td>
<td>Delete a Project screen always checks the first project when clicking.</td>
<td>Resolved</td>
<td>FIXED</td>
</tr>
<tr>
<td>BAM-89</td>
<td>Add decent 404 and 500 pages</td>
<td>Resolved</td>
<td>FIXED</td>
</tr>
<tr>
<td>BAM-140</td>
<td>double slash (&quot;&quot;) in link</td>
<td>Resolved</td>
<td>FIXED</td>
</tr>
<tr>
<td>BAM-70</td>
<td>Phantom project folders created when creating a new project/build</td>
<td>Resolved</td>
<td>FIXED</td>
</tr>
</tbody>
</table>
Bamboo 0.3 Upgrade Guide

This page last changed on Jan 17, 2007 by rosie@atlassian.com.

Upgrading from Bamboo 0.2 to 0.3

Please follow the Bamboo Upgrade Guide.

Upgrading from Bamboo 0.1

In addition to the above, please read the Upgrade Guide for every version you are skipping during the upgrade. The complete list of Upgrade Guides is available here.
Bamboo 0.4 Release Notes

This page last changed on Mar 06, 2007 by rosie@atlassian.com.

⚠ Bamboo 1.1 has now been released – see the Bamboo 1.1 Release Notes

⚠ Upgrading? Please see the Bamboo 0.4 Upgrade Guide.

New in Bamboo 0.4

- New Test Case page, which aggregates data for one test case across all builds
- New test summary provides information on the "Top 10 failing tests"
- Inspect each build as they are building; find out it's estimated time remaining, percentage completion, and who triggered the build.
- Indexable test results allow analysis of test and team performance across builds.

<table>
<thead>
<tr>
<th>T</th>
<th>Key</th>
<th>Summary</th>
<th>Assignee</th>
<th>Reporter</th>
<th>Pr</th>
<th>Status</th>
<th>Res</th>
<th>Created</th>
<th>Updated</th>
<th>Due</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>BAM-221</td>
<td>View CVS URL is optional - mark as such</td>
<td>Ben Kuo</td>
<td>Riaz Khanmohamed</td>
<td></td>
<td>FIXED</td>
<td>Resolved</td>
<td>Sep 14, 2006</td>
<td>Sep 15, 2006</td>
<td></td>
</tr>
<tr>
<td></td>
<td>BAM-220</td>
<td>Optional field appears to be mandatory on update (cvs password)</td>
<td>Ben Kuo</td>
<td>Riaz Khanmohamed</td>
<td></td>
<td>FIXED</td>
<td>Resolved</td>
<td>Sep 14, 2006</td>
<td>Oct 05, 2006</td>
<td></td>
</tr>
<tr>
<td></td>
<td>BAM-213</td>
<td>Grammar on Log page</td>
<td>Ben Kuo</td>
<td>Riaz Khanmohamed</td>
<td></td>
<td>FIXED</td>
<td>Resolved</td>
<td>Sep 14, 2006</td>
<td>Sep 14, 2006</td>
<td></td>
</tr>
<tr>
<td></td>
<td>BAM-211</td>
<td>Keep refresh option for logs</td>
<td>Ben Kuo</td>
<td>Riaz Khanmohamed</td>
<td></td>
<td>FIXED</td>
<td>Resolved</td>
<td>Sep 14, 2006</td>
<td>Sep 15, 2006</td>
<td></td>
</tr>
<tr>
<td></td>
<td>BAM-210</td>
<td>Log page refreshing</td>
<td>Ben Kuo</td>
<td>Riaz Khanmohamed</td>
<td></td>
<td>FIXED</td>
<td>Resolved</td>
<td>Sep 14, 2006</td>
<td>Sep 15, 2006</td>
<td></td>
</tr>
<tr>
<td></td>
<td>BAM-207</td>
<td>WebUrl link should link to the diff of the file</td>
<td>Mark Chaimungkorn</td>
<td>Mark Chaimungkorn</td>
<td></td>
<td>FIXED</td>
<td>Resolved</td>
<td>Sep 14, 2006</td>
<td>Sep 14, 2006</td>
<td></td>
</tr>
<tr>
<td></td>
<td>BAM-196</td>
<td>Adding</td>
<td>Mark MattyJ</td>
<td></td>
<td></td>
<td>FIXED</td>
<td></td>
<td>Sep 06,</td>
<td>Sep 06,</td>
<td></td>
</tr>
<tr>
<td>JIRA Key</td>
<td>Description</td>
<td>Resolution</td>
<td>Status</td>
<td>Priority</td>
<td>Reporter</td>
<td>Assignee</td>
<td>Priority</td>
<td>Priority</td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
<td>------------</td>
<td>--------</td>
<td>----------</td>
<td>-----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-194</td>
<td>Implement Lucene for indexing of data</td>
<td>FIXED</td>
<td>Resolved</td>
<td></td>
<td>Edwin Wong</td>
<td>Edwin Wong</td>
<td>Sep 05, 2006</td>
<td>Sep 19, 2006</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-74</td>
<td>Cloning projects &amp; the changeDetection field</td>
<td>FIXED</td>
<td>Resolved</td>
<td></td>
<td>Ben Kuo Ben Kuo</td>
<td>Ben Kuo Ben Kuo</td>
<td>Jul 18, 2006</td>
<td>Jan 21, 2007</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Bamboo 0.4 Upgrade Guide

This page last changed on Jan 17, 2007 by rosie@atlassian.com.

Upgrading from Bamboo 0.3 to 0.4

Please follow the Bamboo Upgrade Guide.

If you have existing build results data in Bamboo, you will need to reindex this before some pages display correctly in Bamboo.

Upgrading from Bamboo 0.2 and earlier.

In addition to the above, please read the Upgrade Guide for every version you are skipping during the upgrade. The complete list of Upgrade Guides is available here.
Bamboo 0.5 Release Notes

This page last changed on Mar 06, 2007 by rosie@atlassian.com.

⚠️ Bamboo 1.1 has now been released – see the Bamboo 1.1 Release Notes

⚠️ Upgrading? Please see the Bamboo 0.5 Upgrade Guide.

New in Release 0.5

Enhanced UI Navigation

In this release, Bamboo has seen a major facelift to its UI including several improvements to the way you can navigate through the system.

The new bread-crumbs system lets you keep track of where you are in Bamboo and jump direct to parent pages.

Navigate between builds with the next and previous links, or jump direct to a result if you have the build number. You can also see how the build has performed in the recent string of runs at a glance.

Relevant information

Bamboo release 0.5 also sees a major revamp to information presentation, focusing on relevant information for you. This occurs at two levels.

At a single build result level, Bamboo highlights important information about a particular build run: Was it successful? Which tests failed? What was the compile error?

Bamboo also intelligently analyses build results: Which failing tests are new? Which ones were already failing in the previous build? In which build did these failures originally occur? Which tests did this build fix? Which build fixed the failure? How long has the build been failing for?

⚠️ BAM-1000 (Updated by bkuo) has failed with 3 new failing tests. It has been failing since BAM-999 (Updated by bkuo, 9 minutes before).

This build is fixed in BAM-1001.

On a more aggregate level, Bamboo collates information across build results to provide insightful statistics on how your project has performed. Charts highlight trends in important metrics such as test failures and build times in a visual manner.
For example, this chart demonstrates the improvement of the build from March, as more builds were run with increasing success. The handy build results filter also lets you choose the set of results for which you wish to view summary information and charts on.

Bamboo also highlights the periods for which the builds have failed, when they were fixed, and how long it took for the failure to be rectified.

**Recent Failures**

- Average time to fix a failure: **9 hours, 54 minutes**
- Average number of builds between fixes: **2.58 builds**
- The longest time taken to fix a failure is **6 days, 3 hours, 1 minute**, from failure starting in build **1**.
- The greatest number of builds taken to fix a failure is **26**, from failure starting in build **1**.

<table>
<thead>
<tr>
<th>Failed</th>
<th>Fixed</th>
<th>Time to Fix</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>902</strong> (Updated by jschumacher) Oct 9, 2006 2:05:41 PM - 1 day ago</td>
<td><strong>903</strong> (Manual build) Oct 9, 2006 3:11:02 PM - 1 day ago</td>
<td>1 builds 25 minutes</td>
</tr>
<tr>
<td><strong>B92</strong> (Updated by aro) Oct 5, 2006 1:19:41 PM - 5 days ago</td>
<td><strong>B94</strong> (Manual build) Oct 5, 2006 10:23:34 AM - 4 days ago</td>
<td>2 builds 21 hours, 3 minutes</td>
</tr>
</tbody>
</table>
Revised Project Editing

The new project editing facility makes it faster and easier to edit an existing project.

<table>
<thead>
<tr>
<th>Source Repository</th>
<th>Builder Configuration</th>
<th>Build Artifacts</th>
<th>Build Notification</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Repository:</strong></td>
<td>CVS</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>CVS Root:</strong></td>
<td>pserver:<a href="mailto:mchai@cvs.atlassian.com">mchai@cvs.atlassian.com</a>:/cvsroot/atlasci</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Module:</strong></td>
<td>confluence</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Web Repository URL:</strong></td>
<td><a href="http://resches.sydne">http://resches.sydne</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Change Detection:</strong></td>
<td>Triggered Build Strate</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Screenshots
Confluence 842: Build Result Summary

Changes

This build was the result of the following change:

tdavies | fix tests

When: Sep 26, 2006 10:49:24 AM - 2 weeks ago
Duration: 31 minutes

Tests

- Build 842 had a total of 1005 tests.
- All tests were successful.
- 2 failures occurring on the previous build are fixed in Build 842.

Fixed Tests

<table>
<thead>
<tr>
<th>Test</th>
<th>Failing Since Build</th>
</tr>
</thead>
<tbody>
<tr>
<td>TestOneConfluenceCachingBandanaPersister</td>
<td></td>
</tr>
<tr>
<td>Retrieve</td>
<td>CONF-635 (20 hours before)</td>
</tr>
<tr>
<td>Retrieve non serializable</td>
<td>CONF-841 (59 minutes before)</td>
</tr>
</tbody>
</table>

Build Errors

The build generated some errors. See the full build log for more details.

Note: Some input files use or override a deprecated API.
Note: Recompile with -deprecation for details.
Note: Some input files use or override a deprecated API.
Note: Recompile with -deprecation for details.
<table>
<thead>
<tr>
<th>Key</th>
<th>Summary</th>
<th>Priority</th>
<th>Status</th>
<th>Reporter</th>
</tr>
</thead>
<tbody>
<tr>
<td>BAM-283</td>
<td>Build results over 1000 doesn't display properly in some pages</td>
<td></td>
<td>Resolved</td>
<td>Ben Kuo</td>
</tr>
<tr>
<td>BAM-265</td>
<td>Save option when editing build details.</td>
<td></td>
<td>Resolved</td>
<td>Justen Stepka</td>
</tr>
<tr>
<td>BAM-262</td>
<td>Ability to suspend builds</td>
<td></td>
<td>Resolved</td>
<td>Mark</td>
</tr>
<tr>
<td>BAM-257</td>
<td>link error to jira</td>
<td></td>
<td>Resolved</td>
<td>Chaimungkalanont</td>
</tr>
<tr>
<td>BAM-256</td>
<td>New instance of bamboo, error upgrading allbuild.xml</td>
<td></td>
<td>Resolved</td>
<td>Riaz Khanmohamed</td>
</tr>
<tr>
<td>BAM-246</td>
<td>Changing context path seems to cause installation to fail</td>
<td></td>
<td>Resolved</td>
<td>Mark</td>
</tr>
<tr>
<td>BAM-245</td>
<td>Build Summary Page improvements</td>
<td></td>
<td>Resolved</td>
<td>Chaimungkalanont</td>
</tr>
<tr>
<td>BAM-244</td>
<td>Bread Crumbs</td>
<td></td>
<td>Resolved</td>
<td>Mark</td>
</tr>
<tr>
<td>BAM-242</td>
<td>properties file location incorrect</td>
<td></td>
<td>Resolved</td>
<td>Chaimungkalanont</td>
</tr>
<tr>
<td>BAM-235</td>
<td>Ant xml files can't be read</td>
<td></td>
<td>Resolved</td>
<td>Riaz Khanmohamed</td>
</tr>
<tr>
<td>BAM-231</td>
<td>Support period date in incorrect format</td>
<td></td>
<td>Resolved</td>
<td>Riaz Khanmohamed</td>
</tr>
<tr>
<td>BAM-197</td>
<td>Next and Previous build link for Test + TestCase screen</td>
<td></td>
<td>Resolved</td>
<td>Mark</td>
</tr>
<tr>
<td>BAM-192</td>
<td>Create Current Building Page</td>
<td></td>
<td>Resolved</td>
<td>Edwin Wong</td>
</tr>
<tr>
<td>BAM-184</td>
<td>Add a failing since link on Build Summary page</td>
<td></td>
<td>Resolved</td>
<td>Mark</td>
</tr>
<tr>
<td>BAM-6</td>
<td>Change edit project details from one wizard into a series of forms</td>
<td></td>
<td>Resolved</td>
<td>Scott Farquhar</td>
</tr>
<tr>
<td>BAM-279</td>
<td>Clicking on Build Link throws Internal Server error</td>
<td></td>
<td>Resolved</td>
<td>Jainthra Fernandes</td>
</tr>
<tr>
<td>BAM-273</td>
<td>Sending IM notifications to Google Talk</td>
<td></td>
<td>Resolved</td>
<td>Gwyn Evans</td>
</tr>
<tr>
<td>BAM-270</td>
<td>NPE - Failed to notify a build event notifier about the build completion of build “Topup EJBs”</td>
<td></td>
<td>Resolved</td>
<td>Gwyn Evans</td>
</tr>
<tr>
<td>BAM-250</td>
<td>Save functionality on every Page of Wizard</td>
<td></td>
<td>Resolved</td>
<td>Jainthra Fernandes</td>
</tr>
<tr>
<td>BAM-234</td>
<td>Cancelling a build part way is marked as a successful build</td>
<td>Resolved</td>
<td>Riaz Khanmohamed</td>
<td></td>
</tr>
<tr>
<td>BAM-227</td>
<td>build timer on project screen</td>
<td>Resolved</td>
<td>Riaz Khanmohamed</td>
<td></td>
</tr>
<tr>
<td>BAM-189</td>
<td>Mail template reporting wrong authors count</td>
<td>Resolved</td>
<td>Mark Chaimungkalanont</td>
<td></td>
</tr>
<tr>
<td>BAM-135</td>
<td>Option to disable projects</td>
<td>Resolved</td>
<td>Michael Mekaail</td>
<td></td>
</tr>
<tr>
<td>BAM-261</td>
<td>cvs branch/head consistency</td>
<td>Resolved</td>
<td>Riaz Khanmohamed</td>
<td></td>
</tr>
</tbody>
</table>
Bamboo 0.5 Upgrade Guide

This page last changed on Jan 17, 2007 by rosie@atlassian.com.

Upgrading from Bamboo 0.4 to 0.5

Please follow the Bamboo Upgrade Guide.

Upgrading from Bamboo 0.3 and earlier

In addition to the above, please read the Upgrade Guide for every version you are skipping during the upgrade. The complete list of Upgrade Guides is available here.
Bamboo 0.5.1 Release Notes

This page last changed on Mar 06, 2007 by rosie@atlassian.com.

⚠️ Bamboo 1.1 has now been released – see the [Bamboo 1.1 Release Notes](#)

⚠️ Upgrading? Please see the [Bamboo 0.5.1 Upgrade Guide](#).

Release 0.5.1 includes the following bug fixes.

### Bug fixes

<table>
<thead>
<tr>
<th>Key</th>
<th>Summary</th>
<th>Pr</th>
<th>Status</th>
<th>Reporter</th>
</tr>
</thead>
<tbody>
<tr>
<td>BAM-312</td>
<td>Test case page does not load correctly</td>
<td></td>
<td>Resolved</td>
<td>Edwin Wong</td>
</tr>
<tr>
<td>BAM-295</td>
<td>Not compatible with JDK 1.4</td>
<td></td>
<td>Resolved</td>
<td>Nick Sieger</td>
</tr>
<tr>
<td>BAM-272</td>
<td>start-bamboo.bat fails to run when the folder name has a _, and a _ in it</td>
<td></td>
<td>Resolved</td>
<td>Gwyn Evans</td>
</tr>
</tbody>
</table>
Bamboo 0.5.1 Upgrade Guide

This page last changed on Jan 17, 2007 by rosie@atlassian.com.

Upgrading from Bamboo 0.5 to 0.5.1

Please follow the Bamboo Upgrade Guide.

Upgrading from Bamboo 0.4 and earlier

In addition to the above, please read the Upgrade Guide for every version you are skipping during the upgrade. The complete list of Upgrade Guides is available here.
Bamboo 0.6 Release Notes

This page last changed on Mar 06, 2007 by rosie@atlassian.com.

⚠ Bamboo 1.1 has now been released – see the Bamboo 1.1 Release Notes

⚠ Upgrading? Please see the Bamboo 0.6 Upgrade Guide.

New in Release 0.6

JIRA issue integration

In this release, Bamboo has made initial steps to integrate with JIRA. Need to keep track of which JIRA issues builds were related to? Well now you can simply add the JIRA issue key with your commit messages and Bamboo will be able to link the each build result to issues in JIRA.

Changes

This build was the result of the following change:

mchaj Builds are now slightly more robust. A build that throws an exception will still complete (BAM-95) and also should now not send mail before it writes the build results (BAM-387)

You can also compare across all builds for build results relating to a particular JIRA issue for easy tracking of issue-build history.

BAM-6: Builds Results

The table below shows all the builds which is related to JIRA issue: BAM-6.

<table>
<thead>
<tr>
<th>Build Number</th>
<th>Reason</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>BAM-583</td>
<td>Updated by mchaj</td>
<td>2 months ago</td>
</tr>
<tr>
<td>BAM-581</td>
<td>Updated by mchaj</td>
<td>2 months ago</td>
</tr>
<tr>
<td>BAM-575</td>
<td>Updated by mchaj</td>
<td>2 months ago</td>
</tr>
<tr>
<td>BAM-574</td>
<td>Updated by mchaj</td>
<td>2 months ago</td>
</tr>
<tr>
<td>BAM-573</td>
<td>Updated by mchaj</td>
<td>2 months ago</td>
</tr>
<tr>
<td>BAM-572</td>
<td>Updated by mchaj</td>
<td>2 months ago</td>
</tr>
<tr>
<td>BAM-571</td>
<td>Updated by mchaj</td>
<td>2 months ago</td>
</tr>
</tbody>
</table>

Authors

Bamboo has now introduced the concept of authors which allows you to see all the build results across all builds that a user has contributed to (and broken!)
<table>
<thead>
<tr>
<th>Build</th>
<th>When</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>JIRA-1087</td>
<td>27 minutes ago</td>
<td>Change 'divs' to be tables again, as this fixes functional tests.</td>
</tr>
<tr>
<td>JIRA-1080</td>
<td>5 hours ago</td>
<td>Ajax javascript to load the issue operations in a div, rather than a whole</td>
</tr>
<tr>
<td></td>
<td></td>
<td>tests :)</td>
</tr>
<tr>
<td>JIRA-1021</td>
<td>1 week ago</td>
<td>Initial javascript for dynamic updating of the edit issue screens in JIRA.</td>
</tr>
<tr>
<td>JIRA-1014</td>
<td>1 week ago</td>
<td>Put the body content in a div</td>
</tr>
<tr>
<td>JIRA-1013</td>
<td>1 week ago</td>
<td>Produce some slightly nicer HTML</td>
</tr>
<tr>
<td>JIRA-1012</td>
<td>1 week ago</td>
<td>Remove 1.6% of javascript from every page by making the quicksearch j</td>
</tr>
<tr>
<td>JIRA-1003</td>
<td>1 week ago</td>
<td>A very basic database version of the comment RSS feed. Committing to</td>
</tr>
<tr>
<td>JIRA-995</td>
<td>1 week ago</td>
<td>Remove the need for an issue key which was just used for logging anyth</td>
</tr>
<tr>
<td>JIRA-909</td>
<td>1 week ago</td>
<td>JIRA-10640 - encode the file name, as windows doesn't like certain file na</td>
</tr>
<tr>
<td>JIRA-907</td>
<td>1 week ago</td>
<td>Use helper method, rather than duplicating code.</td>
</tr>
</tbody>
</table>

Look at the trends of broken vs fixed builds for each author!

**Breakages and Fixes**

- **Broken** means the build has failed but the previous build was successful.
- **Fixed** means that the build was successful but the previous build has failed.

**Broken By Author:** 12 (6% of all builds triggered)

**Fixed By Author:** 9 (5% of all builds triggered)

Difference of fixes and breaks: -3

And even and compare authors to see who has broken the most builds.

![Graph showing trends of broken builds](image)

Look towards the future as we move towards relating authors with user profiles so that more specific notifications can be sent.

**Improved Fisheye links**

Tracking code changes is now simpler as the code changes is now linked to the full file history, the code
changes and the current revision of each file.

If you are performing an upgrade from previous versions, Bamboo will upgrade your data the first time it starts up. This may take a while to complete if you have substantial amount of data.

With the addition of author level features across builds, users who are upgrading their installation of Bamboo from previous versions should ensure that they perform a full re-index operation again.

Other updates & Bug fixes

<table>
<thead>
<tr>
<th>Key</th>
<th>Summary</th>
<th>Pr</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>BAM-346</td>
<td>Catastrophic failure trying to view test results for a particular build on Panda</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-486</td>
<td>viewing build config for ant-based builds</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-394</td>
<td>FishEye file linking</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-383</td>
<td>Error when clearing build dependency list</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-377</td>
<td>error during page refresh</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-376</td>
<td>Confirmation message not given when deleting multiple builds</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-368</td>
<td>Inconsistencies on case-sensitive filesystems</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-361</td>
<td>Perforce provider stops working after Bamboo restart</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-353</td>
<td>default base url is invalid</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-351</td>
<td>build timer incorrect</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-349</td>
<td>side menubar issues</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-348</td>
<td>RSS feed link invalid</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-344</td>
<td>create build link wrong on dash</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-343</td>
<td>Build Queues should be able to choose which projects it would accept</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-341</td>
<td>Cannot change version repositories</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-333</td>
<td>An issue view page for Build Results</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-332</td>
<td>Retrieve issue information through RPC from JIRA</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-331</td>
<td>Setup JIRA instance administration</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>Issue</td>
<td>Description</td>
<td>Resolution</td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------------</td>
<td></td>
</tr>
<tr>
<td>BAM-330</td>
<td>Parse commit messages for issue keys</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-329</td>
<td>Displaying related JIRA issues in Bamboo</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-321</td>
<td>Fix up create &amp; edit Screens</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-308</td>
<td>Edit Builder Configuration causes error in .5</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-302</td>
<td>quick-search to non-existent builds throws error</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-266</td>
<td>Spacing error on Configuration -&gt; Build Notifications listing</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-190</td>
<td>Create Author Pages</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-95</td>
<td>Build hangs if exception is thrown during build</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-38</td>
<td>Add Bash as a default custom command builder</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-395</td>
<td>Encryption exceptions after a mail send failure</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-375</td>
<td>Use NumberFormat &quot;#&quot; for build numbers in instant messages</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-300</td>
<td>Log error catching</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-292</td>
<td>Disabled build should also hide the build option</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-208</td>
<td>Build table doesn't always refresh correctly</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-314</td>
<td>double slash (&quot;/&quot; ) in link</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-290</td>
<td>Bamboo checklist page doesn't give example of bamboo, home property</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-271</td>
<td>Error page should link back to the home page</td>
<td>Resolved</td>
<td></td>
</tr>
</tbody>
</table>
Bamboo 0.6 Upgrade Guide

Please follow the Bamboo Upgrade Guide.

Upgrading from Bamboo 0.5.1 to 0.6

In addition to the above, please read the Upgrade Guide for every version you are skipping during the upgrade. The complete list of Upgrade Guides is available here.
Bamboo 0.7 Release Notes

This page last changed on Mar 06, 2007 by rosie@atlassian.com.

⚠️ Bamboo 1.1 has now been released – see the [Bamboo 1.1 Release Notes](#)

⚠️ Upgrading? Please see the [Bamboo 0.7 Upgrade Guide](#).

New in Release 0.7

Cross Build Reports

In this release, Bamboo introduce a new build report generator. Use the report generator to identify patterns, linkages, and trends on key build metrics such as: build success ratio, number of failures, number of tests, and build duration.

<table>
<thead>
<tr>
<th>Report Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Report:</strong> Percentage of Successful Builds</td>
</tr>
<tr>
<td><strong>Builds:</strong> Build Duration, Build Activity, Number of Build Failures, Number of Tests, Percentage of Successful Builds</td>
</tr>
</tbody>
</table>

| Main Build |
| Atlassian Spring |
| Main Build |
| atlassian-renderer |
| Main Build |

The report generator also allow comparison of these trends across different builds.
Percentage of Successful Builds

Comparing success percentages gives you an idea of how stable a build is compared to seriously wrong.

<table>
<thead>
<tr>
<th>Chart</th>
<th>Data Table</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="chart_image" alt="Percentage of Successful Builds chart" /></td>
<td></td>
</tr>
</tbody>
</table>

Author to User Profile Linking

Extending on the concept of authors introduced in the previous release, Bamboo now allows you to link your author alias to your Bamboo user profile.

Linking your profile to an alias will enable you to view your builds (including most recent, broken, and fixed builds) in your profile page.

Edwin: User Profile

<table>
<thead>
<tr>
<th>User Details</th>
<th>Builds Summary</th>
<th>Last 10 Builds</th>
<th>Last 10 Broken</th>
</tr>
</thead>
<tbody>
<tr>
<td>Build</td>
<td>When</td>
<td>Comments</td>
<td></td>
</tr>
<tr>
<td>BAM-MAIN-78</td>
<td>6 months ago</td>
<td>Changed B.args bamboo</td>
<td></td>
</tr>
<tr>
<td>BAM-MAIN-57</td>
<td>6 months ago</td>
<td>Added HackUtils!</td>
<td></td>
</tr>
</tbody>
</table>

In addition, Bamboo now retrieves details from your user profile wherever your repository alias is referenced.

Looking forward, Bamboo will further enhance the personalized experience by introducing favourite projects, user avatars, and more.

Build Grouping at Project Level

In any development environment, it is very likely that one project will only consist of one build. For example, you may have a BRANCH build on top of your HEAD build for a given project. Bamboo release 0.7 allows you to group these builds together with the introduction of a new project level.
Confluence: Project Summary

<table>
<thead>
<tr>
<th>Status</th>
<th>Build</th>
<th>Latest</th>
<th>Last Ran</th>
</tr>
</thead>
<tbody>
<tr>
<td>✗</td>
<td>Head</td>
<td>CONF-HEAD-1</td>
<td>1 minute ago</td>
</tr>
<tr>
<td>✗</td>
<td>Branch</td>
<td>CONF-BRANCH-1</td>
<td>19 seconds ago</td>
</tr>
</tbody>
</table>

Special note for Beatlejuice users

Bamboo does not support upgrading from a previous Beatlejuice instance to Bamboo 0.7. You will need to upgrade Beatlejuice to a previous Bamboo version (0.5.2 is recommended) before upgrading to Bamboo 0.7.

Other updates & Bug fixes

<table>
<thead>
<tr>
<th>Key</th>
<th>Summary</th>
<th>Pr</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>BAM-463</td>
<td>Cannot disable Build Expiry</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-465</td>
<td>Dashboard ordered by project key rather than name</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-435</td>
<td>Write an upgrade documentation</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-386</td>
<td>Automate deploying for all Maven II Projects once build is successful</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-7</td>
<td>Use project IDs instead of names in all URLs</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-469</td>
<td>Build Reports</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-456</td>
<td>Code change filenames from CVS branch have &quot;Attic&quot; in the path</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-454</td>
<td>Improve page titles</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-453</td>
<td>Use profile names on View Authors page</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-452</td>
<td>Log the user who manually initiated or cancelled a build</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-446</td>
<td>New CVS builds build twice</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-444</td>
<td>Re-adding deleted builds (with same name/key) throws 500 error</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-438</td>
<td>Forgot Password email includes context path twice</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-426</td>
<td>Enabling build obscure</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-424</td>
<td>Add links to help files in add/edit build configuration screens</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-421</td>
<td>Perforce validation throws uncaught exceptions</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-412</td>
<td>Build edits trigger builds with ALL code changes</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-407</td>
<td>Unable to get change logs for Perforce</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-405</td>
<td>Problem loading project data - main project summary</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-374</td>
<td>Viewing test case across 90 days gets Internal server error</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-373</td>
<td>Tests that are not run are incorrectly labeled as 'fixed'</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-372</td>
<td>Having all queues disabled means that commits are lost during that period</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-355</td>
<td>InvocationTargetException during re-indexing - Out of memory</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-342</td>
<td>Upgrade tasks don't run in order</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-325</td>
<td>Upgrade task to move artifacts and build results</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-324</td>
<td>Add upgrade task to move Build Definitions</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-320</td>
<td>Build Definitions into Hibernate</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-318</td>
<td>Add tests graph to Tests page of build</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-274</td>
<td>Display related jiras in a build</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-224</td>
<td>Validation not working for edit configuration of build definitions</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-191</td>
<td>Create Commit Page</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-188</td>
<td>Add validation when adding artifacts</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-158</td>
<td>Distributions have confusing directory structure,</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-141</td>
<td>Introduce Project -&gt; Build Definition concept</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-92</td>
<td>Project added to build queue more than once</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-18</td>
<td>Changing Trigger type forces a new build</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>Issue</td>
<td>Description</td>
<td>Status</td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>-----------------------------------------------------------</td>
<td>--------</td>
<td></td>
</tr>
<tr>
<td>BAM-447</td>
<td>build key case issue</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-439</td>
<td>500 error on change password page when not logged in</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-431</td>
<td>Quick enable build option</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-423</td>
<td>Typos in &quot;no dependent builds&quot; message</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-415</td>
<td>Notification recipients cannot be cleared</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-414</td>
<td>Builds set to branch/tag by default when editing</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-406</td>
<td>ui build history boxes not aligned</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-86</td>
<td>Constant polling with queues disabled</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-437</td>
<td>Turn off HSQL query debugging</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-432</td>
<td>exe installer default path error</td>
<td>Resolved</td>
<td></td>
</tr>
</tbody>
</table>
**Special note for 0.7**

Bamboo 0.7 has a tremendous number of changes under the hood. If the upgrade fails, your data may NOT be recoverable (since it involves moving from the file system). Please ensure you backup before attempting an upgrade!

As part of the internal changes, if you have large amounts of data in Bamboo, upgrading may potentially take up to a few hours. The length of time taken to update Bamboo is dependent on the number of builds and build results your Bamboo instance contains. For example, updating a Bamboo instance with just over 3000 build results (aggregate of around 25 builds) took around half an hour. Re-indexing took approximately, a further 6 hours. So it may be best to leave this as an overnight job.

Please backup before attempting an upgrade!

---

**Upgrading from Bamboo 0.6 to 0.7**

Please follow the [Bamboo Upgrade Guide](#).

---

**Upgrading from Bamboo 0.5 and earlier**

In addition to the above, please read the Upgrade Guide for every version you are skipping during the upgrade. The complete list of Upgrade Guides is available [here](#).
Bamboo 0.8 Release Notes

This page last changed on Mar 06, 2007 by rosie@atlassian.com.

⚠ Bamboo 1.1 has now been released – see the Bamboo 1.1 Release Notes

⚠ Upgrading? Please see the Bamboo 0.8 Upgrade Guide.

The Bamboo 0.8 release is our biggest yet and is chock full of new features!

Brand spanking new UI

Bamboo 0.8 comes with a new modern look and feel as well as usability improvements.

- Clear, concise summary of your builds
- Latest status widget - see the status of your build, anywhere, anytime.
- Alt + E (this depends on your browser]) for quick edit of your Build Plan from any build screen

Labels Labels Labels

Get organised with labels! Easily create labels for builds and view the most frequently used labels.

**Labels**

This page lists all labels against builds in plan Test - Coverage Test. The bigger the text, the more builds associated with it.

View the labels: Alphabetically | By Popularity

0 1 cool 2 3 4 5 6 7 8 9 a
b c d e f g h i j k l m n o p q r s t u v
Bored of labelling everything yourself? You can even get Bamboo to automatically label for you!

**Build Result BAM-MAIn-1493**

Labels: release

Enter labels to add to this build:

- **Done**
- **Cancel**

**Clover Reports Plugin**

Check out the test coverage for your code! You can now get code coverage trends data at your finger tips. Is your team improving on their coverage? Or are things going south.

To enable, make sure your build produces Clover XML reports and configure their location in your build plan.

**Favourite Build Plans**

Too many projects starting to drown your dashboard? Choose your favourite build plans and cut down on the clutter.

<table>
<thead>
<tr>
<th>Status</th>
<th>Project - Plan</th>
<th>Latest Completed Build</th>
<th>Last Run</th>
<th>Reason</th>
<th>Duration</th>
<th>Test Count</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Bamboo - HEAD</td>
<td>BAM-MAIn-1493</td>
<td>1 hour</td>
<td>Updated by Bam</td>
<td>6 minutes</td>
<td>1265 passed</td>
<td><img src="image1" alt="icon" /></td>
</tr>
<tr>
<td></td>
<td>JIRA - Main Build</td>
<td>JIRA-MAIn-1254</td>
<td>1 hour</td>
<td>Updated by Jira</td>
<td>4 minutes</td>
<td>2684 passed</td>
<td><img src="image2" alt="icon" /></td>
</tr>
</tbody>
</table>

**Sloooowest running tests**

On top of the existing "Most number of failures" report for tests, there's also the slowest running tests. Find out what's throttling your build and nut it out!

<table>
<thead>
<tr>
<th>Test</th>
<th>Average Duration (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detects updates to a svn repository</td>
<td>21.25 (21 seconds)</td>
</tr>
<tr>
<td>Not delete all source directories when getting all source code</td>
<td>6.72 (6.72 seconds)</td>
</tr>
<tr>
<td>Ensures the web repository url is a valid url</td>
<td>6.47 (6.47 seconds)</td>
</tr>
<tr>
<td>Successful build with single target</td>
<td>4.53 (4 seconds)</td>
</tr>
</tbody>
</table>

**Pretty URLs**
Bamboo is now armed with "Nice URLs" (e.g. browse/BAM-MAIN-1465/test for test of build BAM-MAIN-1465). Not only do they make you feel warm and fuzzy on the inside, they can keep you better organised and links are much easier to share and guess.

Projects, Plans, Builds

For 0.8, we've renamed our key concepts to make it far less confusing for everyone. Now, Projects (e.g. Confluence) can have multiple Plans (e.g. Unit Test, Functional test) which then gets run to produce Builds (e.g. CONF-UNIT-234).

We've also made a myriad of general improvements:

- Rename and move projects and plans
- New Build Strategies
  - Daily Builds - Quick and easy way to run a build at a time in the day
  - Manual Build - Need to stop your build from being triggered? The manual build strategy lets you manually trigger off the build only!
  - Cron for flexible building - Have more complex scheduling needs? You can write your own Cron expressions to schedule your build anyway you like!
- XML Import and Export - stay safe and secure, backup that precious build data!

Legendary Pluggability

Bamboo now has more plugin points than you can poke a set of stumps at! In fact, our Clover Report Plugin was written entirely as a plugin.

We haven't had time to document them all yet, but they include arbitrary build actions, post build actions, custom builders and custom reports. Stay tuned!

Bamboo Plugin for JIRA 3.7

Just like Bamboo can show the JIRA keys related with a particular build, JIRA now has a plugin to show the related builds for a particular issue. Simply install the plugin to your JIRA instance, point to your Bamboo server in the JIRA administration and you can view the Bamboo builds as a tab panel for each issue.
BAM-6: Builds Results

The table below shows all the builds which are related to JIRA issue BAM-6.

<table>
<thead>
<tr>
<th>Build Number</th>
<th>Reason</th>
<th>Date</th>
<th>Duration</th>
<th>Test Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>BAM-MAIN-563</td>
<td>Updated by michai</td>
<td>3 months ago</td>
<td>2 minutes</td>
<td>1342 passed</td>
</tr>
<tr>
<td>BAM-MAIN-561</td>
<td>Updated by michai</td>
<td>3 months ago</td>
<td>2 minutes</td>
<td>4 out of 1342 failed</td>
</tr>
<tr>
<td>BAM-MAIN-575</td>
<td>Updated by michai</td>
<td>3 months ago</td>
<td>4 minutes</td>
<td>1342 passed</td>
</tr>
<tr>
<td>BAM-MAIN-574</td>
<td>Updated by michai</td>
<td>3 months ago</td>
<td>4 minutes</td>
<td>1342 passed</td>
</tr>
<tr>
<td>BAM-MAIN-573</td>
<td>Updated by michai</td>
<td>3 months ago</td>
<td>4 minutes</td>
<td>1342 passed</td>
</tr>
<tr>
<td>BAM-MAIN-572</td>
<td>Updated by michai</td>
<td>3 months ago</td>
<td>4 minutes</td>
<td>1342 passed</td>
</tr>
<tr>
<td>BAM-MAIN-571</td>
<td>Updated by michai</td>
<td>3 months ago</td>
<td>4 minutes</td>
<td>1342 passed</td>
</tr>
<tr>
<td>BAM-MAIN-565</td>
<td>Updated by michai</td>
<td>3 months ago</td>
<td>4 minutes</td>
<td>1341 passed</td>
</tr>
<tr>
<td>BAM-MAIN-563</td>
<td>Updated by michai</td>
<td>3 months ago</td>
<td>3 minutes</td>
<td>1340 passed</td>
</tr>
<tr>
<td>BAM-MAIN-562</td>
<td>Updated by michai</td>
<td>3 months ago</td>
<td>3 minutes</td>
<td>1340 passed</td>
</tr>
<tr>
<td>BAM-MAIN-561</td>
<td>Updated by michai</td>
<td>3 months ago</td>
<td>4 minutes</td>
<td>1340 passed</td>
</tr>
<tr>
<td>BAM-MAIN-567</td>
<td>Updated by michai</td>
<td>3 months ago</td>
<td>4 minutes</td>
<td>1340 passed</td>
</tr>
<tr>
<td>BAM-MAIN-565</td>
<td>Updated by michai</td>
<td>3 months ago</td>
<td>4 minutes</td>
<td>1340 passed</td>
</tr>
<tr>
<td>BAM-MAIN-559</td>
<td>Updated by michai</td>
<td>3 months ago</td>
<td>5 minutes</td>
<td>1340 passed</td>
</tr>
<tr>
<td>BAM-MAIN-531</td>
<td>Updated by edwin.ken, michai</td>
<td>4 months ago</td>
<td>2 minutes</td>
<td>1336 passed</td>
</tr>
<tr>
<td>BAM-MAIN-528</td>
<td>Updated by michai</td>
<td>4 months ago</td>
<td>2 minutes</td>
<td>1336 passed</td>
</tr>
<tr>
<td>BAM-MAIN-528</td>
<td>Updated by michai</td>
<td>4 months ago</td>
<td>2 minutes</td>
<td>1337 passed</td>
</tr>
<tr>
<td>BAM-MAIN-528</td>
<td>Updated by michai</td>
<td>4 months ago</td>
<td>5 minutes</td>
<td>1338 passed</td>
</tr>
<tr>
<td>BAM-MAIN-471</td>
<td>Updated by michai</td>
<td>4 months ago</td>
<td>5 minutes</td>
<td>1330 passed</td>
</tr>
</tbody>
</table>

Other Updates and Bug Fixes

<table>
<thead>
<tr>
<th>Key</th>
<th>Summary</th>
<th>Pr</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>BAM-382</td>
<td>error viewing charts throws error</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-468</td>
<td>Reloading pages from remote machines is very slow.</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-532</td>
<td>Longest Running test report</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-531</td>
<td>getBuildResults returns an empty error list if there are problems</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-529</td>
<td>Internet Explorer crashes on Build Configuration Page</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-527</td>
<td>Default plan should be called 'Default' rather than 'Main build'</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-524</td>
<td>Clear all errors option</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-522</td>
<td>Confusing label on an admin field</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-521</td>
<td>Typo in report name</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-519</td>
<td>Ability to select Favourite Builds (Plans)</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-516</td>
<td>Add tooltips to the build results navigator</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-512</td>
<td>Internal Server Error on editing build for Confluence (PostgreSQL, Tomcat) project</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-510</td>
<td>UI Improvements</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-503</td>
<td>&quot;Cannot find or execute 'null' at 'null' &quot; while trying to create a new build</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-502</td>
<td>Additional Build Strategies</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-494</td>
<td>Wrong title on 'Test Running Duration' graph</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-487</td>
<td>Server error creating build</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-482</td>
<td>Allow moving of builds between projects</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-481</td>
<td>Checking out non-existent module causes infinite building loop</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-476</td>
<td>Manually stopping a build throws error</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-475</td>
<td>Nice / Pretty URLs</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-473</td>
<td>Tagging Build Results</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-471</td>
<td>Deleting builds may result in a non existent build queued in pipelines</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-467</td>
<td>Build Expire Seems running in an infinite loop</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-443</td>
<td>Restrict height of &lt;pre&gt; &quot;windows&quot; such that you don't need to scroll down to access the horizontal scrollbar</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-416</td>
<td>Stopping a build puts a ThreadDeath error on the dashboard</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-384</td>
<td>Current build status always available</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-365</td>
<td>Extensible system for collecting project metrics across builds</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-340</td>
<td>Bamboo build panel for JIRA issues</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-148</td>
<td>Add 'change detection' option for manual build only</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>Issue</td>
<td>Description</td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-99</td>
<td>XML Export</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-578</td>
<td>Bamboo not displaying</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-528</td>
<td>webapp hangs on refresh occasionally</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-508</td>
<td>Back button when Creating a Build</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-501</td>
<td>Project/Build Terminology Confusing</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-488</td>
<td>configuring artifacts - links that do nothing and navigation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-457</td>
<td>Manual build does not pick up changes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-397</td>
<td>link to latest live build</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-371</td>
<td>Poll at specific times</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-143</td>
<td>Update Quartz usage to utilise Spring</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-48</td>
<td>Allow renaming of projects</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Resolved
Bamboo 0.8 Upgrade Guide

If any of your project or plan keys have a "-" in them, try the workaround detailed at http://jira.atlassian.com/browse/BAM-600. Affects upgrades from 0.7.1 or below to 0.8 or higher.

Upgrading from Bamboo 0.7 to 0.8

Please follow the Bamboo Upgrade Guide.

Upgrading from Bamboo 0.6 and earlier

In addition to the above, please read the Upgrade Guide for every version you are skipping during the upgrade. The complete list of Upgrade Guides is available here.
Bamboo 0.9 Release Notes

This page last changed on Mar 06, 2007 by rosie@atlassian.com.

⚠️ Bamboo 1.1 has now been released – see the Bamboo 1.1 Release Notes

⚠️ Upgrading? Please see the Bamboo 0.9 Upgrade Guide.

The Bamboo 0.9 release is our fastest iteration ever and brings about small but important changes. It is primarily a maintenance release but contains the ability to add comments, a feature we are sure will be popular among teams of developers.

Comments Comments Comments

<table>
<thead>
<tr>
<th>Status</th>
<th>Project - Plan</th>
<th>Latest Completed Build</th>
<th>Last Run</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>🚫</td>
<td>Bamboo - HEAD</td>
<td>BAM-MAIN-1585</td>
<td>2 days ago</td>
<td>Updated by Ben Kuo</td>
</tr>
<tr>
<td>✔️</td>
<td>Confluence - Main Build</td>
<td>CCNF-MAIN-1241</td>
<td>2 days ago</td>
<td>Updated by Christopher Owen</td>
</tr>
<tr>
<td>✔️</td>
<td>JIRA - Main Build</td>
<td>JIRA-MAIN-1277</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Successful builds are green, failed builds are red. Feed for all builds or all failed builds.

Previously we brought you organisation and communication through labels. Now we bring you the ability to comment on a build. No more puzzling over what's happening about that broken build. Just simply attach a comment quickly to the broken build result and your team members will be kept informed.

**Build Result BAM-MAIN-1585**

Labels: NONE  Add

<table>
<thead>
<tr>
<th>Summary</th>
<th>Tests</th>
<th>Changes</th>
<th>Artifacts</th>
<th>Logs</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1585</td>
</tr>
</tbody>
</table>

Comments

 Edwin Wong (Dec 21, 2006 7:11:16 PM)
 I'm glad you like them. I'm the one that implemented comments =]

 Ben Kuo (Dec 21, 2006 7:07:52 PM)
 Cool comments. I broke the build!

Maintenance

This release has been primarily a maintenance release involving tasks such as cleaning up the new user interface. Some of these changes include pagination so that some pages that used to take forever to load (such as the completed builds page) now load in a much more satisfactory time frame.
## Other Updates and Bug Fixes

<table>
<thead>
<tr>
<th>Key</th>
<th>Summary</th>
<th>Pr</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>BAM-593</td>
<td>Fatal upgrade error 0.7 - 0.8</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-40</td>
<td>CVS Passwords displayed in plain text</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-590</td>
<td>Users with a space in their full name will cause perforce validation to fail</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-587</td>
<td>Chart URLs broken</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-584</td>
<td>Maven builder is only recognized as valid by checking for maven.bat file.</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-581</td>
<td>Incorrect/confusing error message</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-576</td>
<td>Create Time to fix report</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-573</td>
<td>Crowd integration break</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-567</td>
<td>Web Repository URL should ensure that it doesn't create double slashes for the URL</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-562</td>
<td>Error when viewing the Tests Summary tab</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-551</td>
<td>Make the form styles match the new UI</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-550</td>
<td>Hibernate exceptions (duplicates) on setting favourites after import</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-544</td>
<td>Authors pages should have tabs</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-541</td>
<td>Tweaks for the reports page</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-540</td>
<td>Minor tweaks for the Authors page</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-535</td>
<td>Import does not import labels / labellings</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-534</td>
<td>Nice / Pretty URLs to build results by jira issues</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-506</td>
<td>Build process hangs indefinitely</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-474</td>
<td>Allow commenting on build results</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-466</td>
<td>Permissions on checkout error</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-315</td>
<td>Pagination for the build results history table</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>Issue</td>
<td>Description</td>
<td>Resolution</td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------------</td>
<td></td>
</tr>
<tr>
<td>BAM-289</td>
<td>Current building page should differentiate between a build being built, and in a queue.</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-168</td>
<td>IM Server passwords in plaintext</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-163</td>
<td>Passwords are displayed in plaintext</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-598</td>
<td>Cannot delete build results when no index file exists</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-580</td>
<td>Help link issue</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-539</td>
<td>Novell Groupwise IM integration</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-530</td>
<td>Dashboard shouldn't call SetFavouriteAction so many times</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-526</td>
<td>Error setting up subversion with SSH</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-319</td>
<td>Pagination for summary/results page</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-591</td>
<td>Build queue configuration forgets which plans you had selected</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-565</td>
<td>Disabled builds should be greyed</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-546</td>
<td>Some comments around getLatestBuildResults</td>
<td>Resolved</td>
<td></td>
</tr>
</tbody>
</table>
Bamboo 0.9 Upgrade Guide

Upgrading from Bamboo 0.8 to 0.9

Please follow the Bamboo Upgrade Guide.

Please note that, after upgrading to Bamboo 0.9, the admin menu is slightly misaligned in Internet Explorer (please see BAM-597).

Known problems

[BAM-597] The admin menu is slightly misaligned in Internet Explorer

Upgrading from Bamboo 0.7 and earlier

In addition to the above, please read the Upgrade Guide for every version you are skipping during the upgrade. The complete list of Upgrade Guides is available here.
Bamboo 1.0 Release Notes

This page last changed on Mar 06, 2007 by rosie@atlassian.com.

⚠️ Bamboo 1.1 has now been released – see the Bamboo 1.1 Release Notes

Atlassian is proud to announce the final release of Bamboo 1.0! Bamboo 1.0 is the first official release of Atlassian's new Continuous Integration and Build Server.

Bamboo is more than just a build server — it is an entire Build Telemetry system designed to provide you with unprecedented insight into your development processes.

To check out Bamboo's features and see what it can do for you, please visit our Feature Tour.

⚠️ Upgrading from a pre-release version? Please see the Bamboo 1.0 Upgrade Guide.

⚠️ Doing an upgrade? Make sure you re-index Bamboo by going to the Administration section and hitting 'Re-index'.

Changes since RC2

The final steps to 1.0 since RC2 has been focused on resolving issues. Release 1.0 includes over 30 issues resolved.

In addition, the 1.0 release also sports another revised "All Plans" tab in the dashboard.
Other updates and bug fixes.

<table>
<thead>
<tr>
<th>Key</th>
<th>Summary</th>
<th>Pr</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>BAM-875</td>
<td>User page no longer show tabs with author information on them</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-857</td>
<td>Document our external Javascript widgets</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-821</td>
<td>Unable to export build configuration - no info on how to repair</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-879</td>
<td>Where are my nice build result commit message tool tips?</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-878</td>
<td>NumberFormatException for Test</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-876</td>
<td>Move the Clover plugin to opensource as an example of a Bamboo plugin</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-873</td>
<td>All Projects table shows the 'little hand' icon over rows that can't be expanded</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>Issue ID</td>
<td>Description</td>
<td>Resolution</td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------------</td>
<td></td>
</tr>
<tr>
<td>BAM-867</td>
<td>Test mail should contain clickable base url</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-860</td>
<td>New more condensed dashboard</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-839</td>
<td>Tests Page Has URL Escapes</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-837</td>
<td>Allow Properties to be Passed to Ant</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-835</td>
<td>Build completed time on summary page is actually build start time</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-803</td>
<td>Use minified version of js libs in 1.0 final</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-796</td>
<td>Number Format Exception</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-788</td>
<td>IM bot should reconnect before sending message if it was disconnected</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-783</td>
<td>Build test result tab taking to long to load</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-779</td>
<td>Report graphs are not displaying data: build duration and # of tests</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-778</td>
<td>Clicking the previous build button while viewing changes - got stacktrace</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-775</td>
<td>Disabled plans should have visual cue on the Summary page</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-723</td>
<td>Plugin Guide</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-722</td>
<td>Administrator's Guide</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-721</td>
<td>Bamboo User Guide</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-720</td>
<td>Upgrade Guide (generic)</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-719</td>
<td>Release Notes &amp; Upgrade Guides: reformat as per JIRA's/Confluence's</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>BAM-718</td>
<td>Installation Guide (WAR)</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-717</td>
<td>Installation Guide (Standalone)</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>BAM-706</td>
<td>Added two builds to the queue, canceled the 2nd one, got a hibernate</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td></td>
<td>exception</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-628</td>
<td>Test reponsibility for a build summary</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-610</td>
<td>HTML in test output log is doubly-escaped</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-16</td>
<td>Ability to externally embed full build status</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-807</td>
<td>Bamboo passes bad parameter diff ViewVC</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-770</td>
<td>Bamboo User ID should also be a repository alias</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>Ticket</td>
<td>Description</td>
<td>Resolution</td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>------------------------------------------------------------------------------</td>
<td>------------</td>
<td></td>
</tr>
<tr>
<td>BAM-690</td>
<td>Improve validation for CVS :ext</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-499</td>
<td>Use 307 response code instead of meta-refresh when hitting path of bamboo</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-433</td>
<td>exe installer home setting issue</td>
<td>Resolved</td>
<td></td>
</tr>
</tbody>
</table>
Bamboo 1.0 Upgrade Guide

This page last changed on Feb 20, 2007 by edwin@atlassian.com.

Upgrading from Bamboo 1.0-RC2 to 1.0

Please follow the Bamboo Upgrade Guide

⚠️ You will need to reindex your data after the upgrade is complete and Bamboo has started. To do this, go to the indexing page under the Administration section in Bamboo.

Upgrading from Bamboo 1.0-RC1 and earlier

In addition to the above, please read the Upgrade Guide for every version you are skipping during the upgrade. The complete list of Upgrade Guides is available here.
Bamboo 1.0-Beta Release Notes

This page last changed on Mar 06, 2007 by rosie@atlassian.com.

⚠️ Bamboo 1.1 has now been released – see the Bamboo 1.1 Release Notes

The Atlassian Bamboo team is proud to announce the release of Bamboo 1.0 beta. This release includes over 40 bug fixes and improvements.

⚠️ Upgrading? Please see the Bamboo 1.0-Beta Upgrade Guide.

New in Release 1.0 - Beta

Anonymous access and sign on control.

In this release, you can now control whether your Bamboo is a public or private instance via the anonymous access and sign on options. Anonymous access allows users not signed in to view read only sections of Bamboo. Sign on allows users to create their own account for login. Disable these options to fully protect your Bamboo instance.

Auto favourite feature

Bamboo gets smarter with an auto-favourite marking feature. It'll mark those builds you commit against as your favourites.

Longest time to fix tests

Get a view of which tests in your builds are taking the longest the fix.

Other updates and bug fixes

<table>
<thead>
<tr>
<th>Key</th>
<th>Summary</th>
<th>Pr</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>BAM-697</td>
<td>CVS connection fails if password has @ in it</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-652</td>
<td>Checkboxes dont work properly when removing dependant builds</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-644</td>
<td>Script builder fails for windows</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-616</td>
<td>Capture code changes for dependent and scheduled builds</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-566</td>
<td>The back button on create plan wizard clears previously selected values</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-162</td>
<td>Passwords in plaintext</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-714</td>
<td>Support for SVN File Protocol</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-693</td>
<td>If project only has one plan, the project summary should redirect to the Plan Summary Page</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-685</td>
<td>Commit comments lose line breaks</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-684</td>
<td>Can't add builder of type Ant</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-676</td>
<td>Unable to re-index due to locked file</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-674</td>
<td>Build fails to start</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-661</td>
<td>Security and Login Improvements</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-656</td>
<td>No way to 'complete' setting up a project as 'save' hidden by javascript</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-647</td>
<td>Split webapp WAR module into a JAR and a WAR module</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-637</td>
<td>Reports build table not sorted</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-635</td>
<td>Auto favourite functionality</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-632</td>
<td>Breadcrumbs should have build numbers</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-631</td>
<td>Reports on top ten longest time to fix for tests</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-630</td>
<td>Test summary page still using the old style</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-625</td>
<td>Redirect after a plan is created</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-624</td>
<td>IE caches ajax response for comments and labels</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-622</td>
<td>Last screen of create build broken on Safari</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-609</td>
<td>error displaying build queue admin page</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-608</td>
<td>too many files open error</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-606</td>
<td>Invalid path to clover throws error</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-595</td>
<td>Adding a comment from the Summary page doesn't work in IE</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-572</td>
<td>Need option to disable signups</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>Issue</td>
<td>Description</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td>-------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-570</td>
<td>Testing for mail and IM servers should be more visible</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-569</td>
<td>Auto report grouping for Tests doesn't seem to work</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-563</td>
<td>Validation for report not being selected</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-543</td>
<td>Minor tweaks of the Admin pages</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-542</td>
<td>Plugin Points for Web Fragments</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-490</td>
<td>Ability to run a Bamboo in 'private mode'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-91</td>
<td>Static files are not cached, increases size of downloads</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-668</td>
<td>JIRA tabs shows up regardless</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-659</td>
<td>Edit configuration needs formatting fixes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-612</td>
<td>Label grammar</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-611</td>
<td>Allow two character plan keys</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-603</td>
<td>Accessing /api/index.action throws a freemarker error</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-602</td>
<td>Minor issues with the build status wizard section on the editBuildConfiguration.action page</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-658</td>
<td>Project names not ordered in dropdown to create project</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-657</td>
<td>Typo and user interface improvement</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-579</td>
<td>Make top right options clearer</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Bamboo 1.0-Beta Upgrade Guide

This page last changed on Jan 17, 2007 by rosie@atlassian.com.

Upgrading from Bamboo 0.9 to 1.0-Beta

Please follow the Bamboo Upgrade Guide.

Upgrading from Bamboo 0.8 and earlier

In addition to the above, please read the Upgrade Guide for every version you are skipping during the upgrade. The complete list of Upgrade Guides is available here.
Bamboo 1.0-RC1 Release Notes

This page last changed on Mar 06, 2007 by rosie@atlassian.com.

⚠️ Bamboo 1.1 has now been released – see the Bamboo 1.1 Release Notes

The Atlassian Bamboo team is proud to announce the release of Bamboo 1.0 RC-1.

⚠️ Upgrading? Please see the Bamboo 1.0-RC1 Upgrade Guide.

New in Release 1.0 - RC1

New Bamboo Dashboard

Bamboo now showcases a brand new feature packed dashboard, organized to give you all your vital build information at a glance.

Atlassian Bamboo

"My Bamboo"

The "My Bamboo" tab is tailored to provide you with the most relevant information for you. Keep check of those build plans you really care about with the "Favourites" portlet.

My Favourite Builds

| Condence | 1 out of 1 build plans failing. |
| Test      | Expand |
| Blah > TST-BLAH-61 | Collapse |
| Run: 1 hour ago | Manual build | Duration: 11 seconds | Tests: 2 passed |

| Coverage Test > TST-COVER-184 |
| Run: 4 hours ago | Updated by admin | Duration: 11 seconds | Tests: 2 passed |

The "Broken Builds" portlet highlights those build plans currently broken which you contributed to.
My Broken Builds

<table>
<thead>
<tr>
<th>Test</th>
<th>Collapse</th>
</tr>
</thead>
<tbody>
<tr>
<td>🟧 Blah &gt; TST-BLAH-61</td>
<td>🟧 Coverage Test &gt; TST-COVER-184</td>
</tr>
</tbody>
</table>

Ran: 1 hour ago | Manual build | Duration: 11 seconds | Tests: 2 passed
Ran: 4 hours ago | Updated by admin | Duration: 11 seconds | Tests: 2 passed

Keep check of all your recent commits with "My Changes" portlet.

My Latest Changes

<table>
<thead>
<tr>
<th>Build</th>
<th>When</th>
<th>Comments</th>
<th>Test Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONF-HEAD-22</td>
<td>4 hours ago</td>
<td>Brydie Just Committed this - Fake Jira Number (BAM-6656)</td>
<td>[details] 2 passed</td>
</tr>
<tr>
<td>DD-BB-2</td>
<td>4 hours ago</td>
<td>Brydie Just Committed this - Fake Jira Number (BAM-6656)</td>
<td>[details] 2 passed</td>
</tr>
<tr>
<td>TST-DEF-27</td>
<td>4 hours ago</td>
<td>Brydie Just Committed this - Fake Jira Number (BAM-6656)</td>
<td>[details] 2 passed</td>
</tr>
</tbody>
</table>

"All Plans"

The all build plans list is now slicker and grouped hierarchically by project. Collapse those projects you are not concerned about. Bamboo even remembers which ones you have collapsed.

"Current Activity"

See what's building on your Bamboo instance with "Current Activity" tab. It also shows a handy list of those build plans which recently completed a build.
**Recently Completed Builds**

<table>
<thead>
<tr>
<th>Build Number</th>
<th>Reason</th>
<th>Date</th>
<th>Duration</th>
<th>Test Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>TST-BLAH-61</td>
<td>Manual build</td>
<td>1 hour ago</td>
<td>11 seconds</td>
<td>2 passed</td>
</tr>
<tr>
<td>TST-BLAH-60</td>
<td>Manual build</td>
<td>1 hour ago</td>
<td>11 seconds</td>
<td>2 passed</td>
</tr>
<tr>
<td>TST-BLAH-59</td>
<td>Manual build</td>
<td>1 hour ago</td>
<td>11 seconds</td>
<td>2 passed</td>
</tr>
<tr>
<td>TST-BLAH-58</td>
<td>Manual build</td>
<td>1 hour ago</td>
<td>11 seconds</td>
<td>2 passed</td>
</tr>
<tr>
<td>TST-BLAH-57</td>
<td>Manual build</td>
<td>1 hour ago</td>
<td>11 seconds</td>
<td>2 passed</td>
</tr>
<tr>
<td>CCNF-HEAD-33</td>
<td>Manual build</td>
<td>3 hours ago</td>
<td>37 seconds</td>
<td>No tests found!</td>
</tr>
<tr>
<td>CCNF-HEAD-32</td>
<td>Manual build</td>
<td>3 hours ago</td>
<td>54 seconds</td>
<td>2 passed</td>
</tr>
</tbody>
</table>

All the portlets on the dashboard are also dynamic - see your build status icons, build progress bars, build completed lists on the fly.

### Dynamic Updates

Getting sick of refreshing the screen to see the status of your build? Now, the latest status bar in the top right corner will always be up-to-date, without interfering with what you are doing.

**Latest Status:** TEST-DEF-58 is building

### JIRA portlets

Included with this release is a plugin for JIRA. This plugin allows you to put Bamboo portlets on your JIRA dashboard so you can better monitor your build plans.

There are two portlet types available, Bamboo Status and Bamboo Plan Summary.

#### Bamboo Status

This will display the current status of build plans in a list format. You can display all of the plans or just your favourites.
**Build Status Summary: Bamboo Keg (Favourite Plans)**

<table>
<thead>
<tr>
<th>Status</th>
<th>Project - Plan</th>
<th>Latest Completed Build</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Aliassian Log Analysis - Default</td>
<td>ALA-MAIN-5</td>
<td>Updated by Justen Stepka</td>
</tr>
<tr>
<td>✔️</td>
<td>Bamboo - Branch v0.6.x</td>
<td>BAM-BRANCH06-4</td>
<td>Updated by Ben Kuo</td>
</tr>
<tr>
<td>✔️</td>
<td>Bamboo - HEAD</td>
<td>BAM-MAIN-1847</td>
<td>Dependency build</td>
</tr>
<tr>
<td>❌</td>
<td>Test - Beta</td>
<td>TEST-BETA-3</td>
<td>Manual build</td>
</tr>
</tbody>
</table>

**Bamboo Plan Summary**

This will display a graphical summary of a specific build plan. You can also filter the build results to be shown in the graphs, just as you can in Bamboo. The two types of graphs are:

- Build Duration & Number of Failures per build
- % Successful Builds & Average Duration per Time Period.

**Plan Summary: BAM-MAIN**

You are welcome to try out the new JIRA portlet [here](#).

**Talk to Bamboo via IM**
Bamboo Release 1.0 - RC1 also introduces a new innovative way for you to interact with Bamboo - via IM. You can respond to Bamboo's IM notification message with commands to comment or label a build result.

Usage:

- comment [build key] <comment message>
- label [build key] <labels>

Entering a build key is optional. If none is specified, Bamboo will look up the last time it corresponded with you and the build that was in context. The context gets updated when either you specify a build key in your command, or when Bamboo sends you a notification about a particular build.

Other updates and bug fixes

<table>
<thead>
<tr>
<th>Key</th>
<th>Summary</th>
<th>Pr</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>BAM-784</td>
<td>Improve build list UI</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-780</td>
<td>Freemarker template error on AtlassianUser project on Keg</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>Issue</td>
<td>Description</td>
<td>Status</td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------</td>
<td></td>
</tr>
<tr>
<td>BAM-795</td>
<td>Project keys should be able to have 2 letters only</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-786</td>
<td>No tab initially selected on dashboard</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-765</td>
<td>Stacktrace if session times out in middle of build configuration wizard</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-759</td>
<td>Display build trigger IP address on 'view plan configuration' page</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-754</td>
<td>Add XFire (Crowd) dependency libraries and adjust configuration file to have stubbed out Crowd configurations.</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-751</td>
<td>Exception while cancelling build configuration / build artifacts</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-735</td>
<td>on main &quot;Build Results&quot; page, in right-hand &quot;percentage&quot; box, change &quot;Successful Runs&quot; to &quot;Successful Builds&quot;</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-733</td>
<td>A sucessfull building maven 2 multi-project build is failing due to missing test results</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-726</td>
<td>Re-insert missing bamboo-init.properties file</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-666</td>
<td>Improve email notifications</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-626</td>
<td>Ajaxify Dashboard</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-618</td>
<td>AJAXify favourites</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-613</td>
<td>Display project, plan hierarchy on dashboard</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-390</td>
<td>build queue/log out of sync with dashboard status</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-20</td>
<td>Build Queue sometimes doesn't match actual project building</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-744</td>
<td>Bamboo does not work properly with Opera 9.10</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-728</td>
<td>change &quot;build form&quot; to &quot;build plan&quot;</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-689</td>
<td>Rouque JNDI mail settings label</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-564</td>
<td>Live activity logs can sometimes error out</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-789</td>
<td>Clumsy wording in dependency build cause</td>
<td>Resolved</td>
<td></td>
</tr>
</tbody>
</table>
Bamboo 1.0-RC1 Upgrade Guide

This page last changed on Feb 01, 2007 by edwin@atlassian.com.

Upgrading from Bamboo 1.0-Beta to 1.0-RC1

Please follow the Bamboo Upgrade Guide.

Upgrading from Bamboo 0.9 and earlier

In addition to the above, please read the Upgrade Guide for every version you are skipping during the upgrade. The complete list of Upgrade Guides is available here.
Bamboo 1.0-RC2 Release Notes

This page last changed on Mar 06, 2007 by rosie@atlassian.com.

⚠️ Bamboo 1.1 has now been released – see the Bamboo 1.1 Release Notes

The Atlassian Bamboo team is proud to announce the release of Bamboo 1.0 RC-2.

⚠️ Upgrading? Please see the Bamboo 1.0-RC2 Upgrade Guide.

New in Release 1.0 - RC2

Bamboo 1.0-RC2 is mainly to address a few important bugs

- The memory leak issues that were introduced in Release 1.0 - RC1 have been resolved. You can now have our new feature packed dashboard without the memory overhead.
- There were some JDK1.4 incompatibility issues that have been addressed as part of this release

Bamboo Installer

Bamboo Release 1.0 - RC2 has had its standalone installer revamped. It is now encapsulated inside a Java Service Wrapper to allow you to install and run the standalone Bamboo as a service. The benefits of running Bamboo as a service include:
1. Bamboo happily runs in the background (no console window required)
2. Bamboo server does not have to be manually started every time you want to use it.

Of course we have not removed any of the current methods of running and installing the standalone Bamboo so you are still able to use Bamboo as you always have.

We have also made the start files available in your Windows start menu for easier access.

Other updates and bug fixes

<table>
<thead>
<tr>
<th>Key</th>
<th>Summary</th>
<th>Pr</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>BAM-832</td>
<td>Error after you submit &quot;Setup Administrator User&quot; page.</td>
<td>![ ]</td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-826</td>
<td>JDK1.4 Incompatible Thread.UncaughtExceptionHandler</td>
<td>![ ]</td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-824</td>
<td>Internal Server Error when accessing <a href="http://localhost:8085">http://localhost:8085</a></td>
<td>![ ]</td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-799</td>
<td>Security filter should not apply to REST calls</td>
<td>![ ]</td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-753</td>
<td>Standalone Jetty starts without configuration file</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-828</td>
<td>Significant client-side (browser) memory leaks</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-818</td>
<td>Contact Administrators should not be protected with security</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-809</td>
<td>Remove author related tabs from individual user's profile page</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-805</td>
<td>Firefox left on a Bamboo page leaks memory and hogs CPU</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-804</td>
<td>No program-start menu entry created on installation with the windows installer</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-801</td>
<td>Replace client side redirect from the index.htm with a server side one</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-797</td>
<td>The Expand and Collapse link on the View Build Page</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-773</td>
<td>Cannot pass java opts to standalone version</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-712</td>
<td>Shutdown script for Bamboo standalone server</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-629</td>
<td>Easier access to test histories</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-575</td>
<td>Sort the file listing in the artifact browser</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-459</td>
<td>Include other useful batches (install-as-service, etc.) in installation</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-169</td>
<td>Reimplement Java Service Wrapper to be bundled as part of the standard installer</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-700</td>
<td>Build progress is not accurate enough</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-621</td>
<td>Code change filenames from CVS branch have &quot;Attic&quot; in the display path</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-724</td>
<td>fix footer link &quot;Powered by Atlassian Bamboo&quot;</td>
<td>Resolved</td>
<td></td>
</tr>
</tbody>
</table>
Bamboo 1.0-RC2 Upgrade Guide

This page last changed on Feb 19, 2007 by rosie@atlassian.com.

Upgrading from Bamboo 1.0-RC1 to 1.0-RC2

Please follow the Bamboo Upgrade Guide

Upgrading from Bamboo 1.0-Beta and earlier

In addition to the above, please read the Upgrade Guide for every version you are skipping during the upgrade. The complete list of Upgrade Guides is available here.
Bamboo 1.0.1 Release Notes

This page last changed on Mar 14, 2007 by edwin@atlassian.com.

⚠️ Bamboo 1.1 has now been released – see the Bamboo 1.1 Release Notes

Atlassian is proud to announce the release of Bamboo 1.0.1! Bamboo 1.0.1 is largely a bug fix build with over 20 issues resolved, including:

- Support for SVN cached default authentication.
- IE7 Javascript issues.
- Startup Script issues.

New startup procedures for Mac OS X and Linux distributions

The Bamboo startup procedure for Mac OS X and Linux distributions have now changed. Instead of using the Java Service Wrapper by invoking `run-bamboo` (in Mac OS X) or `start-bamboo` in Linux, the default startup script has been replaced by a generic `bamboo.sh` script in the root Bamboo installation folder. Using this script bypasses the Java Service Wrapper.

Usages for bamboo.sh

- `start` - starts Bamboo
- `stop` - stops Bamboo
- `console` - runs Bamboo in the console
- `status` - checks the status of Bamboo.

The Java Service Wrapper is still available, and you can startup Bamboo with it if you so choose. To do this, simply run your startup command in the `/wrapper` folder rather than the installation root folder.

Updates and Issues fixed.

<table>
<thead>
<tr>
<th>Key</th>
<th>Summary</th>
<th>Pr</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>BAM-943</td>
<td>Sessions need to closed in the finally block</td>
<td><img src="image" alt="pr" /></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-937</td>
<td>Importing data doesn't guarantee unique ids</td>
<td><img src="image" alt="pr" /></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-935</td>
<td>SVN Repository doesn't use default authentication</td>
<td><img src="image" alt="pr" /></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-958</td>
<td>Links to source is broken</td>
<td><img src="image" alt="pr" /></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-939</td>
<td>Need to convert build level plugins to use web fragments</td>
<td><img src="image" alt="pr" /></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-908</td>
<td>Standalone Bamboo cannot start in certain</td>
<td><img src="image" alt="pr" /></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-889</td>
<td>Linux environments Precedence: bulk mail header causing notifications to be blocked</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-975</td>
<td>Edit Configuration broken</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-965</td>
<td>Plan Summary does not render in IE 7 when logged in as a user</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-936</td>
<td>Smack Client does not recognize project/plan keys with numbers</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-933</td>
<td>Export and Import doesn't work when moving to a new Bamboo Home path</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-925</td>
<td>Viewing User via authors and via profile need to be separate requests as different info is needed in both</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-917</td>
<td>If an initial build has no queues to go into, it may cause repeated clean builds</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-899</td>
<td>Unable to Edit Build Configuration</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-897</td>
<td>Null pointer exception creating a build plan that uses svn+ssh</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-851</td>
<td>Bamboo cannot run on 64-bit linux machines</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-848</td>
<td>Responding via IM to build notifications is unreliable, the comment is not ascribed to me</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-841</td>
<td>IE7 Fails Often, Cancels Page</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-790</td>
<td>IE 7 sometimes can't display build page</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-746</td>
<td>Headless Unix Server</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-692</td>
<td>Manual build strategy gobbles up CVS errors</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-634</td>
<td>Sample plugin: Out of Memory tagging</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-297</td>
<td>&quot;Disabled&quot; status should be noted prominently on build summary</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-90</td>
<td>Lower priority of spawned build processes</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-959</td>
<td>Broken builds have incorrect links when restarting builds</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-918</td>
<td>Builders edit screen not populating existing</td>
<td>Resolved</td>
<td></td>
</tr>
</tbody>
</table>
BAM-916
Subversion Event Handler is not all that Null safe
Resolved

BAM-869
Error creating new build plan
Resolved
Bamboo 1.0.1 Upgrade Guide

Upgrading from Bamboo 1.0 to 1.0.1

Please follow the Bamboo Upgrade Guide

⚠️ You will need to reindex your data after the upgrade is complete and Bamboo has started. To do this, go to the indexing page under the Administration section in Bamboo.

Upgrading from Bamboo 1.0 and earlier

In addition to the above, please read the Upgrade Guide for every version you are skipping during the upgrade. The complete list of Upgrade Guides is available here.
Bamboo 1.0.2 Release Notes

This page last changed on Mar 28, 2007 by edwin@atlassian.com.

⚠️ Bamboo 1.1 has now been released – see the Bamboo 1.1 Release Notes

Atlassian is proud to announce the release of Bamboo 1.0.2! Bamboo 1.0.2 is mainly a bug fix release with over 10 issues resolved.

In addition, Bamboo 1.0.2 also sees added support for ssh private key authentication for both Subversion and CVS repositories.

### Updates and Issues fixed

<table>
<thead>
<tr>
<th>Key</th>
<th>Summary</th>
<th>Pr</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>BAM-982</td>
<td>Bamboo fails to start under JDK 1.4</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1098</td>
<td>No page associated with this URI</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1026</td>
<td>Links Are Incorrect When Using 'latest' as the build in the URL</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1023</td>
<td>Internal error when deleting plans with dependencies</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1006</td>
<td>Cannot view logs of latest build</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1002</td>
<td>Perforce commands need better logging</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-994</td>
<td>Internal server error when trying to view TestData history</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-989</td>
<td>Duplicate JARs in classpath</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-988</td>
<td>Error when entered License Key</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-987</td>
<td>FreeMarker template error in plan summary</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-983</td>
<td>Freemarker template error when viewing 'latest' builds</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-962</td>
<td>Provide ability to point to a CVS tag instead of HEAD/Branch</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-852</td>
<td>Internal error after &quot;Specify Source Repository&quot;</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-806</td>
<td>Support for SSH private</td>
<td></td>
<td>Resolved</td>
</tr>
</tbody>
</table>
key authentication
(possible using jsch)

BAM-791  
svn+ssh support for private key auth

BAM-691  
CVS Client should use the CVS_RSH environment variable if available

BAM-1031  
Clicking on latest build from home screen does not render the Artifacts or JIRA tabs

BAM-980  
Clicking on Tests tab in Build Result Summary renders empty screen

Resolved
Resolved
Resolved
Resolved
Bamboo 1.0.2 Upgrade Guide

This page last changed on Mar 14, 2007 by edwin@atlassian.com.

Upgrading from Bamboo 1.0.1 to 1.0.2

Please follow the Bamboo Upgrade Guide

Upgrading from Bamboo 1.0.1 and earlier

In addition to the above, please read the Upgrade Guide for every version you are skipping during the upgrade. The complete list of Upgrade Guides is available here.
Bamboo 1.0.3 Release Notes

This page last changed on Apr 12, 2007 by edwin@atlassian.com.

⚠️ Bamboo 1.1 has now been released – see the Bamboo 1.1 Release Notes

Atlassian is proud to announce the release of Bamboo 1.0.3! Bamboo 1.0.3 is mainly a bug fix release with over 10 issues resolved.

In this release, the focus has been on improving SVN integration (detection of SVN Externals) and CVS integration (detection of ampersand modules).

Updates and Issues fixed

<table>
<thead>
<tr>
<th>Key</th>
<th>Summary</th>
<th>Pr</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>BAM-1005</td>
<td>Setup fails</td>
<td></td>
<td>!Resolved</td>
</tr>
<tr>
<td>BAM-1017</td>
<td>Never Can Retrieve Changelogs</td>
<td></td>
<td>!Resolved</td>
</tr>
<tr>
<td>BAM-1063</td>
<td>Clover doesn't use the checkbox to determine if it should run.</td>
<td></td>
<td>!Resolved</td>
</tr>
<tr>
<td>BAM-1008</td>
<td>Breaks of display in &quot;All plans&quot; Dashboard when svn comments have html inside.</td>
<td></td>
<td>!Resolved</td>
</tr>
<tr>
<td>BAM-1000</td>
<td>Standalone Bamboo cannot start on Solaris</td>
<td></td>
<td>!Resolved</td>
</tr>
<tr>
<td>BAM-993</td>
<td>To provide easier configuration between Crowd and Bamboo the attached crowd-ehcache.xml file will need to be added to the bamboo release</td>
<td></td>
<td>!Resolved</td>
</tr>
<tr>
<td>BAM-986</td>
<td>Emails should be more intelligent</td>
<td></td>
<td>!Resolved</td>
</tr>
<tr>
<td>BAM-976</td>
<td>IM message recipients input accumulates square brackets on form load</td>
<td></td>
<td>!Resolved</td>
</tr>
<tr>
<td>BAM-964</td>
<td>&quot;FreeMarker template error!&quot; on &quot;Plan Summary&quot; page in JDK 1.4</td>
<td></td>
<td>!Resolved</td>
</tr>
<tr>
<td>BAM-884</td>
<td>Redirection error when browsing to Bamboo pages</td>
<td></td>
<td>!Resolved</td>
</tr>
<tr>
<td>BAM-882</td>
<td>Add an option to use SVN Externals</td>
<td></td>
<td>!Resolved</td>
</tr>
<tr>
<td>Ticket</td>
<td>Description</td>
<td>Status</td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>------------------------------------------------------------------------------</td>
<td>----------</td>
<td></td>
</tr>
<tr>
<td>BAM-862</td>
<td>Java error on startup</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-844</td>
<td>Commit changes do not trigger builds due to the use of SVN:externals</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-710</td>
<td>Internal server error: org.springframework.dao.DataIntegrityViolationException when deleting projects</td>
<td>Resolved</td>
<td></td>
</tr>
</tbody>
</table>
Bamboo 1.0.3 Upgrade Guide

This page last changed on Mar 27, 2007 by edwin@atlassian.com.

Upgrading from Bamboo 1.0.2 to 1.0.3

In this version, an upgrade task has been added to upgrade your CVS commit files data to a correct path (which includes module name). This may take a while to run, and it is strongly recommended that you back up your xml-data directory before proceeding. For fuller instructions please follow the Bamboo Upgrade Guide.

Upgrading from Bamboo 1.0.1 and earlier

In addition to the above, please read the Upgrade Guide for every version you are skipping during the upgrade. The complete list of Upgrade Guides is available here.
Bamboo 1.0.4 Release Notes

This page last changed on Apr 18, 2007 by edwin@atlassian.com.

⚠️ Bamboo 1.1 has now been released – see the Bamboo 1.1 Release Notes

Atlassian is proud to announce the release of Bamboo 1.0.4! Bamboo 1.0.4 is mainly a bug fix release with over 10 issues resolved.

In this release, the focus has been on resolving connectivity issues with Subversion and Perforce

Perforce Improvements

There have been a few changes in Bamboo's Perforce integration

- Bamboo will now cache the client root rather than polling the repository continuously to obtain it
  This reduces the load on the Perforce server considerably. However, if you change the root in the client definition on Perforce, Bamboo will require a restart to pick up the change
- Bamboo now uses changelist numbers to detect source code changes rather than a timestamp
  This will avoid all sorts of problems that occur when the Bamboo server clock and Perforce server clock are out of sync
- Bamboo now picks up multi line change descriptions from Perforce
- Bamboo can now generate web urls for perforce files when using Fisheye

Updates and Issues fixed

<table>
<thead>
<tr>
<th>Key</th>
<th>Summary</th>
<th>Pr</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>BAM-1056</td>
<td>Failed to get the build source code: svn: report aborted</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-974</td>
<td>Bamboo penetrated perforce server with repeated requests on plan creation</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-750</td>
<td>Perforce changes are not displayed when a manual build is executed</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1113</td>
<td>Perforce modifications not causing build</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1096</td>
<td>Change the way bamboo detects changes in perforce</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1085</td>
<td>Subversion code refresh failing to pick up new revisions</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1078</td>
<td>BuildChangeDetector</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>Ticket</td>
<td>Description</td>
<td>Resolution</td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>------------------------------------------------------------------------------</td>
<td>------------</td>
<td></td>
</tr>
<tr>
<td>BAM-1028</td>
<td>Bamboo throws exception when it polls Subversion repository</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-979</td>
<td>Different time zone on Perforce server does not work</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-890</td>
<td>SVN triggered update failing</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-747</td>
<td>Perforce repository polling build plan not building</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-871</td>
<td>Manual builds still poll the perforce server</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-829</td>
<td>Only first line of change description is displayed for Perforce changes</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-823</td>
<td>Web Repository URL is not persisted for Perforce repositories</td>
<td>Resolved</td>
<td></td>
</tr>
</tbody>
</table>
Bamboo 1.0.4 Upgrade Guide

This page last changed on Apr 15, 2007 by rosie@atlassian.com.

Upgrading from Bamboo 1.0.3 to 1.0.4

In this version, an upgrade task has been added to update Perforce plans to use the change list number rather than the timestamp when detecting changes. Please ensure that you have connectivity to the Perforce server before you upgrade.

If Bamboo encounters any errors during the upgrade task it will set the Perforce plan's last change list number to 0. This means that the next time you build that plan there may be some unusual results (eg. picking up every single change list). Once this build is complete normal behaviour will resume.

It is strongly recommended that you back up your xml-data directory before proceeding. For full instructions please follow the Bamboo Upgrade Guide.

Upgrading from Bamboo 1.0.2 and earlier

In addition to the above, please read the Upgrade Guide for every version you are skipping during the upgrade. The complete list of Upgrade Guides is available here.
**Bamboo 1.0.5 Release Notes**

This page last changed on May 07, 2007 by edwin@atlassian.com.

⚠️ Bamboo 1.1 has now been released – see the Bamboo 1.1 Release Notes

Atlassian is proud to announce the release of Bamboo 1.0.5! Bamboo 1.0.5 is mainly a bug fix release related to subversion connectivity issues.

### Updates and Issues fixed

<table>
<thead>
<tr>
<th>Key</th>
<th>Summary</th>
<th>Pr</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>BAM-1139</td>
<td>Locked externals in SVN causes infinite building loop</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1137</td>
<td>Authentication always fails for subversion repository</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1108</td>
<td>Removing last build queue blocks use</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1100</td>
<td>Cannot log into Bamboo</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1060</td>
<td>Bamboo source update problem: &quot;Failed to get the build source code&quot;</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1050</td>
<td>Null pointer when relogging in after session has died</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-911</td>
<td>Cannot authenticate with SVN repository</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1125</td>
<td>Project Creation Fails with Self Signed SSL Certificate for SVN</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1118</td>
<td>Fisheye link from Perforce project causes exception</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1022</td>
<td>Login link on comment page broken</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-891</td>
<td>Error when logging in to open source project</td>
<td></td>
<td>Resolved</td>
</tr>
</tbody>
</table>
Bamboo 1.0.5 Upgrade Guide

This page last changed on Apr 18, 2007 by edwin@atlassian.com.

Upgrading from Bamboo 1.0.4 to 1.0.5

It is strongly recommended that you back up your xml-data directory before proceeding. For full instructions please follow the Bamboo Upgrade Guide.

Upgrading from Bamboo 1.0.4 and earlier

In addition to the above, please read the Upgrade Guide for every version you are skipping during the upgrade. The complete list of Upgrade Guides is available here.
Bamboo 1.1 Release Notes

This page last changed on May 08, 2007 by mark@atlassian.com.

The Atlassian Bamboo team is proud to announce the release of Bamboo 1.1! This release contains a whole host of new features targetted to make your build plans even more powerful and flexible.

⚠️ Want to see Bamboo 1.1 in action? Check out our live opensource instance.

Advanced Notifications

In this release, we have extended Bamboo notifications framework to provide more flexibility, allowing you to select the how, who and when of notifications.

Notification Rules

Rather than having static fields for emails/IM recipients, Bamboo now allow you to define your own notifications for your build plans as a set of rules, giving you greater granularity in controlling exactly which recipient gets notified and when.

<table>
<thead>
<tr>
<th>Notification Trigger</th>
<th>Notification Recipients</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Failed Builds And First Successful</td>
<td>Users: test, admin</td>
<td>Edit</td>
</tr>
<tr>
<td>Notify After 5 Failed Builds</td>
<td>Roles: Watcher</td>
<td>Edit</td>
</tr>
<tr>
<td>All Completed Builds</td>
<td>Roles: Committer</td>
<td>Edit</td>
</tr>
</tbody>
</table>

Notification Triggers

In release 1.1, we introduce notification triggers, defining exactly when you would like a notification to be sent by Bamboo. By default, you can select a notification to be sent on "all builds completion", "after X failed builds" or "failed builds and first successful build". Want more customized triggers? You can now write your own as a notification condition plugin.

Add Build Notification

Notification Trigger:  All Completed Builds

Roles:  Failed Builds And First Successful
        After X Failed Builds
        Watcher - Users who have marked this build as favorite
        Committer

Notification Preferences

Different users prefer to get notified in different ways. Bamboo now lets you control that, via the new user notification preferences.
**Set Your Notification Preference**

How would you like bamboo to send you notifications:
- Do not send notifications
- Send instant message
- Send text email
- Send email and instant message

Please ensure that you have set the appropriate address for the notification type.

**Dynamic recipients**

Only want to receive a notification when you have committed against the build? Want to opt-in to receive notifications on the build plan that you are keeping an eye on? Bamboo 1.1 introduces two new dynamic recipient roles: committers (those users who have committed to the plan triggering the particular build to execute) and watchers (those users who have marked the build plan as their favourite), which allow you to do just that!

**Roles:**
- Committer - Users who have committed to the build
- Watcher - Users who have marked this build as favourite

Select the various roles you wish to receive notifications.

**Build Metadata**

Every build process is different, and each build will have its own information that you may want to keep track of and use on top of the information that Bamboo stores about your build. This is particularly the case if you run custom plugins in your build process.

**Pass them to your build**

One way to use your build metadata is to pass it along to your builder as a property or target. To do this, you simply specify your variables in your target (or goal) field in your builder configuration. During build execution, the variables will be substituted with the actual values from your build metadata.

**Goal:**

```
clean test -DbuildNumber=${(bamboo.buildNumber)}
```

The maven goal you want Bamboo to execute each time the source code changes. You can also define system properties such as `-Djava.awt.headless=true`.

**Global Variables**

Bamboo 1.1 also allows you the option to specify variables globally. When a build begins, the global variables will be populated to the build’s metadata. This is a handy option for you to control many plans in one go.
Global Variables

You can use this page to view, add and delete global variables. Global variables are available on every build run in Bamboo.

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>bambooVersion</td>
<td>1.1</td>
<td></td>
</tr>
</tbody>
</table>

View your metadata

Use the "Metadata" tab to keep track of all of your build's metadata.

<table>
<thead>
<tr>
<th>Metadata</th>
</tr>
</thead>
</table>

This build has the following metadata. These are property key value pairs describing the build. You can specify your own metadata in the build process via plugins.

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>buildKey</td>
<td>EXT-BLAH</td>
</tr>
<tr>
<td>buildNumber</td>
<td>31</td>
</tr>
<tr>
<td>bambooVersion</td>
<td>1.1</td>
</tr>
</tbody>
</table>

File Trigger Inclusions/Exclusions

In this release, we also introduce the file trigger inclusion/exclusion filter. Instead of listening and picking up all changes from a repository, you can now use regex patterns to define those files which you do (or don't) want to trigger builds.

Include / Exclude Files: Exclude all changes that matches to the following pattern

File Pattern: ".*documentation.*"

A regular expression to match the file to be included / excluded.

More pluggability

In release 1.1, we have added more plugin points to make Bamboo even more extensible than before. On top of the notification condition plugin point, we have also added pre-build action plugins, as well as repository plugins.

- Repository Plugins Not using SVN, CVS, or Perforce? You can now write a plugin to integrate with your very own source control.
- Pre-build Plugins Similar to the post-build action plugin, the pre-build action plugin will allow you to perform any custom task you may wish. The only difference is, of course, that it occurs before the build execution begins.
Improved Maven 2 error log parsing

Bamboo now intelligently parses the Maven 2 error log for possible errors in the build errors log, giving you a better view of what really went wrong in your build summary.

**Error Summary**

- The build generated some errors. See the [full build log](#) for details.

[INFO] --------------------------------------------------------------
[ERROR] BUILD FAILURE
[INFO] --------------------------------------------------------------
[INFO] Compilation failure

```
DefaultBuildQueue.java:[85,15] cannot find symbol
symbol :  method setDBQueue(com.atlassian.bamboo.buildqueue.BuildQueue)
location:  class com.atlassian.bamboo.builder.DefaultBuildExecutor
```

[INFO] --------------------------------------------------------------
[INFO] For more information, run Maven with the -v switch

LDAP and external user management support

In release 1.1, we have improved our user management capability to support externally sourced users and groups, including LDAP, and [Crowd](#)

Performance of Dashboard

With this release, we have also made significant performance improvements to the dashboard, which should see its load times reduce dramatically.

Other updates and bug fixes

On top of these features, we have also made a whole host of bug fixes, with over x bugs fixed since release 1.0.5.

<table>
<thead>
<tr>
<th>Key</th>
<th>Summary</th>
<th>Pr</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>BAM-940</td>
<td>Make Repository pluggable</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1189</td>
<td>Add global variables to be used in plans</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1183</td>
<td>Tool tip in the dashboard for the plans.</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1163</td>
<td>Dashboard very slow</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1134</td>
<td>&quot;operation not permitted&quot; clicking Completed Builds tab</td>
<td></td>
<td>Resolved</td>
</tr>
</tbody>
</table>

Atlassian JIRA (43 issues)

Document generated by Confluence on May 08, 2007 23:50
<table>
<thead>
<tr>
<th>Issue</th>
<th>Description</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>BAM-1117</td>
<td>Ant builder checks last 5 lines for BUILD SUCCESSFUL</td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1115</td>
<td>Artifact URL fails to escape invalid characters</td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1090</td>
<td>Allow pluggable pre-build actions</td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1088</td>
<td>Hide User Info From Non-Admins</td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1080</td>
<td>Lock obtain timed out error</td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1073</td>
<td>No way to pass proxy information down to Ant</td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1059</td>
<td>Implementation of new style of Notifications</td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1058</td>
<td>Error viewing Build: Expression failingSinceBuild.buildResultsSummary is undefined</td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1053</td>
<td>ANT_HOME is used for ant</td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1032</td>
<td>NullPointerExcepting accessing RSS feed</td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1029</td>
<td>LDAP integration</td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1020</td>
<td>Inconsistent Test Results between builds</td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1016</td>
<td>Allow include/exclude patterns for triggering build</td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1011</td>
<td>bamboo.home is not a valid environment variable</td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1010</td>
<td>Error summary should parse logs in Maven 2 builds</td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-996</td>
<td>FreeMarker template error in Completed Build Results screen</td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-952</td>
<td>Bamboo uses older m2 version even when maven 2.0.5 builder configured for the plan</td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-950</td>
<td>Out of memory error while processing Clover 2.0a4 results</td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-942</td>
<td>Upgrade jfreechart version due to concurrency bug</td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-938</td>
<td>Editing LDAP users in Bamboo</td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-934</td>
<td>OutOfMemory while checking a build result</td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-847</td>
<td>Remote API for manual checkout and build.</td>
<td>Resolved</td>
</tr>
<tr>
<td>Issue</td>
<td>Description</td>
<td>Status</td>
</tr>
<tr>
<td>-----------</td>
<td>-----------------------------------------------------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>BAM-819</td>
<td>'Important Files and Directories'</td>
<td>Closed</td>
</tr>
<tr>
<td>BAM-787</td>
<td>add screenshots (when Clover data is available)</td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-767</td>
<td>We able to add custom project specific Build Telemetry data</td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-729</td>
<td>make terminology consistent (&quot;build&quot; vs. &quot;plan&quot;)</td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-641</td>
<td>Ability to ignore file patterns for updates</td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-633</td>
<td>Add Recent Activity tab for the dashboard</td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-574</td>
<td>The interface used to provide a view of artifacts has issues dealing with subdirectories</td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-561</td>
<td>Configure dependencies in the opposite direction</td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-493</td>
<td>Better handling of large number of unit tests</td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-146</td>
<td>Build time graph should use minutes in scale</td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1173</td>
<td>Add an option to use a non Ajax dashboard &quot;hideDashboard&quot;</td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1124</td>
<td>Crowd 1.0.6 + Bamboo 1.0.4</td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-792</td>
<td>Pass in bamboo parameters such as project and plan key to builders</td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-604</td>
<td>API access to the SCM repository's build identifier for a given build</td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-681</td>
<td>Checkboxes for build queues</td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-680</td>
<td>Picker for e-mail notifications</td>
<td>Resolved</td>
</tr>
</tbody>
</table>

If you want to check out a live Bamboo instance, take a look at our [opensource instance](https://atlassian.com).
Bamboo 1.1 Upgrade Guide

This page last changed on May 07, 2007 by edwin@atlassian.com.

Upgrading from Bamboo 1.0.5 to 1.1

It is strongly recommended that you back up your xml-data directory before proceeding. For full instructions please follow the Bamboo Upgrade Guide.

Please note that the upgrade process may take a while to complete.

Upgrading from Bamboo 1.0.5 and earlier

In addition to the above, please read the Upgrade Guide for every version you are skipping during the upgrade. The complete list of Upgrade Guides is available here.
Bamboo Upgrade Guide

This page last changed on Mar 14, 2007 by edwin@atlassian.com.

⚠️ Before you begin
Please read the Release Notes and Upgrade Guides for the version you are upgrading to.

Step 1. Backing up

You need to back up three directories. Your Bamboo Home, Build Data Path and Configuration Path directories. If you do not remember what these are, you can simply go to the System Info page in the Administration menu of your Bamboo instance to check.

<table>
<thead>
<tr>
<th>Bamboo Paths</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuration Path: /opt/bamboo/xml-data/configuration</td>
</tr>
<tr>
<td>Build Path: /opt/bamboo/xml-data/projects</td>
</tr>
<tr>
<td>Bamboo Home: /opt/bamboo/bamboochrome</td>
</tr>
</tbody>
</table>

Step 2. Upgrading Bamboo

Windows Self Extracting installer:

Launching the installer Bamboo_windows_version.exe will install Bamboo. The install wizard requires the input of two directories:

1. Installation directory (BAMBOO_INSTALL) - the directory where the application files will be installed to.

⚠️ Make sure that you either specify a new directory to install to or delete the old Bamboo installation directory as legacy files may cause problems.

1. Bamboo Home directory (BAMBOO_HOME) - this is the directory where Bamboo will store all data for this installation. Please point this to the Bamboo Home directory as was seen in the Administration System Info screen in step 1.

Generic .tgz or .zip archive:

1. Unpack the archived standalone version to the directory of your choice. All unpacked files reside in a
Bamboo directory. Note: To unpack the .tgz distributable, run the command `tar xvf filename` (for .tgz distributable).

Make sure that you either unzip into a new directory or delete the old Bamboo directory as legacy files may cause problems.

1. Check your Bamboo Home directory - this is the directory where Bamboo will store all data for this installation. Make sure that the file named `bamboo-init.properties` in the `Bamboo\webapp\WEB-INF\classes` directory contains the `bamboo.home` line with with the absolute path to your Bamboo Home (`BAMBOO_HOME`) directory. If it does not exist, you will need to add it.

   ```
   bamboo.home=BAMBOO_HOME
   ```

When specifying the Bamboo home directory, please ensure that you enter the absolute path. On Windows systems, backslashes needs to be escaped. For example instead of `C:\dev\bamboo-home` use

   ```
   bamboo.home=C:\\dev\\bamboo-home
   ```

You may want to specify the Bamboo home directory separate from the installation directory. This will ensure that your data is not lost when upgrading or re-installing Bamboo.

Alternatively you may specify an environment variable 'bamboo.home' which specifies the absolute path to your Bamboo Home directory. Bamboo will check if an environment variable is defined.

Step 3. Start Bamboo

Once you have installed Bamboo and set the `bamboo.home` property, you can start Bamboo. The upgrade tasks will fire in the startup sequence of Bamboo. As a safeguard, you can monitor the `atlassian-bamboo.log` to ensure that the upgrade process has completed successfully.

Step 4. Re-Index Bamboo (as indicated in release notes)

Bamboo maintains an index of its results. This allow Bamboo to display aggregate build results information across builds. You may need to perform a re-index of Bamboo if the upgrade process requires it. This step may or may not be required (depending on the upgrade versions). To re-index, simply go to "Administration", then "Indexing", and hit the Reindex button. You only need to do this if you have existing data in Bamboo.

Depending on the number of builds and tests you may have, the indexing process may take a
significant amount of time. During this period, Bamboo will not be available. Also, it is advisable to ensure that all build queues are disabled and no builds are progressing when you start the re-indexing process. If you have a large instance, we suggest you reindex overnight.

Troubleshooting

If you have any problems during upgrade, please raise a support request at https://support.atlassian.com/ and attach your atlassian-bamboo.log so we can help you find out what's gone wrong.
Bamboo User's Guide

01. Getting Started
02. Working with Projects and Plans
03. Working with Build Results
04. Working with Tests
05. Reporting on Plan Trends
06. Reporting on Author Trends
07. Working with Comments
08. Working with Labels
09. Subscribing to RSS Feeds
10. Working with Instant Messenger (IM) Notifications
11. Editing your User Profile
Appendix A. Bamboo Glossary

Table of Contents

01. Getting Started

1.1 Using the Bamboo Dashboard
1.2 Viewing Bamboo's Current Activity
1.3 Viewing your Latest Build Results
1.4 Working with Favourites

Download

You can download the Bamboo User's Guide in PDF, HTML or XML formats.

About

Bamboo is a continuous integration (CI) server. Bamboo assists software development teams by providing:

- automated building and testing of software source-code.
- status updates on successful/failed builds.
- reporting tools for statistical analysis.

The Bamboo User's Guide provides information about using Bamboo. If you need information about installing Bamboo or configuring builds, please visit Bamboo Documentation Home.

If you have a question about using Bamboo that hasn't been answered here, please let us know.
02. Working with Projects and Plans

- 2.1 About Projects and Plans
- 2.2 Viewing a Plan's Details
- 2.3 Viewing a Plan's Activity Log

03. Working with Build Results

- 3.1 About Builds and Build Results
- 3.2 Viewing a Build Result
- 3.3 Viewing a Build Log
- 3.4 Viewing the Code Changes that triggered a Build
- 3.5 Viewing Clover Code-Coverage for a Build Result
- 3.6 Viewing Metadata for a Build Result
- 3.7 Viewing JIRA Issues for a Build Result

04. Working with Tests

- 4.1 Viewing Test Results for a Build
- 4.2 Viewing a Test's History
- 4.3 Viewing Test Statistics for a Plan

05. Reporting on Plan Trends

- 5.1 Viewing Build Statistics for a Plan
- 5.2 Generating Reports across multiple Plans
  - 'Build Activity per Plan' Report
  - 'Build Duration per Plan' Report
  - 'Clover Code Coverage per Plan' Report
  - 'Clover Lines of Code per Plan' Report
  - 'Number of Build Failures per Plan' Report
  - 'Number of Tests per Plan' Report
  - 'Percentage of Successful Builds per Plan' Report
  - 'Time to Fix per Plan' Report

06. Reporting on Author Trends

- 6.1 Viewing Build Statistics for all Authors
- 6.2 Viewing Build Results for an Author
- 6.3 Generating Reports on selected Authors
  - 'Build Activity per Author' Report
  - 'Number of Build Failures per Author' Report
  - 'Number of Builds Broken per Author' Report
  - 'Number of Builds Fixed per Author' Report
07. Working with Comments

- 7.1 About Comments
- 7.2 Commenting about a Build Result
- 7.3 Viewing Comments about a Build Result
- 7.4 Viewing Code Check-in Comments

08. Working with Labels

- 8.1 About Labels
- 8.2 Labelling a Build Result
- 8.3 Removing a Label from a Build Result
- 8.4 Viewing Labelled Build Results
- 8.5 Viewing Popular Labels

09. Subscribing to RSS Feeds

- 9.1 Subscribing to an RSS Feed for All Build Results for All Plans
- 9.2 Subscribing to an RSS Feed for Failed Builds for All Plans
- 9.3 Subscribing to an RSS Feed for All Build Results for a Particular Plan
- 9.4 Subscribing to an RSS Feed for Failed Builds for a Particular Plan
- 9.5 Subscribing to an RSS Feed for Labelled Build Results

10. Working with Instant Messenger (IM) Notifications

- 10.1 About Instant Messenger (IM) Notifications
- 10.2 Labelling a Build Result via IM
- 10.3 Commenting about a Build Result via IM

11. Editing your User Profile

- 11.1 Changing your Password
- 11.2 Changing your Notification Preferences
- 11.3 Associating your Author Name with your User Profile

Appendix A. Bamboo Glossary

- activity log
- artifact
- author
- build
- build activity
- build duration
- build log
- build plan
- build queue
- build result
- build telemetry
- builder
- favourites
- label
- plan
- project
- triggering
01. Getting Started

1. Getting Started

- 1.1 Using the Bamboo Dashboard
- 1.2 Viewing Bamboo's Current Activity
- 1.3 Viewing your Latest Build Results
- 1.4 Working with Favourites
  - 1.4.1 Adding a Plan to your Favourites
  - 1.4.2 Removing a Plan from your Favourites
1.1 Using the Bamboo Dashboard

The Dashboard is your Bamboo 'home' page. The Dashboard contains three tabs:

- 'All Plans' — a list of [build plans](#) and each plan's latest [build result](#).
- 'Current Activity' — Bamboo's [build queues](#), showing which plans Bamboo is currently building and which plans are waiting to be built.
- 'My Bamboo' — a convenient summary of information that is relevant to you:
  - plans which you have nominated as your [favourites](#).
  - [your latest build results](#) (i.e. builds that were triggered by your latest [code changes](#)).
  - a summary of your [build statistics](#).

Screenshot: Bamboo Dashboard-'All Plans' tab

Atlassian Bamboo

<table>
<thead>
<tr>
<th>Atlassian Bucket</th>
<th>Main Build &gt; BUCKET-MAIN-74</th>
<th>Ran: 1 week age</th>
<th>Updated by Joe Smith</th>
<th>Duration: 1 minute</th>
<th>Tests: 32 passed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Atlassian Config</td>
<td>Clover Build &gt; CONFIG-CLOVER</td>
<td>Ran: 1 week age</td>
<td>Manual build</td>
<td>Duration: 50 seconds</td>
<td>Tests: 51 passed</td>
</tr>
<tr>
<td>Atlassian Core</td>
<td>Main Build &gt; CORE MAIN1</td>
<td>Ran: 2 weeks age</td>
<td>Updated by Joe Smith</td>
<td>Duration: 1 minute</td>
<td>Tests: 51 passed</td>
</tr>
<tr>
<td>Atlassian Event</td>
<td>Main Build &gt; EVENT-MON-117</td>
<td>Ran: 1 month age</td>
<td>Manual build</td>
<td>Duration: 10 minutes</td>
<td>Tests: 110 passed</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Each blue name (e.g. 'Atlassian Bucket') represents a [project](#). The name(s) to the right of each project name are the [plan](#) (e.g. 'Main Build') belonging to that project, and the [build number](#) (e.g. 'BUCKET-MAIN-74') represents the latest build result for the plan, while the icon indicates the plan's current status:

- ![ ](image) This plan's latest build was successful.
- ![ ](image) This plan's latest build failed.
- ![ ](image) Bamboo is currently checking-out the source-code for this plan, in preparation for starting a build.
- ![ ](image) Bamboo is currently executing a build for this plan.
- ![ ](image) This plan has been disabled.

You can:

- click the plan name (e.g. 'Main Build') to [view the plan details](#).
- click the build number (e.g. 'BUCKET-MAIN-75') to [view the build result](#).
- click the author's name to [view the author's details](#) (the author is the person who triggered the build by checking-in code).

**Handy Hint**

You can return to the Dashboard from anywhere in Bamboo by clicking the 'Home' link in the top navigation bar.
only if you have logged in to Bamboo.
2 only if your Bamboo User Profile has been associated with your Author Name.

RELATED TOPICS

- 1.1 Using the Bamboo Dashboard
- 1.2 Viewing Bamboo's Current Activity
- 1.3 Viewing your Latest Build Results
- 1.4 Working with Favourites
  - 1.4.1 Adding a Plan to your Favourites
  - 1.4.2 Removing a Plan from your Favourites

Bamboo Glossary
Return to Bamboo Documentation Home
1.2 Viewing Bamboo's Current Activity

This page last changed on Feb 19, 2007 by rosie@atlassian.com.

Sometimes you may want to see which plans are currently being built, and which plans (if any) are waiting in a build queue.

A build queue controls the sequence of builds. Bamboo administrators can specify how many build queues there are in the system. For example, if there are two build queues, two builds can occur in parallel while subsequent builds will wait in a queue.

To view Bamboo's current activity,

1. Click the 'Home' link in the top navigation bar. This will display the Dashboard.
2. Click the 'Current Activity' tab. This will display Bamboo's Build Queues, as well as a list of Recently Completed Builds.

Screenshot: Bamboo Dashboard-'Current Activity' tab

The above screenshot shows a Bamboo system that has two Build Queues: 'Fast Builds' and 'Queue 2'. The 'Fast Builds' queue is currently building a plan called 'Crowd - Main Build'. You can:

- click a plan name (e.g. 'Crowd - Main Build') to view the plan details.
- click the icon to view the plan's continuous scrolling activity log.

Additionally, in the 'Recently Completed Builds' section, you can:

- click a build number (e.g. 'CWD-MAIN-189') to view the build result.
- click a 'Reason' (e.g. 'Updated by...') to view the code changes that triggered the build.
1.1 Using the Bamboo Dashboard
1.2 Viewing Bamboo's Current Activity
1.3 Viewing your Latest Build Results
1.4 Working with Favourites
   - 1.4.1 Adding a Plan to your Favourites
   - 1.4.2 Removing a Plan from your Favourites

Bamboo Glossary
Return to Bamboo Documentation Home
1.3 Viewing your Latest Build Results

This page last changed on Feb 19, 2007 by rosie@atlassian.com.

To view your latest build results,

1. Click the 'Home' link in the top navigation bar. This will display the Dashboard.
2. Click the 'My Bamboo' tab.
3. Your 10 latest build results (that is, builds that were triggered when you checked-in code) are listed in the 'My Latest Changes' section.

Handy Hint
Click any build number (e.g. 'BAM-MAIN-1846') to view the build result.

Screenshot: 'My Bamboo--My Latest Changes'

<table>
<thead>
<tr>
<th>Build</th>
<th>When</th>
<th>Comments</th>
<th>Test Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>BAM-MAIN-1845</td>
<td>2 days ago</td>
<td>Moved the powered by to the website</td>
<td>1308 passed</td>
</tr>
<tr>
<td>BAM-MAIN-1844</td>
<td>3 days ago</td>
<td>[maven-release-plugin] prepare release atlassian_bamboo_1_0_rc1</td>
<td>1308 passed</td>
</tr>
<tr>
<td>BAM-MAIN-1843</td>
<td>3 days ago</td>
<td>Updated so that by default JS toggle thing works</td>
<td>1308 passed</td>
</tr>
<tr>
<td>BAM-MAIN-1842</td>
<td>3 days ago</td>
<td>[maven-release-plugin] prepare for next development iteration</td>
<td>1308 passed</td>
</tr>
<tr>
<td>BAM-MAIN-1840</td>
<td>3 days ago</td>
<td>IE doesn't quite seem to like clearer &lt;br&gt; tags. Using &lt;div&gt; instead</td>
<td>1308 passed</td>
</tr>
<tr>
<td>BAM-MAIN-1839</td>
<td>3 days ago</td>
<td>Fixed footer for IE, some grammatically errors and the default favicon</td>
<td>1308 passed</td>
</tr>
<tr>
<td>BAM-MAIN-1838</td>
<td>3 days ago</td>
<td>Updated the stop in the showBuildQueue to redirect correctly</td>
<td>1308 passed</td>
</tr>
<tr>
<td>BAM-MAIN-1835</td>
<td>3 days ago</td>
<td>Updated the install4j version and the installer profile</td>
<td>1308 passed</td>
</tr>
<tr>
<td>BAM-MAIN-1834</td>
<td>3 days ago</td>
<td>Added an offline argument for the release builds</td>
<td>1308 passed</td>
</tr>
<tr>
<td>BAM-MAIN-1833</td>
<td>3 days ago</td>
<td>Restored the ability to restore the hide / fold thing through javascript</td>
<td>1308 passed</td>
</tr>
</tbody>
</table>

⚠️ If your Bamboo User Profile has not yet been associated with your Author Name, there will be no
My Latest Changes' section.

RELATED TOPICS

- 1.1 Using the Bamboo Dashboard
- 1.2 Viewing Bamboo's Current Activity
- 1.3 Viewing your Latest Build Results
- 1.4 Working with Favourites
  - 1.4.1 Adding a Plan to your Favourites
  - 1.4.2 Removing a Plan from your Favourites

Bamboo Glossary
Return to Bamboo Documentation Home
1.4 Working with Favourites

Whereas the 'All Plans' tab on the Bamboo Dashboard lists every plan that exists in your Bamboo system, the 'My Bamboo' tab lists just your chosen favourites — that is, the plans you work with the most. You can easily add and remove plans from your favourites.

To view your favourite plans,

1. Click the 'Home' link in the top navigation bar. This will display the Dashboard.
2. Click the 'My Bamboo' tab.
3. Your favourite plans are listed in the 'My Favourite Plans' section.

Screenshot: 'My Bamboo--My Favourite Plans'

![My Favourite Builds]

My Favourite Builds

<table>
<thead>
<tr>
<th>Bamboo</th>
<th>Collapse</th>
</tr>
</thead>
<tbody>
<tr>
<td>✔ HEAD &gt; BAM-MAIN-1848</td>
<td>![Star]</td>
</tr>
<tr>
<td>Ran: 2 days ago</td>
<td>Updated by Edwin Wong</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Test</th>
<th>Collapse</th>
</tr>
</thead>
<tbody>
<tr>
<td>✗ Crowd Copy &gt; TEST-CWD-15</td>
<td>![Star]</td>
</tr>
<tr>
<td>Ran: 3 days ago</td>
<td>Manual build</td>
</tr>
</tbody>
</table>

| ✗ JIRA Unit > TEST-JRA-24 | ![Star] |
| Ran: 11 hours ago | Scheduled build | Duration: 4 minutes | Tests: 14 out of 2630 failed! |

| ✔ Long 1 > TEST-LLLLL-24 | ![Star] |
| Ran: 11 hours ago | Scheduled build | Duration: 2 minutes | Tests: 1 passed |

⚠️ Depending on how your administrator has configured each plan's notifications, Bamboo may send you notifications about the build results for some or all of your favourite plans. You can choose whether you would like to receive your notifications by email and/or Instant Messaging (IM).

RELATED TOPICS

- 1.1 Using the Bamboo Dashboard
- 1.2 Viewing Bamboo's Current Activity
• 1.3 Viewing your Latest Build Results
• 1.4 Working with Favourites
  ° 1.4.1 Adding a Plan to your Favourites
  ° 1.4.2 Removing a Plan from your Favourites

Bamboo Glossary
Return to Bamboo Documentation Home
1.4.1 Adding a Plan to your Favourites

To add a plan to your favourites,

1. Click the 'Home' link in the top navigation bar. This will display the Dashboard.
2. Click the 'All Plans' tab.
3. This will display a list of all plans in your Bamboo system. (Note: Plans that have already been added to your favourites are indicated by a yellow star icon. Plans that have not been added to your favourites are indicated by a grey star icon.)
4. Locate the plan and click the grey star icon: ⭐
5. Click the 'My Bamboo' tab.
6. Verify that the plan is now listed in the 'My Favourite Plans' section.

✔ Handy Hint
   If your administrator has enabled 'Auto-Favourites', a plan will be automatically added to your favourites the first time you check-in code for that plan.

RELATED TOPICS

- 1.1 Using the Bamboo Dashboard
- 1.2 Viewing Bamboo's Current Activity
- 1.3 Viewing your Latest Build Results
- 1.4 Working with Favourites
  - 1.4.1 Adding a Plan to your Favourites
  - 1.4.2 Removing a Plan from your Favourites

Bamboo Glossary
Return to Bamboo Documentation Home
1.4.2 Removing a Plan from your Favourites

To remove a plan from your favourites,

1. Click the 'Home' link in the top navigation bar. This will display the Dashboard.
2. Click the 'All Plans' tab.
3. This will display a list of all plans in your Bamboo system. (Note: Plans that have been added to your favourites are indicated by a yellow star icon. Plans that have not been added to your favourites are indicated by a grey star icon.)
4. Locate the plan and click the yellow star icon: 🔹
5. Click the 'My Bamboo' tab.
6. Verify that the plan is not listed in the 'My Favourite Plans' section.

RELATED TOPICS

- 1.1 Using the Bamboo Dashboard
- 1.2 Viewing Bamboo's Current Activity
- 1.3 Viewing your Latest Build Results
- 1.4 Working with Favourites
  - 1.4.1 Adding a Plan to your Favourites
  - 1.4.2 Removing a Plan from your Favourites

Bamboo Glossary
Return to Bamboo Documentation Home
2. Working with Projects and Plans

- 2.1 About Projects and Plans
- 2.2 Viewing a Plan's Details
- 2.3 Viewing a Plan's Activity Log
2.1 About Projects and Plans

A Bamboo plan (or build plan) is the "recipe" for a build.

A plan defines: what gets built (i.e. the source-code repository); how the build is triggered; which builder to use; what tests to run; what artifacts the build will produce; who will be notified of the build result; and any labels with which the build result or build artifacts will be tagged.

Every plan belongs to a project.

A project enables easy identification of plans that are logically related to each other, which is useful for instance when generating reports across multiple plans. Each project has a Name (e.g. "CRM System") and a Key (e.g. "CRM"). The Project Key is prefixed to the relevant Plan Keys, e.g. the "CRM" project could have plans "CRM-TRUNK" and "CRM-BRANCH". Every Bamboo plan is listed on the Dashboard, from where you can:

- Click on a Plan Name to view the plan details
- Click on a Build Number to view the plan's latest build result

Projects and plans can only be configured by Bamboo administrators. Please see the Bamboo Administrator's Guide for details.

RELATED TOPICS

- 2.1 About Projects and Plans
- 2.2 Viewing a Plan's Details
- 2.3 Viewing a Plan's Activity Log

Bamboo Glossary
Return to Bamboo Documentation Home
2.2 Viewing a Plan's Details

To view a plan's details,

- From the Dashboard, locate and click a Plan Name from the list;
- OR:
- From within a build result, click the Plan Name at the top left of the screen.

The Plan Summary will be displayed as follows:

Screenshot: 'Plan Summary'
The green box indicates that this plan's latest build was successful. Note that a red box in this position would indicate that the plan's latest build failed, while a blue box would indicate that a build is currently in progress.

- Click the build number (i.e. 'BUCKET-MAIN-74') to view the build result.
- Click the 'Updated by' link to view the code changes that triggered the latest build result.
- {___}Build duration_ is the total time taken to
execute a **build plan** --- that is, the time taken to compile the code and run all of the plan's tests.

The large '%' box indicates the success rate of this plan's recent builds. This percentage is calculated on the last 25 builds, or as per your selection via the blue down-arrow:

- Showing Last 25 builds

Click the blue down-arrow to choose how you would like the percentage and graphs on this screen to be calculated. Choose from the following:

- this plan's last 25 builds.
- this plan's builds in the last 7 days.
- this plan's builds in the last 30 days.
- this plan's builds in the last 90 days.
- all of this plan's builds. The percentage and graphs on this screen will all be recalculated automatically when you choose a different option.

The 'Summary' tab provides a quick snapshot of the current status of the plan. For more details:

- Click the 'Activity' tab to view the plan's current activity.
- Click the 'Completed Builds' tab to view a list of **build results** for this plan's recent builds (i.e. the last 25 builds, or as per your selection via the blue down-arrow).
- Click the 'Tests' tab to view a summary of the **test results** for this plan's recent builds (i.e. the last 25 builds, or as per your selection via the blue down-arrow).
- Click the 'Files' tab to view a list of all the files currently contained in this plan's source-code repository.

Tips: Plans can only be configured by an administrator. For details please see the Bamboo Administrator's Guide.

### RELATED TOPICS

- [2.1 About Projects and Plans](#)
• 2.2 Viewing a Plan's Details
• 2.3 Viewing a Plan's Activity Log

Bamboo Glossary
Return to Bamboo Documentation Home
2.3 Viewing a Plan's Activity Log

Every plan has an activity log. An activity log is a temporary display of the latest output from the plan's most recent build log.

To view a plan's activity log,

1. From the Dashboard, locate and click a plan name from the list;
   OR:
   From within a build result, click the plan name at the top left of the screen.
2. Click the 'Activity Log' tab.

The plan's current activity will be displayed. For example, the following screenshot shows a plan for which a build is currently in progress:

Screenshot: Plan Activity
### Confluence - Unit Tests: Live Activity Logs

**Build Number:** 1370 (currently building)

**Started:** 02/17/2007 02:09 PM

**Completed:** 62%

**Time remaining:** Approximately 4 minutes remaining

---

### Changes

This build was the result of the following change:

- **Joe Smith**
  - Added info to support editable comments.

---

### Currently Building

<table>
<thead>
<tr>
<th>Time</th>
<th>Activity Log Entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>02-Feb-2007 15:11:21</td>
<td>de jains.akemochi.TestSuite.run(TestSuite.java:683)</td>
</tr>
<tr>
<td>02-Feb-2007 15:11:21</td>
<td>de org.apache.maven.war.test.UnitBattery.execute(UnitBattery.java:242)</td>
</tr>
<tr>
<td>02-Feb-2007 15:11:21</td>
<td>de org.apache.maven.war.test.UnitBattery.execute(UnitBattery.java:216)</td>
</tr>
<tr>
<td>02-Feb-2007 15:11:21</td>
<td>de org.apache.maven.war.test.UnitBattery.execute(UnitBattery.java:215)</td>
</tr>
<tr>
<td>02-Feb-2007 15:11:21</td>
<td>de org.apache.maven.war.test.UnitSuite.run(UnitSuite.java:163)</td>
</tr>
<tr>
<td>02-Feb-2007 15:11:21</td>
<td>de org.apache.maven.war.test.UnitSuite.run(UnitSuite.java:87)</td>
</tr>
</tbody>
</table>

---

**You can also monitor a plan’s build activity over time by using the "Build Activity per Plan" report. Build activity is the number of builds that occur in a given period of time.**

---

**RELATED TOPICS**

- 2.1 About Projects and Plans
- 2.2 Viewing a Plan’s Details
- 2.3 Viewing a Plan’s Activity Log

**Bamboo Glossary**

Return to Bamboo Documentation Home
3. Working with Build Results

- 3.1 About Builds and Build Results
- 3.2 Viewing a Build Result
- 3.3 Viewing a Build Log
- 3.4 Viewing the Code Changes that triggered a Build
- 3.5 Viewing Clover Code-Coverage for a Build Result
- 3.6 Viewing Metadata for a Build Result
- 3.7 Viewing JIRA Issues for a Build Result
3.1 About Builds and Build Results

This page last changed on May 07, 2007 by rosie@atlassian.com.

A build is one execution of a plan.

Every build has a Build Number, which is appended to the relevant Plan Key to form the Build Key. For example, if a plan with the key "CRM-BRANCH" is executed for the seventeenth time, the build key will be "CRM-BRANCH-17".

Every completed build has a build result:

- 'Successful' — the code compiled, with or without errors, and all tests completed successfully.
- 'Failed' — either the code did not compile, or at least one test failed.

Additionally,

- if the build result is 'Failed', and the previous build result was 'Successful', the build is said to be 'Broken'.
- if the build result is 'Successful', and the previous build result was 'Failed', the build is said to be 'Fixed'.

The latest build result for every plan is listed on the Dashboard. Bamboo can also send notifications about build results.

RELATED TOPICS

- 3.1 About Builds and Build Results
- 3.2 Viewing a Build Result
- 3.3 Viewing a Build Log
- 3.4 Viewing the Code Changes that triggered a Build
- 3.5 Viewing Clover Code-Coverage for a Build Result
- 3.6 Viewing Metadata for a Build Result
- 3.7 Viewing JIRA Issues for a Build Result

Bamboo Glossary
Return to Bamboo Documentation Home
3.2 Viewing a Build Result

This page last changed on May 07, 2007 by ernest@atlassian.com.

To view a plan's most recent build result:

1. Go to the Dashboard.
2. Locate the plan in the list, then click the Build Number.

To view all build results for a plan:

1. Go to the relevant plan.
2. Click the 'Completed Builds' tab, then click the Build Number in the list.

A build result looks like this:

Screenshot: 'Build Result Summary'

Project: Bonnie Plan: Main Build Build: 21

Build Result BONNIE-MAIN-21

Build BONNIE-MAIN-21 was successful.
Has been successful since BONNIE-MAIN-2 (7 months before).
Build completed on 11/01/2007 12:10:23 - 3 months ago
Build took 22 seconds

Tests
75 tests in total

Build Errors

The build generated some errors. See the build log for details.

Error Summary

Note: Some input files were or override a deprecated API.
Note: Reconsider using -Xlint:deprecation for details.

In the above screenshot:

You can click the plan name ('Main Build' in this example) to see the plan details for this build.

You can use the build results navigator to scroll through other build results for this plan.

The 'Summary' tab shows a snapshot of the build result. To see more detail:
Click the 'Tests' tab to view the build's test results.
Click the 'Changes' tab to view the code changes that triggered this build (if applicable).
Click the 'Logs' tab to view a complete build log.
Click the 'Comments' tab to view a trail of comments regarding this build result. You can also click the following icon to add a comment:

Click the 'Metadata' tab to view any metadata that relates to this build result (if applicable).
Click the 'JIRA' tab to view any JIRA issues that relate to this build result (if applicable).

**RELATED TOPICS**

- 3.1 About Builds and Build Results
- 3.2 Viewing a Build Result
- 3.3 Viewing a Build Log
- 3.4 Viewing the Code Changes that triggered a Build
- 3.5 Viewing Clover Code-Coverage for a Build Result
- 3.6 Viewing Metadata for a Build Result
- 3.7 Viewing JIRA Issues for a Build Result

Bamboo Glossary
Return to Bamboo Documentation Home

Return to Bamboo Documentation Home
3.3 Viewing a Build Log

Every build has a build log. A build log is a permanent record of all the output generated by compiling the plan's source-code and executing the tests.

To view a build log,

1. Go to the build result.
2. Click the 'Logs' tab.

Screenshot: 'Build Log'
Build Log

BucketMain generated the following output:

22-Jan-2007 18:20:06 Executing build 74

27-Jan-2007 12:00:00 Running command line: /opt/java/boolee/xml-data/build-git/BUCKET-MAIN/bucket
... using java: /usr/java/jre/...
... using environment variables:

27-Jan-2007 12:00:00 build-status:

27-Jan-2007 12:00:00 clean:

27-Jan-2007 12:00:00 [delete] Deleting directory /opt/bamboe/xml-data/build-git/BUCKET-MAIN/bucket/target

27-Jan-2007 12:00:00 clean:

27-Jan-2007 12:00:00 javasparse filesystem:

27-Jan-2007 12:00:00 [mkdir] Created dir: /opt/bamboe/xml-data/build-git/BUCKET-MAIN/bucket/target/classes

27-Jan-2007 12:00:00 javac compile:

27-Jan-2007 12:00:00 [echo] Compiling to /opt/bamboe/xml-data/build-git/BUCKET-MAIN/bucket/target/classes

27-Jan-2007 12:00:00 [javac] Compiling 21 source files to /opt/bamboe/xml-data/build-git/BUCKET-MAIN/bucket/target/classes

27-Jan-2007 12:00:00 Note: Some input files use or override a deprecated API. Note: Recompile with -Xlint:deprecation for details.

27-Jan-2007 12:00:00 javasrc resources:

27-Jan-2007 12:00:00 Removing 4 files to /opt/bamboe/xml-data/build-git/BUCKET-MAIN/bucket/target/classes

27-Jan-2007 12:00:00 test:prepare file system:

27-Jan-2007 12:00:00 [mkdir] Created dir: /opt/bamboe/xml-data/build-git/BUCKET-MAIN/bucket/target/test

27-Jan-2007 12:00:00 [mkdir] Created dir: /opt/bamboe/xml-data/build-git/BUCKET-MAIN/bucket/target/test/classes

27-Jan-2007 12:00:00 [javac] Compiling 23 source files to /opt/bamboe/xml-data/build-git/BUCKET-MAIN/bucket/target/test/classes

27-Jan-2007 12:00:00 test:src

27-Jan-2007 12:00:00 test:compile:

27-Jan-2007 12:00:00 [javac] Compiling 15 source files to /opt/bamboe/xml-data/build-git/BUCKET-MAIN/

Related Topics:

- 3.1 About Builds and Build Results
- 3.2 Viewing a Build Result
- 3.3 Viewing a Build Log
- 3.4 Viewing the Code Changes that triggered a Build
- 3.5 Viewing Clover Code-Coverage for a Build Result
- 3.6 Viewing Metadata for a Build Result
- 3.7 Viewing JIRA Issues for a Build Result
3.4 Viewing the Code Changes that triggered a Build

If a build was triggered by a code change, the code changes will be shown in the build result.

To view the code changes that triggered a particular build result,

1. Go to the build result.
2. Click the 'Changes' tab.
3. A list of updated files will be shown. Click the filename to view the changes; or, click the version number to view the entire file; or, click the 'diffs' links to view the differences between the current and previous version of each file.

Screenshot: 'Code Changes'

Build Result BUCKET-MAIN-69

This build was the result of the following change:

Joe Smith

- /build/local/proj/1.170/Build/BUCKET/Main/BUCKET-MAIN-69
- /build/local/proj/1.170/Build/BUCKET/Main/BUCKET-MAIN-69

(Note that links to individual source-code files will only be available if your Bamboo administrator has specified a 'Web Repository URL' in the build's plan. For details please see the Bamboo Administrator's Guide.)

A note about build triggering

There are a variety of ways in which a build can be triggered for a plan:

- Code updated — a build can be triggered whenever one or more authors checks-in code.
- Scheduled build — a build can be scheduled to occur at regular intervals.
- Dependency — a build can be triggered whenever a successful build occurs for another plan.
- Manual build — a build can be triggered manually.
- Initial clean build — a build will be triggered when a new plan is created.

The way in which each build was triggered is listed in the 'Reason' column on the Dashboard.
3.1 About Builds and Build Results
3.2 Viewing a Build Result
3.3 Viewing a Build Log
3.4 Viewing the Code Changes that triggered a Build
3.5 Viewing Clover Code-Coverage for a Build Result
3.6 Viewing Metadata for a Build Result
3.7 Viewing JIRA Issues for a Build Result

Bamboo Glossary
Return to Bamboo Documentation Home
3.5 Viewing Clover Code-Coverage for a Build Result

If your organisation uses the Cenqua Clover code-coverage tool, Bamboo can record code-coverage details (e.g. the percentage of code covered by tests) for each build result.

This is only available if the build's plan specifies a Clover directory (for details please refer to the Bamboo Administrator's Guide). Also note that the Clover analysis will only be recorded for successful builds.

To view Clover code-coverage for a build result:

1. Go to the plan.
2. Click the 'Completed Builds' tab, then click the Build Number in the list.
3. This will display the Build Result Summary. Click the 'Clover' tab.

Screenshot: Clover Code-Coverage for a Build Result

Build Result CONFIG-CLOVER-8

Code Coverage

<table>
<thead>
<tr>
<th>Coverage Type</th>
<th>Coverage Percentage</th>
<th>Build Statistics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Build Coverage</td>
<td>35.6% (19%)</td>
<td>LOC: 5675</td>
</tr>
</tbody>
</table>
| | | The lines of code in the build that is covered by tests.
| Statement Coverage | 32.2% | NLOC: 2547 |
| | | The non-comment lines of code in the build.
| Methods Coverage | 96.4% | Files: 40 |
| Conditionals Coverage | 50.0% | Methods: 584 |

Bamboo also provides data on code-coverage trends for a plan over a period of time. For details see:

- 'Clover Code Coverage per Plan' Report.

RELATED TOPICS

- 3.1 About Builds and Build Results
• 3.2 Viewing a Build Result
• 3.3 Viewing a Build Log
• 3.4 Viewing the Code Changes that triggered a Build
• 3.5 Viewing Clover Code-Coverage for a Build Result
• 3.6 Viewing Metadata for a Build Result
• 3.7 Viewing JIRA Issues for a Build Result

Bamboo Glossary
Return to Bamboo Documentation Home
3.6 Viewing Metadata for a Build Result

This page last changed on May 07, 2007 by rosie@atlassian.com.

If your source-code repository provides metadata for your build results, Bamboo will display it.

To view the metadata for a build result:

1. Go to the plan.
2. Click the 'Completed Builds' tab, then click the Build Number in the list.
3. This will display the Build Result Summary. Click the 'Metadata' tab.

**Screenshot: Metadata for a Build Result**

Build Result CORE-MAIN-55

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>buildKey</td>
<td>CORE-MAIN</td>
</tr>
<tr>
<td>buildNumber</td>
<td>55</td>
</tr>
<tr>
<td>CUSTOM EXPANSION NUMBER</td>
<td>30175</td>
</tr>
</tbody>
</table>

**RELATED TOPICS**

- 3.1 About Builds and Build Results
- 3.2 Viewing a Build Result
- 3.3 Viewing a Build Log
- 3.4 Viewing the Code Changes that triggered a Build
- 3.5 Viewing Clover Code-Coverage for a Build Result
- 3.6 Viewing Metadata for a Build Result
- 3.7 Viewing JIRA Issues for a Build Result

Bamboo Glossary

Return to Bamboo Documentation Home
3.7 Viewing JIRA Issues for a Build Result

This page last changed on May 07, 2007 by ernest@atlassian.com.

If your organisation uses the JIRA issue-tracker, Bamboo can automatically create links to any issues mentioned in code check-in comments, as well as displaying a summary in the build result (see below). This provides an easy way to jump to relevant issue(s) to see details about what the code is intended to achieve.

Links to JIRA issues are indicated by a small green arrow to the right of the issue key.

This will only be available if Bamboo-JIRA communication has been enabled by your Bamboo administrator. For details please refer to the Bamboo Administrator’s Guide.

To view the JIRA issues for a build result:

1. Go to the plan.
2. Click the 'Completed Builds' tab, then click the Build Number in the list.
3. This will display the Build Result Summary. Click the 'JIRA' tab.

Screenshot: JIRA Issues for a Build Result

<table>
<thead>
<tr>
<th>Project: atlassian-renderer</th>
<th>Plan: Main Build</th>
<th>Build: 57</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Build Result RENDERER-MAIN-57</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Labels:</strong> NONE Add</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**JIRA Issues**

atlassian-renderer - Main Build 57 is related to the following JIRA issues:

<table>
<thead>
<tr>
<th>Key</th>
<th>Summary</th>
<th>Updated</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>JIRA-11966</td>
<td>Thumbnails in xml renderer do not link to the full sized image</td>
<td>Jan 17, 2007 10:50:07 AM</td>
<td>Show related builds</td>
</tr>
</tbody>
</table>

To view all build results related to a JIRA issue:

1. Click the 'Show related builds' link in the above screenshot.

RELATED TOPICS

- 3.1 About Builds and Build Results
- 3.2 Viewing a Build Result
- 3.3 Viewing a Build Log
- 3.4 Viewing the Code Changes that triggered a Build
- 3.5 Viewing Clover Code-Coverage for a Build Result
- 3.6 Viewing Metadata for a Build Result
• 3.7 Viewing JIRA Issues for a Build Result

Bamboo Glossary
Return to Bamboo Documentation Home
4. Working with Tests

- 4.1 Viewing Test Results for a Build
- 4.2 Viewing a Test's History
- 4.3 Viewing Test Statistics for a Plan
4.1 Viewing Test Results for a Build

Bamboo provides a convenient summary of all the tests that were run when a particular build was executed — as well as full details of any errors. This is useful when you are investigating what caused a build to fail.

To view the tests for a particular build:

1. Go to the **build result**.
2. Click the 'Tests' tab.
3. Any failed tests will be listed first. Scroll down to see a list of successful tests.

**Screenshot 1: Failed Tests for a Build**

**Failed Tests**

Build 3 has the following 10 errors. There were no new test failures since the previous build.

<table>
<thead>
<tr>
<th>Test Name (Existing failed tests)</th>
<th>Failing Since</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>com.atlassian.jira.service.util.handler.TestCVSLogHandler: Warning:</td>
<td></td>
<td>less than 1 second</td>
</tr>
<tr>
<td>Exception in constructor: com.atlassian.jira.service.util.handler.TestCVSLogHandler</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Screenshot 2: Successful Tests for a Build**
Successful Tests

The following 2006 tests have passed. Tests took a total of 2 minutes. Tests are alphabetically sorted. You can also view them sorted by duration.

<table>
<thead>
<tr>
<th>Test Name</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>com.atlassian.jira.action.TestJIRAActionSupport</td>
<td>&lt;1 sec</td>
</tr>
<tr>
<td>Get action name</td>
<td>&lt;1 sec</td>
</tr>
<tr>
<td>Get result</td>
<td>&lt;1 sec</td>
</tr>
<tr>
<td>Add error messages</td>
<td>&lt;1 sec</td>
</tr>
<tr>
<td>Add error messages from a result</td>
<td>&lt;1 sec</td>
</tr>
<tr>
<td>Get delegator</td>
<td>&lt;1 sec</td>
</tr>
<tr>
<td>Remove key or add event</td>
<td>&lt;1 sec</td>
</tr>
<tr>
<td>com.atlassian.jira.action.admin.TestDataExport</td>
<td>&lt;1 sec</td>
</tr>
<tr>
<td>Data export respects the html tag flag</td>
<td>&lt;1 sec</td>
</tr>
<tr>
<td>Data export saves xml tag string</td>
<td>&lt;1 sec</td>
</tr>
<tr>
<td>com.atlassian.jira.action.admin.TestDataReport</td>
<td>&lt;1 sec</td>
</tr>
<tr>
<td>Excludes</td>
<td>1 sec</td>
</tr>
<tr>
<td>com.atlassian.jira.action.admin.TestIndexOptimize</td>
<td>&lt;1 sec</td>
</tr>
<tr>
<td>Null index doesn't work</td>
<td>&lt;1 sec</td>
</tr>
<tr>
<td>Getters and setters</td>
<td>&lt;1 sec</td>
</tr>
<tr>
<td>Validation failure if indexing disabled</td>
<td>&lt;1 sec</td>
</tr>
<tr>
<td>Error if indexing throws up</td>
<td>&lt;1 sec</td>
</tr>
<tr>
<td>com.atlassian.jira.action.admin.TestIndexOptimize</td>
<td>&lt;1 sec</td>
</tr>
</tbody>
</table>

To see a particular test's results for other builds, click the test name.

RELATED TOPICS

- 4.1 Viewing Test Results for a Build
- 4.2 Viewing a Test's History
- 4.3 Viewing Test Statistics for a Plan

Bamboo Glossary

Return to Bamboo Documentation Home
4.2 Viewing a Test's History

This page last changed on Feb 19, 2007 by rosie@atlassian.com.

A test's history shows you:

- The occasions when the test has failed. This can be useful when investigating what code changes were related to a failed test (see below).
- The test's average duration (running time), and whether the duration is increasing or decreasing across builds.

To view a test's history,

1. Go to a plan or a build result.
2. Click the 'Tests' tab.
3. Click the name of the test in which you are interested.
4. (Skip this step if you are looking at a plan.) The test's latest result will be displayed. Click the link 'View test case across builds'.

Build Result RENDERER-MAIN-57
Label: NONE

<table>
<thead>
<tr>
<th>Summary</th>
<th>Tests</th>
<th>Changes</th>
<th>Logs</th>
<th>Comments</th>
<th>JIRA</th>
</tr>
</thead>
</table>

Latest Status: RENDERER-MAIN-57 was successful

Internal anchor

The below summarizes the result of the unit test 'Internal anchor' in build 57 of atlassian-renderer - Main Build

- Description: Internal anchor
- Test Class: com.atlassian.renderer.TestSimple
- Method: testInternalAnchor
- Duration: 0.132 seconds
- Status: Successful

5. The 'Test History' will be displayed as shown below.

Screenshot: Test History
To view the code changes that relate to a failed test,

1. Under 'Recent Failures', click the relevant build result ('47' in the above screenshot).
2. This will display the build result. Click the 'Changes' tab to display the code changes.

**RELATED TOPICS**

- [4.1 Viewing Test Results for a Build](#)
- [4.2 Viewing a Test's History](#)
- [4.3 Viewing Test Statistics for a Plan](#)

**Bamboo Glossary**
Return to [Bamboo Documentation Home](#)
4.3 Viewing Test Statistics for a Plan

Bamboo provides a summary of test results across all of a plan's builds. This helps you to:

- Troubleshoot by identifying which tests fail most frequently, and which tests take longest to fix.
- Manage your build duration by identifying the plan's slowest running tests.
- Ensure quality by monitoring the number of tests over time: are your test cases growing with your code base?

To view the test statistics for all of a plan's builds:

1. Go to the **plan**.
2. Click the 'Tests' tab.
3. The plan's 'Top 10 Most Failing Tests' sub-tab will be displayed. Click the other three sub-tabs to view the plan's 'Top 10 Longest to Fix Tests', 'Top 10 Longest Running Tests', 'Number of Tests' (see screenshots below).

**Screenshot 1: Top 10 Most Failing Tests**

<table>
<thead>
<tr>
<th>Test</th>
<th>Times Failed</th>
<th>Most Recent</th>
</tr>
</thead>
<tbody>
<tr>
<td>F a g</td>
<td>2</td>
<td>37, 36</td>
</tr>
<tr>
<td>Image cone cone on: Image4</td>
<td>1</td>
<td>47</td>
</tr>
<tr>
<td>Link: Link17</td>
<td>1</td>
<td>47</td>
</tr>
<tr>
<td>Escape: escape16</td>
<td>0</td>
<td>47</td>
</tr>
<tr>
<td>Uri render cannot case off: uri9</td>
<td>0</td>
<td>47</td>
</tr>
<tr>
<td>Uri render cannot case on: uri9</td>
<td>0</td>
<td>47</td>
</tr>
<tr>
<td>Link: Link6</td>
<td>0</td>
<td>47</td>
</tr>
<tr>
<td>Tables: table14</td>
<td>0</td>
<td>47</td>
</tr>
</tbody>
</table>

To view a test's **history**, click the test name.

**Screenshot 2: Top 10 Longest to Fix Tests**
### Test 10 Most Falling Tests

<table>
<thead>
<tr>
<th>Test</th>
<th>Average Time to Fix</th>
</tr>
</thead>
<tbody>
<tr>
<td>Faq</td>
<td>1 hour, 2 minutes</td>
</tr>
<tr>
<td>Image camel case on: image4</td>
<td>13 minutes</td>
</tr>
<tr>
<td>Link: link17</td>
<td>13 minutes</td>
</tr>
<tr>
<td>Escape: escape16</td>
<td>10 minutes</td>
</tr>
<tr>
<td>Uri render camel case off: url1</td>
<td>13 minutes</td>
</tr>
<tr>
<td>Uri render camel case on: url1</td>
<td>13 minutes</td>
</tr>
<tr>
<td>Link: link6</td>
<td>13 minutes</td>
</tr>
<tr>
<td>Uri render camel case off: url1</td>
<td>19 minutes</td>
</tr>
<tr>
<td>Tables: table1</td>
<td>19 minutes</td>
</tr>
<tr>
<td>Link: link6</td>
<td>19 minutes</td>
</tr>
</tbody>
</table>

### Screenshot 3: Top 10 Longest Running Tests

<table>
<thead>
<tr>
<th>Test</th>
<th>Average Duration (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Faq</td>
<td>7.40 (seconds)</td>
</tr>
<tr>
<td>Jira release plan</td>
<td>1.53 (seconds)</td>
</tr>
<tr>
<td>Layout</td>
<td>0.69 (less than 1 second)</td>
</tr>
<tr>
<td>Image after macro</td>
<td>0.52 (less than 1 second)</td>
</tr>
<tr>
<td>Image camel case on: default</td>
<td>0.48 (less than 1 second)</td>
</tr>
<tr>
<td>Image camel case off: default</td>
<td>0.44 (less than 1 second)</td>
</tr>
<tr>
<td>Lists: list21</td>
<td>0.49 (less than 1 second)</td>
</tr>
<tr>
<td>Newly inserted link after table</td>
<td>0.42 (less than 1 second)</td>
</tr>
<tr>
<td>Emotions need to styled text</td>
<td>0.41 (less than 1 second)</td>
</tr>
<tr>
<td>Insert renders as u</td>
<td>0.41 (less than 1 second)</td>
</tr>
</tbody>
</table>

*These times are based on the last 25 builds only.

### Screenshot 4: Number of Tests
The chart below shows the number of tests across time. This provides a rough indication of the level of testing over time for the build.

**RELATED TOPICS**

- 4.1 Viewing Test Results for a Build
- 4.2 Viewing a Test's History
- 4.3 Viewing Test Statistics for a Plan

**Bamboo Glossary**

Return to Bamboo Documentation Home
05. Reporting on Plan Trends

5. Reporting on Trends

- 5.1 Viewing Build Statistics for a Plan
- 5.2 Generating Reports across multiple Plans
  - 'Build Activity per Plan' Report
  - 'Build Duration per Plan' Report
  - 'Clover Code Coverage per Plan' Report
  - 'Clover Lines of Code per Plan' Report
  - 'Number of Build Failures per Plan' Report
  - 'Number of Tests per Plan' Report
  - 'Percentage of Successful Builds per Plan' Report
  - 'Time to Fix per Plan' Report
5.1 Viewing Build Statistics for a Plan

To view a plan's build statistics,

- From the Dashboard, locate and click a Plan Name from the list;
  OR:
- From within a build result, click the Plan Name at the top left of the screen.

The Plan Summary will be displayed as follows:

Screenshot: 'Plan Summary'
Atlassian Bucket - Main Build: Plan Summary

Latest build BUCKET-MAIN-74 was successful
Reasons: Updated by Travis  Completed 2 days ago  Duration: 30 seconds

96%
Successful Runs: 24/25
Average Duration: 39 seconds

Build Timing & Number of Failures per Build

% Successful Builds & Avg Duration per Time Period

Note: Charts are drawn from the results of the last 25 builds. Use the drop down in the right corner to change your filter size.

Feed for all builds or just the failed builds.

Recent Failures

- Average time to fix a failure: 42 minutes
- Average number of builds between tests: 1 builds
- The longest time taken to fix a failure is 42 minutes, from failure starting in build 73.
- The greatest number of builds taken to fix a failure is 1, from failure starting in build 73.

<table>
<thead>
<tr>
<th>Failed Builds</th>
<th>Fixed In</th>
<th>Time Taken to Fix</th>
</tr>
</thead>
<tbody>
<tr>
<td>73 (updated by Travis)</td>
<td>74 (updated by Travis)</td>
<td>1 builds 42 minutes</td>
</tr>
</tbody>
</table>

In the above screenshot:

The green box indicates that this plan's latest build was successful. Note that a red box in this position would indicate that the plan's latest build failed, while a blue box would indicate that a build is currently in progress.

- Click the build number (i.e. 'BUCKET-MAIN-71') to view the build result.
- Click the 'Updated by' link to view the code changes that triggered the latest build result.
- Build duration is the total time taken to...
execute a **build plan** — that is, the time taken to compile the code and run all of the plan's tests.

| ![96% Success](image) | Successful Runs: 24/25  
|-----------------------|-------------------------  
| Average Duration: 30 seconds |

The large '%' box indicates the success rate of this plan's recent builds. This percentage is calculated on the last 25 builds, or as per your selection via the blue down-arrow:

- Showing Last 25 builds

Click the blue down-arrow to choose how you would like the percentage and graphs on this screen to be calculated. Choose from the following:

- this plan's last 25 builds.
- this plan's builds in the last 7 days.
- this plan's builds in the last 30 days.
- this plan's builds in the last 90 days.
- all of this plan's builds. The percentage and graphs on this screen will all be recalculated automatically when you choose a different option.

The 'Summary' tab provides a quick snapshot of the current status of the plan. For more details:

- Click the 'Activity' tab to view the plan's *current activity*.
- Click the 'Completed Builds' tab to view a list of *build results* for this plan's recent builds (i.e. the last 25 builds, or as per your selection via the blue down-arrow).
- Click the 'Tests' tab to view a summary of the *test results* for this plan's recent builds (i.e. the last 25 builds, or as per your selection via the blue down-arrow).
- Click the 'Files' tab to view a list of all the files currently contained in this plan's source-code repository.

### RELATED TOPICS

- **5.1 Viewing Build Statistics for a Plan**
- **5.2 Generating Reports across multiple Plans**
  - 'Build Activity per Plan' Report
  - 'Build Duration per Plan' Report
  - 'Clover Code Coverage per Plan' Report
  - 'Clover Lines of Code per Plan' Report
  - 'Number of Build Failures per Plan' Report
  - 'Number of Tests per Plan' Report
  - 'Percentage of Successful Builds per Plan' Report
° 'Time to Fix per Plan' Report

Bamboo Glossary
Return to Bamboo Documentation Home
5.2 Generating Reports across multiple Plans

Bamboo provides a report generator that enables you to compare build statistics across one or more plans, using a variety of different metrics. To report on build statistics per plan,

1. Click the 'Reports' link in the top navigation bar. This will display the 'Report Parameters' screen as shown below.
2. 'Report' — choose from the available reports. Available reports include:
   - 'Build Activity per Plan' Report
   - 'Build Duration per Plan' Report
   - 'Clover Code Coverage per Plan' Report
   - 'Clover Lines of Code per Plan' Report
   - 'Number of Build Failures per Plan' Report
   - 'Number of Tests per Plan' Report
   - 'Percentage of Successful Builds per Plan' Report
   - 'Time to Fix per Plan' Report
   Additionally, your Bamboo administrator may configure custom reports by using plugins. For details please see the Bamboo Administrator's Guide.
3. 'Build plans' — choose the plan(s) on which you want to report. You can use the <Ctrl> key to select multiple plans.
   📌 Project names are shown in italics, e.g. 'Geronimo SVN'. Plan names are shown in non-italics, e.g. 'Main Build'.
4. 'Group By' — choose whether your report's horizontal axis should show days, months or weeks. You can also specify 'Auto', which varies by report, but will generally default to 'week'.
5. 'Date Filter' — choose from:
   - 'All builds'
   - 'Last 7 days'
   - 'Last 30 days'
   - 'Last 90 days'
   - 'Select Range' — choosing this option will display two boxes in which you will need to specify the 'from' and 'to' dates (dd/MM/yyyy).
6. Click the 'Submit' button to generate your report.
## Report Parameters

<table>
<thead>
<tr>
<th>Report:</th>
<th>Select...</th>
</tr>
</thead>
</table>
| Build Plans: | - Geronimo SVN Man Build  
- Jetty Man Build  
- Python SVN Man Build  
- Spring  
- JDK 1.6  
- Struts 2 SVN Man Build  
- Sun HotSpot JVM Man Build  
- Sun Java Compiler  
- Webwork  
- XWork  
- XWork-1.2.Branched  |
| Group By: | Auto |
| Date Filter: | All Builds |

### RELATED TOPICS

- 5.1 Viewing Build Statistics for a Plan
- 5.2 Generating Reports across multiple Plans  
  - 'Build Activity per Plan' Report  
  - 'Build Duration per Plan' Report  
  - 'Clover Code Coverage per Plan' Report  
  - 'Clover Lines of Code per Plan' Report  
  - 'Number of Build Failures per Plan' Report  
  - 'Number of Tests per Plan' Report  
  - 'Percentage of Successful Builds per Plan' Report  
  - 'Time to Fix per Plan' Report

Bamboo Glossary
Return to Bamboo Documentation Home
'Build Activity per Plan' Report

Build activity is the number of builds that occur in a given period of time.

You can choose the plan(s) and time period on which you want to report.

Sample Report: 'Build Activity per Plan'

**Build Activity**

How many builds are triggered in a given time period? This indicates the level of activity for the build.

![Build Activity Chart](chart.png)

**RELATED TOPICS**

- [5.1 Viewing Build Statistics for a Plan](#)
- [5.2 Generating Reports across multiple Plans](#)
  - 'Build Activity per Plan' Report
  - 'Build Duration per Plan' Report
  - 'Clover Code Coverage per Plan' Report
  - 'Clover Lines of Code per Plan' Report
  - 'Number of Build Failures per Plan' Report
  - 'Number of Tests per Plan' Report
  - 'Percentage of Successful Builds per Plan' Report
  - 'Time to Fix per Plan' Report

**Bamboo Glossary**
'Build Duration per Plan' Report

Build duration is the total time taken to execute a build plan — that is, the time taken to compile the code and run all of the plan's tests.

You can choose the plan(s) and time period on which you want to report.

Sample Report: 'Build Duration per Plan'

Build Duration

The report shows how long your build takes over time. Is it getting slower or faster?

RELATED TOPICS

- 5.1 Viewing Build Statistics for a Plan
- 5.2 Generating Reports across multiple Plans
  - 'Build Activity per Plan' Report
  - 'Build Duration per Plan' Report
  - 'Clover Code Coverage per Plan' Report
  - 'Clover Lines of Code per Plan' Report
  - 'Number of Build Failures per Plan' Report
  - 'Number of Tests per Plan' Report
  - 'Percentage of Successful Builds per Plan' Report
  - 'Time to Fix per Plan' Report

Bamboo Glossary
'Clover Code Coverage per Plan' Report

This report will only be available if your administrator has specified 'Clover output will be produced' in the plan's configuration. For details please see the Bamboo Administrator's Guide.

You can choose the plan(s) and time period on which you want to report.

Sample Report: 'Clover Code Coverage per Plan'

Clover Code Coverage

Comparing code coverage gives you an idea of how well the code base is tested. 100% coverage means that all code elements have been covered by your tests.

RELATED TOPICS

- 5.1 Viewing Build Statistics for a Plan
- 5.2 Generating Reports across multiple Plans
  - 'Build Activity per Plan' Report
  - 'Build Duration per Plan' Report
  - 'Clover Code Coverage per Plan' Report
  - 'Clover Lines of Code per Plan' Report
  - 'Number of Build Failures per Plan' Report
  - 'Number of Tests per Plan' Report
  - 'Percentage of Successful Builds per Plan' Report
  - 'Time to Fix per Plan' Report
'Clover Lines of Code per Plan' Report

This page last changed on Apr 15, 2007 by rosie@atlassian.com.

This report will only be available if your administrator has specified 'Clover output will be produced' in the plan's configuration. For details please see the Bamboo Administrator's Guide.

You can choose the plan(s) and time period on which you want to report.

Sample Report: 'Clover Lines of Code per Plan'

Clover Lines of Code

Provides an indication of the size of the code base for the build.

![Graph showing lines of code over time]

RELATTED TOPICS

- 5.1 Viewing Build Statistics for a Plan
- 5.2 Generating Reports across multiple Plans
  - 'Build Activity per Plan' Report
  - 'Build Duration per Plan' Report
  - 'Clover Code Coverage per Plan' Report
  - 'Clover Lines of Code per Plan' Report
  - 'Number of Build Failures per Plan' Report
  - 'Number of Tests per Plan' Report
  - 'Percentage of Successful Builds per Plan' Report
  - 'Time to Fix per Plan' Report
'Number of Build Failures per Plan' Report

You can choose the plan(s) and time period on which you want to report.

Sample Report: 'Number of Build Failures per Plan'

Number of Build Failures

How many builds are being broken? A high value indicates a relatively unstable build that tends to be broken often.

RELATED TOPICS

- 5.1 Viewing Build Statistics for a Plan
- 5.2 Generating Reports across multiple Plans
  - 'Build Activity per Plan' Report
  - 'Build Duration per Plan' Report
  - 'Clover Code Coverage per Plan' Report
  - 'Clover Lines of Code per Plan' Report
  - 'Number of Build Failures per Plan' Report
  - 'Number of Tests per Plan' Report
  - 'Percentage of Successful Builds per Plan' Report
  - 'Time to Fix per Plan' Report

Bamboo Glossary
Return to Bamboo Documentation Home
'Number of Tests per Plan' Report

You can choose the plan(s) and time period on which you want to report.

Sample Report: 'Number of Tests per Plan'

**Number of Tests**

How many tests does your build have? This provides a rough indication of the level of testing overtime for the build.

![Graph showing number of tests over time]

### RELATED TOPICS

- 5.1 Viewing Build Statistics for a Plan
- 5.2 Generating Reports across multiple Plans
  - 'Build Activity per Plan' Report
  - 'Build Duration per Plan' Report
  - 'Clover Code Coverage per Plan' Report
  - 'Clover Lines of Code per Plan' Report
  - 'Number of Build Failures per Plan' Report
  - 'Number of Tests per Plan' Report
  - 'Percentage of Successful Builds per Plan' Report
* Time to Fix per Plan* Report

**Bamboo Glossary**

Return to [Bamboo Documentation Home](#)
'Percentage of Successful Builds per Plan' Report

You can choose the plan(s) and time period on which you want to report.

Sample Report: 'Percentage of Successful Builds per Plan'

Percentage of Successful Builds

Comparing success percentages gives you an idea of how stable a build is compared to one another. 100% means your build is always rock solid. 0% means something is seriously wrong.

--

Chart  Data Table  Builds

<table>
<thead>
<tr>
<th>%Build Successful</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
</tr>
<tr>
<td>90</td>
</tr>
<tr>
<td>80</td>
</tr>
<tr>
<td>70</td>
</tr>
<tr>
<td>60</td>
</tr>
<tr>
<td>50</td>
</tr>
<tr>
<td>40</td>
</tr>
<tr>
<td>30</td>
</tr>
<tr>
<td>20</td>
</tr>
<tr>
<td>10</td>
</tr>
<tr>
<td>0</td>
</tr>
</tbody>
</table>

14-Dec  21-Dec  28-Dec  4-Jan

- Geronimo SVN - Main Build
- XWork - Main Build

--

RELATED TOPICS

- 5.1 Viewing Build Statistics for a Plan
- 5.2 Generating Reports across multiple Plans
  - 'Build Activity per Plan' Report
  - 'Build Duration per Plan' Report
  - 'Clover Code Coverage per Plan' Report
  - 'Clover Lines of Code per Plan' Report
  - 'Number of Build Failures per Plan' Report
  - 'Number of Tests per Plan' Report
  - 'Percentage of Successful Builds per Plan' Report
  - 'Time to Fix per Plan' Report

Bamboo Glossary


'Time to Fix per Plan' Report

This page last changed on Feb 19, 2007 by rosie@atlassian.com.

You can choose the plan(s) and time period on which you want to report.

Sample Report: 'Time to Fix per Plan'

Time to Fix

How long does it take on average to fix problems? This provides an indication of how quickly breakages are resolved for the plan.

![Time to Fix Chart]

<table>
<thead>
<tr>
<th>Chart</th>
<th>Data Table</th>
<th>Builds</th>
</tr>
</thead>
</table>

320,000
300,000
280,000
260,000
240,000
220,000
200,000
180,000
160,000
140,000
120,000
100,000
80,000
60,000
40,000
20,000
0

Dec-2006 Jan-2007

RELATED TOPICS

- 5.1 Viewing Build Statistics for a Plan
- 5.2 Generating Reports across multiple Plans
  - 'Build Activity per Plan' Report
  - 'Build Duration per Plan' Report
  - 'Clover Code Coverage per Plan' Report
  - 'Clover Lines of Code per Plan' Report
  - 'Number of Build Failures per Plan' Report
  - 'Number of Tests per Plan' Report
  - 'Percentage of Successful Builds per Plan' Report
  - 'Time to Fix per Plan' Report

Document generated by Confluence on May 08, 2007 23:50 Page 298
06. Reporting on Author Trends

This page last changed on Feb 19, 2007 by rosie@atlassian.com.

6. Reporting on Author Trends

- 6.1 Viewing Build Statistics for all Authors
- 6.2 Viewing Build Results for an Author
- 6.3 Generating Reports on selected Authors
  - 'Build Activity per Author' Report
  - 'Number of Build Failures per Author' Report
  - 'Number of Builds Broken per Author' Report
  - 'Number of Builds Fixed per Author' Report
  - 'Percentage of Successful Builds per Author' Report
6.1 Viewing Build Statistics for all Authors

An author is any person who checks-in code to a repository that is associated with a Bamboo plan. An author need not be a Bamboo user. To view a summary of all authors' statistics,

1. Click the 'Authors' link in the top navigation bar.
2. This will display the following screen, where you can click any column-header to sort in ascending order (or click twice to sort in descending order).

Screenshot: 'Authors Summary—sorted by 'Fixed' (descending)'

<table>
<thead>
<tr>
<th>Name</th>
<th>Triggered</th>
<th>Failed</th>
<th>% Failed</th>
<th>Broken</th>
<th>Fixed</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Joe Bloggs</td>
<td>932</td>
<td>275</td>
<td>29%</td>
<td>124</td>
<td>126</td>
<td>5</td>
</tr>
<tr>
<td>Mary Smith</td>
<td>454</td>
<td>06</td>
<td>21%</td>
<td>47</td>
<td>46</td>
<td>.3</td>
</tr>
<tr>
<td>Tom Brown</td>
<td>431</td>
<td>89</td>
<td>22%</td>
<td>50</td>
<td>40</td>
<td>.10</td>
</tr>
<tr>
<td>Sally Jones</td>
<td>492</td>
<td>102</td>
<td>24%</td>
<td>41</td>
<td>32</td>
<td>.9</td>
</tr>
</tbody>
</table>

Handy Hint
You can click any author's name to see their recent build results.

RELATED TOPICS

- 6.1 Viewing Build Statistics for all Authors
- 6.2 Viewing Build Results for an Author
- 6.3 Generating Reports on selected Authors
  - 'Build Activity per Author' Report
  - 'Number of Build Failures per Author' Report
  - 'Number of Builds Broken per Author' Report
  - 'Number of Builds Fixed per Author' Report
  - 'Percentage of Successful Builds per Author' Report

Bamboo Glossary
Return to Bamboo Documentation Home
6.2 Viewing Build Results for an Author

This page last changed on Feb 19, 2007 by rosie@atlassian.com.

An author is any person who checks-in code to a repository that is associated with a Bamboo plan. An author need not be a Bamboo user.
To view an author's build results,

1. Click the 'Authors' link in the top navigation bar.
2. This will display the 'Authors Summary' screen. Click the relevant author's name.
3. This will display the author's 'User Details' (email address, etc). Click through the following tabs to view recent build results:
   - 'Builds Summary' — a statistical summary of all the author's builds.
   - 'Last 10 Builds' — a list of the last 10 builds that were triggered by this author.
   - 'Last 10 Broken' — a list of the last 10 builds that were triggered by this author, where the build failed and the previous build for the same plan was successful.
   - 'Last 10 Fixed' — a list of the last 10 builds that were triggered by this author, where the build was successful and the previous build for the same plan failed.

Screenshot 1: 'User Details' tab

<table>
<thead>
<tr>
<th>User Details</th>
<th>Builds Summary</th>
<th>Last 10 Builds</th>
<th>Last 10 Broken</th>
<th>Last 10 Fixed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Username:</td>
<td>mark</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Full Name:</td>
<td>Mark Chaimungkaloneni</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Email:</td>
<td><a href="mailto:mark@atlassian.com">mark@atlassian.com</a></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Jobber Address:</td>
<td><a href="mailto:wanderingmulu@atlassian.com">wanderingmulu@atlassian.com</a></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Groups:</td>
<td>bamboo-admin, bamboo-user, bamboo-developers</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Source Repository Alias:</td>
<td>moshai</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Screenshot 2: 'Builds Summary' tab
Builds triggered by mohai

- Builds triggered by an author are those builds which contains changes committed by the author.
- All builds triggered: 993
- Failed Builds: 223 (22%)
- Successful Builds: 770 (72%)

Diagnostics and Fixes

- Broken means the build has failed but the previous build was successful. Fixed means that the build was successful but the previous build has failed.
- Broken By Author: 123 (13% of all builds triggered)
- Fixed By Author: 127 (13% of all builds triggered)
- Difference of fixes and breaks: 4

Screenshot 3: 'Last 10 Builds' tab

<table>
<thead>
<tr>
<th>Build</th>
<th>When</th>
<th>Comments</th>
<th>Test Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>DAM-MAIN-1763</td>
<td>9 hours ago</td>
<td>Moved bamboo-wiki.properties sample to the right location (<a href="https://example.com">DAM-7268</a>)</td>
<td>[details] 1313 passed</td>
</tr>
<tr>
<td>DAM-MAIN-1762</td>
<td>1 day ago</td>
<td>Fixed up stupid points on the cancel link on edit antiant <a href="https://example.com">DAM-7269</a></td>
<td>[details] 1313 passed</td>
</tr>
<tr>
<td>DAM-MAIN-1761</td>
<td>2 days ago</td>
<td>Added some WIP stuff for dashboard.</td>
<td>[details] 1213 passed</td>
</tr>
<tr>
<td>DAM-MAIN-1760</td>
<td>2 days ago</td>
<td>Fixed up wording on errors exceeded, not succeeded</td>
<td>[details] 1313 passed</td>
</tr>
<tr>
<td>DAM-MAIN-1759</td>
<td>5 days ago</td>
<td>Added YUI div wrappers... Might be worth checking out?</td>
<td>[details] 1211 passed</td>
</tr>
<tr>
<td>DAM-MAIN-1758</td>
<td>5 days ago</td>
<td>Updated the default values for the logs</td>
<td>[details] 1311 passed</td>
</tr>
<tr>
<td>DAM-MAIN-1757</td>
<td>5 days ago</td>
<td>Added axiom candy for latest build widget and the logs page</td>
<td>[details] 1211 passed</td>
</tr>
<tr>
<td>DAM-MAIN-1756</td>
<td>3 days ago</td>
<td>Fixed</td>
<td>[details] 1211 passed</td>
</tr>
<tr>
<td>DAM-MAIN-1755</td>
<td>5 days ago</td>
<td>[maven-release-plugin] prepare for next development iteration</td>
<td>[details] 1311 passed</td>
</tr>
<tr>
<td>DAM-MAIN-1754</td>
<td>5 days ago</td>
<td>[maven-release-plugin] prepare release atlassian_bamboo_1.0_beta</td>
<td>[details] 1211 passed</td>
</tr>
</tbody>
</table>

Screenshot 4: 'Last 10 Broken' tab
<table>
<thead>
<tr>
<th>Build</th>
<th>When</th>
<th>Comments</th>
<th>Test Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>BAM-MAIN-1731</td>
<td>1 week ago</td>
<td>Added some better help text for (BAM-6448)</td>
<td>[details] 720 passed</td>
</tr>
<tr>
<td>BAM-MAIN-1724</td>
<td>1 week ago</td>
<td>CVS library update test commit</td>
<td>[details] 720 passed</td>
</tr>
<tr>
<td>BAM-MAIN-1710</td>
<td>2 weeks ago</td>
<td>Made sure that we didn’t clear the form values when we check in (BAM-6988)</td>
<td>[details] 1210 passed</td>
</tr>
<tr>
<td>BAM-MAIN-1698</td>
<td>2 weeks ago</td>
<td>Make sure that .rel files builds work on windows (BAM-6649)</td>
<td>[details] 1 out of 1310 failed</td>
</tr>
<tr>
<td>BAM-MAIN-1588</td>
<td>2 weeks ago</td>
<td>Add bamboo-plugin-autofavourite plugin to the default project structure. Can needs be used as a template for other plugin projects</td>
<td>[details] 1310 passed</td>
</tr>
<tr>
<td>BAM-MAIN-1775</td>
<td>2 weeks ago</td>
<td>Make sure we encrypt repository passwords (BAM-1520)</td>
<td>[details] 4 out of 1311 failed</td>
</tr>
<tr>
<td>BAM-MAIN-1583</td>
<td>1 month ago</td>
<td>Added pagination for build result pages (BAM-1524)</td>
<td>[details] 3 out of 1307 failed</td>
</tr>
<tr>
<td>BAM-MAIN-1522</td>
<td>1 month ago</td>
<td>Simplified downloads quite a bit</td>
<td>[details] 2 out of 1285 failed</td>
</tr>
<tr>
<td>BAM-MAIN-1490</td>
<td>1 month ago</td>
<td>You can now edit plans from all build pages</td>
<td>[details] 1285 passed</td>
</tr>
<tr>
<td>BAM-MAIN-1498</td>
<td>1 month ago</td>
<td>Added hooks to make the build results navigator work for test cases</td>
<td>[details] 2 out of 1285 failed</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Removed unused submenu</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Fixed some crunks and added new owners for authors</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Same UI break for the authors page</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Fixed some repository strategy names</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Fixed the YUI tabs to load properly on IE especially you don’t get the big tabs</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Removed some unused code</td>
<td></td>
</tr>
</tbody>
</table>

**Screenshot 5: 'Last 10 Fixed' tab**

<table>
<thead>
<tr>
<th>Build</th>
<th>When</th>
<th>Comments</th>
<th>Test Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>BAM-MAIN-1700</td>
<td>2 weeks ago</td>
<td>Fixed rogue NullPointerException (BAM-5448)</td>
<td>[details] 1210 passed</td>
</tr>
<tr>
<td>BAM-MAIN-1698</td>
<td>2 weeks ago</td>
<td>You’ve gotta add the files</td>
<td>[details] 1510 passed</td>
</tr>
<tr>
<td>BAM-MAIN-1692</td>
<td>2 weeks ago</td>
<td>Fixed some failing tests (BAM-6170)</td>
<td>[details] 1210 passed</td>
</tr>
<tr>
<td>BAM-MAIN-1631</td>
<td>1 month ago</td>
<td>Fixed maven test (BAM-5540)</td>
<td>[details] 1307 passed</td>
</tr>
<tr>
<td>BAM-MAIN-1631</td>
<td>1 month ago</td>
<td>Minor formatting stuff for maven builds (BAM-5534)</td>
<td>[details] 1307 passed</td>
</tr>
<tr>
<td>BAM-MAIN-1697</td>
<td>1 month ago</td>
<td>Fixed pagination test (BAM-2140)</td>
<td>[details] 1307 passed</td>
</tr>
<tr>
<td>BAM-MAIN-1699</td>
<td>1 month ago</td>
<td>Changed some namings of builds to plans</td>
<td>[details] 1285 passed</td>
</tr>
<tr>
<td>BAM-MAIN-1699</td>
<td>1 month ago</td>
<td>Changed some namings of builds to plans</td>
<td>[details] 1285 passed</td>
</tr>
<tr>
<td>BAM-MAIN-1699</td>
<td>1 month ago</td>
<td>quirks to not load when SetFavoriteAction is invoked (BAM-5528)</td>
<td>[details] 1285 passed</td>
</tr>
<tr>
<td>BAM-MAIN-1698</td>
<td>1 month ago</td>
<td>Fixed some tests</td>
<td>[details] 1206 passed</td>
</tr>
<tr>
<td>BAM-MAIN-1697</td>
<td>1 month ago</td>
<td>Updated the build navigator to have a decent tooltip (BAM-5289)</td>
<td>[details] 1282 passed</td>
</tr>
<tr>
<td>BAM-MAIN-1697</td>
<td>1 month ago</td>
<td>Removed unused tests</td>
<td>[details] 1202 passed</td>
</tr>
<tr>
<td>BAM-MAIN-1697</td>
<td>1 month ago</td>
<td>test commit</td>
<td>[details] 1205 passed</td>
</tr>
</tbody>
</table>

⚠️ If your Bamboo User Profile has not yet been associated with your Author Name, there will be no 'User Details' tab.
6.1 Viewing Build Statistics for all Authors
6.2 Viewing Build Results for an Author
6.3 Generating Reports on selected Authors
   - 'Build Activity per Author' Report
   - 'Number of Build Failures per Author' Report
   - 'Number of Builds Broken per Author' Report
   - 'Number of Builds Fixed per Author' Report
   - 'Percentage of Successful Builds per Author' Report

Bamboo Glossary
Return to Bamboo Documentation Home
6.3 Generating Reports on selected Authors

This page last changed on Feb 19, 2007 by rosie@atlassian.com.

An author is any person who checks-in code to a repository that is associated with a Bamboo plan. An author need not be a Bamboo user.

To generate a report on selected authors,

1. Click the 'Authors' link in the top navigation bar.
2. Click the 'Statistics' tab. This will display the 'Report Parameters' screen as shown below.
3. 'Report' — choose from the available reports. Available reports include:
   • 'Build Activity per Author' Report
   • 'Number of Build Failures per Author' Report
   • 'Number of Builds Broken per Author' Report
   • 'Number of Builds Fixed per Author' Report
   • 'Percentage of Successful Builds per Author' Report

   Additionally, your Bamboo administrator may configure custom reports by using plugins. For details please see the Bamboo Administrator’s Guide.

4. 'Authors' — choose the author(s) on whom you want to report. You can use the <Ctrl>key to select multiple author.
5. 'Group By' — choose whether your report's horizontal axis should show days, months or weeks. You can also specify 'Auto', which varies by report, but will generally default to 'month'.
6. Click the 'Submit' button to generate your report.

Screenshot: 'Report Parameters--Authors'
Related Topics

- 6.1 Viewing Build Statistics for all Authors
- 6.2 Viewing Build Results for an Author
- 6.3 Generating Reports on selected Authors
  - 'Build Activity per Author' Report
  - 'Number of Build Failures per Author' Report
  - 'Number of Builds Broken per Author' Report
  - 'Number of Builds Fixed per Author' Report
  - 'Percentage of Successful Builds per Author' Report

Bamboo Glossary
Return to Bamboo Documentation Home
'Build Activity per Author' Report

This page last changed on Feb 19, 2007 by rosie@atlassian.com.

Build activity is the number of builds that occur in a given period of time. You can select the author(s) on whom you want to report.

Sample Report: 'Build Activity per Author'

```
build activity
```

 RELATED TOPICS

- 6.1 Viewing Build Statistics for all Authors
- 6.2 Viewing Build Results for an Author
- 6.3 Generating Reports on selected Authors
  - 'Build Activity per Author' Report
  - 'Number of Build Failures per Author' Report
  - 'Number of Builds Broken per Author' Report
  - 'Number of Builds Fixed per Author' Report
  - 'Percentage of Successful Builds per Author' Report

Bamboo Glossary
Return to Bamboo Documentation Home
'Number of Build Failures per Author' Report

You can select the author(s) on whom you want to report.

Sample Report: 'Number of Build Failures per Author'

**Number of Build Failures**

![Graph showing number of build failures per author over time]

**RELATED TOPICS**

- 6.1 Viewing Build Statistics for all Authors
- 6.2 Viewing Build Results for an Author
- 6.3 Generating Reports on selected Authors
  - 'Build Activity per Author' Report
  - 'Number of Build Failures per Author' Report
  - 'Number of Builds Broken per Author' Report
  - 'Number of Builds Fixed per Author' Report
  - 'Percentage of Successful Builds per Author' Report

Bamboo Glossary
Return to Bamboo Documentation Home
'Number of Builds Broken per Author' Report

You can select the author(s) on whom you want to report.

Sample Report: 'Number of Builds Broken per Author'

### Number of Builds Broken

![Number of Builds Broken Chart]

#### RELATED TOPICS

- 6.1 Viewing Build Statistics for all Authors
- 6.2 Viewing Build Results for an Author
- 6.3 Generating Reports on selected Authors
  - 'Build Activity per Author' Report
  - 'Number of Build Failures per Author' Report
  - 'Number of Builds Broken per Author' Report
  - 'Number of Builds Fixed per Author' Report
  - 'Percentage of Successful Builds per Author' Report

Bamboo Glossary
Return to Bamboo Documentation Home
'Number of Builds Fixed per Author' Report

You can select the author(s) on whom you want to report.

Sample Report: 'Number of Builds Fixed per Author'

![Chart: Number of Builds Fixed](image)

**RELATED TOPICS**

- 6.1 Viewing Build Statistics for all Authors
- 6.2 Viewing Build Results for an Author
- 6.3 Generating Reports on selected Authors
  - 'Build Activity per Author' Report
  - 'Number of Build Failures per Author' Report
  - 'Number of Builds Broken per Author' Report
  - 'Number of Builds Fixed per Author' Report
  - 'Percentage of Successful Builds per Author' Report

**Bamboo Glossary**
Return to bamboo Documentation Home
'Percentage of Successful Builds per Author' Report

This page last changed on Feb 19, 2007 by rosie@atlassian.com.

You can select the author(s) on whom you want to report.

Sample Report: 'Percentage of Successful Builds per Author'

**Percentage of Successful Builds**

![Graph showing percentage of successful builds over time for msmith and tbloggs](image)

**RELATED TOPICS**

- 6.1 Viewing Build Statistics for all Authors
- 6.2 Viewing Build Results for an Author
- 6.3 Generating Reports on selected Authors
  - 'Build Activity per Author' Report
  - 'Number of Build Failures per Author' Report
  - 'Number of Builds Broken per Author' Report
  - 'Number of Builds Fixed per Author' Report
  - 'Percentage of Successful Builds per Author' Report

Bamboo Glossary
Return to Bamboo Documentation Home
07. Working with Comments

This page last changed on Feb 04, 2007 by rosie@atlassian.com.

7. Working with Comments

- 7.1 About Comments
- 7.2 Commenting about a Build Result
- 7.3 Viewing Comments about a Build Result
- 7.4 Viewing Code Check-in Comments
7.1 About Comments

Comments are a useful way to record and share information about builds. There are two types of comments in Bamboo:

- Comments you make when you commit code — these comments are automatically copied into Bamboo from your source-code repository. See Viewing Code Check-in Comments.
- Comments you make about a build result — these are comments that you make ad-hoc about a particular build result. See Commenting about a Build Result and Viewing Comments about a Build Result.

RELATED TOPICS

- 7.1 About Comments
- 7.2 Commenting about a Build Result
- 7.3 Viewing Comments about a Build Result
- 7.4 Viewing Code Check-in Comments

Bamboo Glossary
Return to Bamboo Documentation Home
7.2 Commenting about a Build Result

This page last changed on Feb 19, 2007 by rosie@atlassian.com.

Bamboo allows you to record comments about a build result. This is a convenient way to record relevant information for future reference, and to collaborate with colleagues.

To comment on a build result,

1. From within the Build Result screen, click the 'Comments' tab. A list of existing comments about this build result will be displayed.
2. Type your comment into the 'Add Comment' box, then click the 'Save' button.

Screenshot: Build Result - Comments

Project: Atlassian Buckets
Plan: Main Buckets
Build: 00

Build Result BUCKET-MAIN-69
Labels: NONE

Comments

Rosie Jameson (Apr 28, 2007 19:36:11 AM)
This is another comment.

Matt Ryan (Oct 20, 2006 2:22:16 PM)
This is a reply to Edwin's comment.

Edwin Wong (Dec 8, 2006 4:11:28 PM)
This is a comment.

Add Comment

Comment:

Save

You must login to Bamboo before you can comment on a build result.

RELATED TOPICS

- 7.1 About Comments
- 7.2 Commenting about a Build Result
7.3 Viewing Comments about a Build Result
7.4 Viewing Code Check-in Comments

Bamboo Glossary
Return to Bamboo Documentation Home
7.3 Viewing Comments about a Build Result

This page last changed on Feb 19, 2007 by rosie@atlassian.com.

To view comments about a particular build result,

1. From within the Build Result screen, click the Comments tab. A list of all comments about this build result will be displayed, including author and timestamp:

![Build Result BUCKET-MAIN-89](image)

   **Comments**

   - Rosie Jameson (Jan 29, 2007 10:26:19 AM)
     This is another comment.

   - Matt Ryall (Dec 23, 2006 2:32:51 PM)
     This is a reply to Edwin’s comment.

   - Edwin Wong (Dec 19, 2006 4:42:23 PM)
     This is a comment.

To view comments about all build results for a particular plan,

1. From the Dashboard, click the plan you are interested in.
2. Click the plan’s Completed Builds tab.
3. This will display a list of the plan’s build results. The icon indicates that there are one or more comments about a particular build result. Hold your mouse over the icon to see the comment(s), e.g.:
## Atlassian Bucket - Main Build: Completed Build Results

<table>
<thead>
<tr>
<th>Build Number</th>
<th>Reason</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>BUCKET-MAIN74</td>
<td>Updated by tdavies</td>
<td>8 days ago</td>
</tr>
<tr>
<td>BUCKET-MAIN73</td>
<td>Updated by tdavies</td>
<td>8 days ago</td>
</tr>
<tr>
<td>BUCKET-MAIN72</td>
<td>Updated by tdavies</td>
<td>8 days ago</td>
</tr>
<tr>
<td>BUCKET-MAIN71</td>
<td>Updated by Bamboo Master</td>
<td>2 weeks ago</td>
</tr>
<tr>
<td>BUCKET-MAIN70</td>
<td>Updated by David Loeng</td>
<td>2 weeks ago</td>
</tr>
<tr>
<td>BUCKET-MAIN69</td>
<td>Updated by tdavies</td>
<td>2 months ago</td>
</tr>
<tr>
<td>BUCKET-MAIN68</td>
<td>Updated by David Loeng</td>
<td>2 months ago</td>
</tr>
<tr>
<td>BUCKET-MAIN67</td>
<td>Updated by Rosie Jameson (Jan 29, 2007 10:26:13 AM)</td>
<td>3 months ago</td>
</tr>
<tr>
<td>BUCKET-MAIN66</td>
<td>Updated by Natt Ryall (Dec 29, 2006 2:52:36 PM)</td>
<td>3 months ago</td>
</tr>
<tr>
<td>BUCKET-MAIN65</td>
<td>Updated by Edwin Wong (Dec 19, 2006 4:42:28 PM)</td>
<td>3 months ago</td>
</tr>
<tr>
<td>BUCKET-MAIN64</td>
<td>Updated by Charles Miller</td>
<td>3 months ago</td>
</tr>
<tr>
<td>BUCKET-MAIN63</td>
<td>Updated by Changmin</td>
<td>4 months ago</td>
</tr>
<tr>
<td>BUCKET-MAIN62</td>
<td>Updated by Charles Miller</td>
<td>4 months ago</td>
</tr>
</tbody>
</table>

## Related Topics

- 7.1 About Comments
- 7.2 Commenting about a Build Result
- 7.3 Viewing Comments about a Build Result
- 7.4 Viewing Code Check-in Comments

**Bamboo Glossary**

Return to **Bamboo Documentation Home**
7.4 Viewing Code Check-in Comments

If a build was triggered by a code change, the commit comment (or check-in comment) will be shown in the build result.

To view the code check-in comments for a particular build result,

1. Go to the build result.
2. The build’s commit comment will be shown to the right of the screen, under the heading 'Code Changes'.

A note about build triggering
There are a variety of ways in which a build can be triggered for a plan:

- Code updated — a build can be triggered whenever one or more authors checks-in code.
- Scheduled build — a build can be scheduled to occur at regular intervals.
- Dependency — a build can be triggered whenever a successful build occurs for another plan.
- Manual build — a build can be triggered manually.
- Initial clean build — a build will be triggered when a new plan is created.

The way in which each build was triggered is listed in the 'Reason' column on the Dashboard.

RELATED TOPICS

- 7.1 About Comments
- 7.2 Commenting about a Build Result
- 7.3 Viewing Comments about a Build Result
- 7.4 Viewing Code Check-in Comments

Bamboo Glossary
Return to Bamboo Documentation Home
8. Working with Labels

- 8.1 About Labels
- 8.2 Labelling a Build Result
- 8.3 Removing a Label from a Build Result
- 8.4 Viewing Labelled Build Results
- 8.5 Viewing Popular Labels
8.1 About Labels

A label is a convenient way to tag and group build results that are logically related to each other. Labels can also be used to define RSS feeds.

Labels can be applied to build results automatically, by specifying the label(s) in a build plan (note that only Bamboo administrators can do this). Labels can also be applied ad-hoc to build results by Bamboo users.

RELATED TOPICS

- 8.1 About Labels
- 8.2 Labelling a Build Result
- 8.3 Removing a Label from a Build Result
- 8.4 Viewing Labelled Build Results
- 8.5 Viewing Popular Labels

Bamboo Glossary
Return to Bamboo Documentation Home
8.2 Labelling a Build Result

With Bamboo, you can label your build results in whatever way works best for your team. Labels are not restricted to a particular plan, so you can apply the same label to build results from different plans.

For example, it might not be practical for your QA team to review every build, and you need to know which builds they have reviewed. By using labels such as "qa_passed" and "qa_failed", Bamboo allows them to simply indicate which builds have passed and failed QA.

To label a build result,

1. Go to the build result.
2. Locate the 'Labels' link at the top of the screen: Labels: NONE Add
   (Hint: To view a list of existing labels, click the 'Labels' link.)
3. Click the 'Add' link. This will display the following:

   ![Labels form]

   `The 'Add' link will only be visible if you have logged in to Bamboo.

4. Type the relevant label (or multiple labels, separated by commas). Note that the label will be saved in lowercase characters.
5. Click the 'Done' button.

![You can also label a build result via Instant Messaging (IM).]

RELATED TOPICS

- 8.1 About Labels
- 8.2 Labelling a Build Result
- 8.3 Removing a Label from a Build Result
- 8.4 Viewing Labelled Build Results
- 8.5 Viewing Popular Labels

Bamboo Glossary
Return to Bamboo Documentation Home
8.3 Removing a Label from a Build Result

To remove a label from a build result,

1. Go to the build result.
2. Locate the 'Labels' link at the top of the screen: 
3. Click the 'Edit' link. This will display the following:

   ![Labels input field](image)

4. Click the small red 'x' at the right of the label you want to remove.
5. Click the 'Done' button.

💡 You must login to Bamboo before you can remove a label from a build result.

RELATED TOPICS

- 8.1 About Labels
- 8.2 Labelling a Build Result
- 8.3 Removing a Label from a Build Result
- 8.4 Viewing Labelled Build Results
- 8.5 Viewing Popular Labels

Bamboo Glossary
Return to Bamboo Documentation Home
8.4 Viewing Labelled Build Results

To view all build results which have a particular label,

1. Go to any build result.
2. Click the 'Labels' link at the top of the screen (above the 'Summary' tab).
3. Click the link 'See also labels in all projects'.
4. This will display a list of all labels that are used in Bamboo. Click the label of interest.
5. This will display a list of all build results which have that label.

RELATED TOPICS

- 8.1 About Labels
- 8.2 Labelling a Build Result
- 8.3 Removing a Label from a Build Result
- 8.4 Viewing Labelled Build Results
- 8.5 Viewing Popular Labels

Bamboo Glossary
Return to Bamboo Documentation Home
8.5 Viewing Popular Labels

When labelling a build result, it can be useful to see which labels are most popular, that is, most frequently used by your colleagues.

To view the most popular labels,

1. Go to any build result (not necessarily a labelled one).
2. Click the 'Labels' link at the top of the screen (above the 'Summary' tab).
3. Click the link 'See also labels in all projects'.
4. This will display a list all labels that are used in Bamboo. The most popular labels are indicated by the largest text.

Screenshot: 'Labels'

Labels

This page lists all labels used in Bamboo. The bigger the text, the more build results are associated with this label. Click on a label to see the builds associated with it.

View the labels: Alphabetically | By Popularity

atlassian_bamboo_0_8  bamboo  confluence23  disabled
javapolis  published  qa-passed  qa_failed  qa_passed
random  release

☑ Handy Hint
You can click any label to see a list of all build results which have that label.

RELATED TOPICS

- 8.1 About Labels
- 8.2 Labelling a Build Result
- 8.3 Removing a Label from a Build Result
- 8.4 Viewing Labelled Build Results
- 8.5 Viewing Popular Labels

Bamboo Glossary
Return to Bamboo Documentation Home
9. Subscribing to RSS Feeds

- 9.1 Subscribing to an RSS Feed for All Build Results for All Plans
- 9.2 Subscribing to an RSS Feed for Failed Builds for All Plans
- 9.3 Subscribing to an RSS Feed for All Build Results for a Particular Plan
- 9.4 Subscribing to an RSS Feed for Failed Builds for a Particular Plan
- 9.5 Subscribing to an RSS Feed for Labelled Build Results
9.1 Subscribing to an RSS Feed for All Build Results for All Plans

To subscribe to an RSS feed for all build results for all plans,

1. Go to the Dashboard's 'All' tab.
2. Locate the RSS icon at the bottom of the screen: Feed for all builds or just the failed builds.
3. Right-click the 'all builds' link and copy its URL.
4. Paste the URL into your RSS reader.

RELATED TOPICS

- 9.1 Subscribing to an RSS Feed for All Build Results for All Plans
- 9.2 Subscribing to an RSS Feed for Failed Builds for All Plans
- 9.3 Subscribing to an RSS Feed for All Build Results for a Particular Plan
- 9.4 Subscribing to an RSS Feed for Failed Builds for a Particular Plan
- 9.5 Subscribing to an RSS Feed for Labelled Build Results

Bamboo Glossary
Return to Bamboo Documentation Home
9.2 Subscribing to an RSS Feed for Failed Builds for All Plans

To subscribe to an RSS feed for failed builds for all plans,

1. Go to the Dashboard's 'All' tab.
2. Locate the RSS icon at the bottom of the screen: Feed for all builds or just the failed builds.
3. Right-click the 'failed builds' link and copy its URL.
4. Paste the URL into your RSS reader.

RELATED TOPICS

- 9.1 Subscribing to an RSS Feed for All Build Results for All Plans
- 9.2 Subscribing to an RSS Feed for Failed Builds for All Plans
- 9.3 Subscribing to an RSS Feed for All Build Results for a Particular Plan
- 9.4 Subscribing to an RSS Feed for Failed Builds for a Particular Plan
- 9.5 Subscribing to an RSS Feed for Labelled Build Results

Bamboo Glossary
Return to Bamboo Documentation Home

Return to Bamboo Documentation Home
9.3 Subscribing to an RSS Feed for All Build Results for a Particular Plan

To subscribe to an RSS feed for all build results for a particular plan,

1. Go to the plan.
2. Locate the RSS icon at the bottom of the screen: Feed for all builds or just the failed builds.
3. Right-click the 'all builds' link and copy its URL.
4. Paste the URL into your RSS reader.

RELATED TOPICS

- 9.1 Subscribing to an RSS Feed for All Build Results for All Plans
- 9.2 Subscribing to an RSS Feed for Failed Builds for All Plans
- 9.3 Subscribing to an RSS Feed for All Build Results for a Particular Plan
- 9.4 Subscribing to an RSS Feed for Failed Builds for a Particular Plan
- 9.5 Subscribing to an RSS Feed for Labelled Build Results

Bamboo Glossary
Return to Bamboo Documentation Home
9.4 Subscribing to an RSS Feed for Failed Builds for a Particular Plan

To subscribe to an RSS feed for failed builds for a particular plan,

1. Go to the plan.
2. Locate the RSS icon at the bottom of the screen: Feed for all builds or just the failed builds.
3. Right-click the 'failed builds' link and copy its URL.
4. Paste the URL into your RSS reader.

RELATED TOPICS

- 9.1 Subscribing to an RSS Feed for All Build Results for All Plans
- 9.2 Subscribing to an RSS Feed for Failed Builds for All Plans
- 9.3 Subscribing to an RSS Feed for All Build Results for a Particular Plan
- 9.4 Subscribing to an RSS Feed for Failed Builds for a Particular Plan
- 9.5 Subscribing to an RSS Feed for Labelled Build Results

Bamboo Glossary
Return to Bamboo Documentation Home
9.5 Subscribing to an RSS Feed for Labelled Build Results

This page last changed on Feb 19, 2007 by rosie@atlassian.com.

To subscribe to an RSS feed for all build results with a particular label,

1. Go to the Dashboard.
2. Click any build result (not necessarily a labelled one).
3. Click the 'Labels' link at the top of the screen (above the 'Summary' tab).
4. This will display a list of any labels that are used in the build's plan. Click the link 'See also labels in all projects'.
5. This will display a list of all labels that are used in Bamboo. Click the label of interest.
6. This will display a list of build results which have been labelled with your chosen label. Locate the RSS icon at the bottom of the screen: Feed for builds labelled
7. Right-click the 'Feed for builds labelled' link and copy its URL.
8. Paste the URL into your RSS reader.

What is a label?

RELATED TOPICS

- [9.1 Subscribing to an RSS Feed for All Build Results for All Plans](#)
- [9.2 Subscribing to an RSS Feed for Failed Builds for All Plans](#)
- [9.3 Subscribing to an RSS Feed for All Build Results for a Particular Plan](#)
- [9.4 Subscribing to an RSS Feed for Failed Builds for a Particular Plan](#)
- [9.5 Subscribing to an RSS Feed for Labelled Build Results](#)

Bamboo Glossary
Return to [Bamboo Documentation Home](#)
10. Working with Instant Messenger (IM) Notifications

This page last changed on Feb 11, 2007 by rosie@atlassian.com.

10. Working with Instant Messenger (IM) notifications

- 10.1 About Instant Messenger (IM) Notifications
- 10.2 Labelling a Build Result via IM
- 10.3 Commenting about a Build Result via IM
10.1 About Instant Messenger (IM) Notifications

This page last changed on May 07, 2007 by rosie@atlassian.com.

Bamboo can send you notifications about build results for a particular plan(s). Each plan's recipients are specified by a Bamboo administrator, but you can choose whether you would like to receive your Bamboo notifications via email and/or instant messenger (IM)*.

As well as receiving IM notifications, you can interact with Bamboo via IM. By responding to an IM notification, you can:

- Label a build result via IM
- Comment about a build result via IM

*See 11.2 Changing your Notification Preferences.

RELATED TOPICS

- 10.1 About Instant Messenger (IM) Notifications
- 10.2 Labelling a Build Result via IM
- 10.3 Commenting about a Build Result via IM

Bamboo Glossary
Return to Bamboo Documentation Home
10.2 Labelling a Build Result via IM

You can respond to a Bamboo IM notification message with commands to label or comment on a build result.

To label a build result via Instant Messaging (IM),

In your Instant Messenger, type your comment in the following format:

```
label [build key] <labels>
```

Screenshot: Interacting with Bamboo via IM

- [1:07 PM] bamboo: Test Bash build 59 has FAILED (0 tests failed)
- http://localhost:8085/browse/TST-BLAH-59
- [1:07 PM] edwin: comment sorry guys, this one is my fault
- [1:07 PM] edwin: label broken_by_me
- [1:07 PM] bamboo: TST-BLAH-59 labelled 'broken_by_me'
- [1:08 PM] bamboo: Test Bash build 60 has FAILED (0 tests failed)
- http://localhost:8085/browse/TST-BLAH-60
- [1:08 PM] edwin: comment grr! this is still breaking!
- [1:08 PM] bamboo: TST-BLAH-60 commented.
- [1:08 PM] edwin: label broken_by_me
- [1:08 PM] bamboo: TST-BLAH-60 labelled 'broken_by_me'
- [1:09 PM] edwin: comment TST-BLAH-59 This is caused by a lot of stupidity.

What is a label?

RELATED TOPICS

- 10.1 About Instant Messenger (IM) Notifications
- 10.2 Labelling a Build Result via IM
- 10.3 Commenting about a Build Result via IM
10.3 Commenting about a Build Result via IM

This page last changed on Feb 27, 2007 by rosie@atlassian.com.

You can respond to a Bamboo IM notification message with commands to label or comment on a build result.

To comment on a build result via Instant Messaging (IM),

In your Instant Messenger, type your comment in the following format:

```
comment [build key] <comment message>
```

Screenshot: Interacting with Bamboo via IM

---

**RELATED TOPICS**

- 10.1 About Instant Messenger (IM) Notifications
- 10.2 Labelling a Build Result via IM
- 10.3 Commenting about a Build Result via IM

Bamboo Glossary

Document generated by Confluence on May 08, 2007 23:50
11. Editing your User Profile

This page last changed on Feb 11, 2007 by rosie@atlassian.com.

11. Editing your User Profile

- 11.1 Changing your Password
- 11.2 Changing your Notification Preferences
- 11.3 Associating your Author Name with your User Profile
11.1 Changing your Password

To change your Bamboo password,

1. Click the 'Profile' link in the top right corner of the screen. This will display the 'User Profile' screen.
2. Click the 'Change Password' link.
3. Type your old and new passwords.
4. Click the 'Change Password' button.

RELATED TOPICS

- 11.1 Changing your Password
- 11.2 Changing your Notification Preferences
- 11.3 Associating your Author Name with your User Profile

Bamboo Glossary
Return to Bamboo Documentation Home
11.2 Changing your Notification Preferences

This page last changed on May 07, 2007 by rosie@atlassian.com.

Bamboo can send you notifications about build results. You can choose whether to receive your notifications via email or Instant Messaging (IM), or both. You can also choose not to receive notifications at all.

To change your notification preferences,

1. Click the 'Profile' link in the top right corner of the screen. This will display the 'User Profile' screen.
2. Click the 'Edit Profile' link. The 'Edit User Profile' screen will be displayed, as shown below.
3. Under 'Set Your Notification Preference', choose how you would like to receive your build result notifications:
   - 'Do not send notifications'
   - 'Send instant message'*
   - 'Send email'
   - 'Send email and instant message'*
4. Click the 'Save' button.

* If you select this option, you need to specify your IM address in the 'Jabber Address' field.

Screenshot: User Profile

<table>
<thead>
<tr>
<th>Edit User Profile</th>
</tr>
</thead>
<tbody>
<tr>
<td>Username:</td>
</tr>
<tr>
<td>Full Name:</td>
</tr>
<tr>
<td>Email:</td>
</tr>
<tr>
<td>Jabber Address:</td>
</tr>
</tbody>
</table>

The address to which instant messages will be sent to if they are enabled for a build.

Source Repository Alias: None

The repository name the user commits with in the repositories.

Set Your Notification Preference

How would you like bamboo to send you notifications:
- [ ] Do not send notifications
- [ ] Send instant message
- [ ] Send email
- [ ] Send email and instant message

Please ensure that you have set the appropriate address for the notification type.

Save  Cancel

⚠️ Only Bamboo administrators can enable notifications for a plan. For details please see the Bamboo Administrator’s Guide.
11.1 Changing your Password
11.2 Changing your Notification Preferences
11.3 Associating your Author Name with your User Profile

Bamboo Glossary
Return to Bamboo Documentation Home
11.3 Associating your Author Name with your User Profile

An author is any person who checks-in code to a repository that is associated with a Bamboo plan. An author need not be a Bamboo user. Your Author Name is your login name for the source-code repository.

⚠️ If your Bamboo User Profile has not yet been associated with your Author Name, then:

- your 'My Bamboo' screen will not contain any data about your recent builds.
- your 'Author' information will not include a 'User Details' tab.

To associate your Author Name with your User Profile,

1. Click the 'Profile' link in the top right corner of the screen. This will display the 'User Profile' screen.
2. Click the 'Edit Profile' link.
3. In the 'Source Repository Alias' field, select your Author Name from the list. If your Author Name does not appear in the list, select 'Add Alias' (the second item in the list) then type your Author Name in the 'New Alias' field. Note that your Author Name (Alias) need not be identical to your User Name.
4. Click the 'Save' button.

Screenshot: Bamboo User Profile

Bamboo Profile: Joe Bloggs

Username: jbloggs
Full Name: Joe Bloggs
Email: jbloggs@mycompany.com
Jabber Address: jbloggs@chat.mycompany.com
Source Repository Alias: Add Alias
New Alias: jbloggs

The address to which instant messages will be sent if they are enabled for a build.

Save  Cancel

RELATED TOPICS

- **11.1 Changing your Password**
- **11.2 Changing your Notification Preferences**
- **11.3 Associating your Author Name with your User Profile**
Appendix A. Bamboo Glossary

activity log

Every plan has an activity log. An activity log is a temporary display of the latest output from the plan's most recent build log.

artifact

An artifact (e.g. a JAR file) is something created by a build. A build's artifacts are specified in the build's plan by a Bamboo administrator.

author

An author is any person who checks-in code to a repository that is associated with a Bamboo plan. An author need not be a Bamboo user.

build

A build is one execution of a plan.

Every build has a Build Number, which is appended to the relevant Plan Key to form the Build Key. For example, if a plan with the key "CRM-BRANCH" is executed for the seventeenth time, the build key will be "CRM-BRANCH-17".

build activity

Build activity is the number of builds that occur in a given period of time.

build duration

Build duration is the total time taken to execute a build plan — that is, the time taken to compile the code and run all of the plan's tests. Variations in a plan's build duration can be monitored over time.

build log

Every build has a build log. A build log is a permanent record of all the output generated by compiling the plan's source-code and executing the tests.

build plan

See plan.
A build queue controls the sequence of builds. Bamboo administrators can specify how many build queues there are in the system. For example, if there are two build queues, two builds can occur in parallel while subsequent builds will wait in a queue. Build queues are displayed on the Dashboard.

Every completed build has a build result:

- 'Successful' — the code compiled, with or without errors, and all tests completed successfully.
- 'Failed' — either the code did not compile, or at least one test failed.

Additionally,

- if the build result is 'Failed', and the previous build result was 'Successful', the build is said to be 'Broken'.
- if the build result is 'Successful', and the previous build result was 'Failed', the build is said to be 'Fixed'.

Build telemetry is the insight provided by Bamboo's dynamic reports, charts and collation of build metrics. Build telemetry helps identify trends across build plans and across authors — not just focusing on the results of a single build.

A builder is a software compiler program external to Bamboo. Bamboo supports multiple builders. Once a builder is defined in the Bamboo system, it can then be specified in build plans by a Bamboo administrator.

Each Bamboo user can nominate their favourite plans — that is, the plans they work with the most.

Each user's favourites are displayed on the 'My' page of the Dashboard. Bamboo administrators can also configure each plan to send build result notifications to users who have nominated the plan as one of their favourites.

A label is a convenient way to tag and group build results that are logically related to each other. Labels can also be used to define RSS feeds.

Labels can be applied to build results automatically, by specifying the label(s) in a build plan (note that
only Bamboo administrators can do this). Labels can also be applied ad-hoc to build results by Bamboo users.

**plan**

A Bamboo plan (or build plan) is the "recipe" for a build.

A plan defines: what gets built (i.e. the source-code repository); how the build is triggered; which builder to use; what tests to run; what artifacts the build will produce; who will be notified of the build result; and any labels with which the build result or build artifacts will be tagged.

Every plan belongs to a project.
Each plan has a Plan Key, which is prefixed by the relevant Project Key. E.g. the "CRM" project could have plans "CRM-TRUNK" and "CRM-BRANCH".
Projects and plans can only be configured by Bamboo administrators.

**project**

A project is a collection of plans.
A project enables easy identification of plans that are logically related to each other, which is useful for instance when generating reports across multiple plans.
Each project has a Name (e.g. "CRM System") and a Key (e.g. "CRM"). The Project Key is prefixed to the relevant Plan Keys, e.g. the "CRM" project could have plans "CRM-TRUNK" and "CRM-BRANCH".

**triggering**

There are a variety of ways in which a build can be triggered for a plan:

- Code updated — a build can be triggered whenever one or more authors checks-in code.
- Scheduled build — a build can be scheduled to occur at regular intervals.
- Dependency — a build can be triggered whenever a successful build occurs for another plan.
- Manual build — a build can be triggered manually.
- Initial clean build — a build will be triggered when a new plan is created.

The way in which each build was triggered is listed in the 'Reason' column on the Dashboard. Note that build triggering can only be configured by a Bamboo administrator.

Return to Bamboo Documentation Home
activity log

This page last changed on Jan 28, 2007 by rosie@atlassian.com.

Every plan has an activity log. An activity log is a temporary display of the latest output from the plan's most recent build log.
artifact

An artifact (e.g., a JAR file) is something created by a build. A build’s artifacts are specified in the build’s plan by a Bamboo administrator.
An author is any person who checks-in code to a repository that is associated with a Bamboo plan. An author need not be a Bamboo user.
build

A build is one execution of a plan.

Every build has a Build Number, which is appended to the relevant Plan Key to form the Build Key. For example, if a plan with the key "CRM-BRANCH" is executed for the seventeenth time, the build key will be "CRM-BRANCH-17".
build activity

Build activity is the number of builds that occur in a given period of time.
build duration

Build duration is the total time taken to execute a build plan — that is, the time taken to compile the code and run all of the plan's tests. Variations in a plan's build duration can be monitored over time.
build log

This page last changed on Jan 29, 2007 by rosie@atlassian.com.

Every build has a build log. A build log is a permanent record of all the output generated by compiling the plan's source-code and executing the tests.
build plan

This page last changed on Feb 04, 2007 by rosie@atlassian.com.

See plan.
build queue

This page last changed on May 07, 2007 by rosie@atlassian.com.

A build queue controls the sequence of builds. Bamboo administrators can specify how many build queues there are in the system. For example, if there are two build queues, two builds can occur in parallel while subsequent builds will wait in a queue. Build queues are displayed on the Dashboard.
build result

Every completed build has a build result:

- 'Successful' — the code compiled, with or without errors, and all tests completed successfully.
- 'Failed' — either the code did not compile, or at least one test failed.

Additionally,

- if the build result is 'Failed', and the previous build result was 'Successful', the build is said to be 'Broken'.
- if the build result is 'Successful', and the previous build result was 'Failed', the build is said to be 'Fixed'.

build telemetry

Build telemetry is the insight provided by Bamboo's dynamic reports, charts and collation of build metrics. Build telemetry helps identify trends across build plans and across authors — not just focusing on the results of a single build.
A builder is a software compiler program external to Bamboo. Bamboo supports multiple builders. Once a builder is defined in the Bamboo system, it can then be specified in build plans by a Bamboo administrator.
Each Bamboo user can nominate their favourite plans — that is, the plans they work with the most.

Each user's favourites are displayed on the 'My' page of the Dashboard. Bamboo administrators can also configure each plan to send build result notifications to users who have nominated the plan as one of their favourites.
A label is a convenient way to tag and group build results that are logically related to each other. Labels can also be used to define RSS feeds.

Labels can be applied to build results automatically, by specifying the label(s) in a build plan (note that only Bamboo administrators can do this). Labels can also be applied ad-hoc to build results by Bamboo users.
plan

A Bamboo plan (or build plan) is the "recipe" for a build.

A plan defines: what gets built (i.e. the source-code repository); how the build is triggered; which builder to use; what tests to run; what artifacts the build will produce; who will be notified of the build result; and any labels with which the build result or build artifacts will be tagged.

Every plan belongs to a project. Each plan has a Plan Key, which is prefixed by the relevant Project Key. E.g. the "CRM" project could have plans "CRM-TRUNK" and "CRM-BRANCH". Projects and plans can only be configured by Bamboo administrators.
A project is a collection of **plans**.
A project enables easy identification of plans that are logically related to each other, which is useful for instance when generating **reports** across multiple plans.
Each project has a Name (e.g. "CRM System") and a Key (e.g. "CRM"). The Project Key is prefixed to the relevant Plan Keys, e.g. the "CRM" project could have plans "CRM-TRUNK" and "CRM-BRANCH".
There are a variety of ways in which a build can be triggered for a plan:

- Code updated — a build can be triggered whenever one or more authors checks-in code.
- Scheduled build — a build can be scheduled to occur at regular intervals.
- Dependency — a build can be triggered whenever a successful build occurs for another plan.
- Manual build — a build can be triggered manually.
- Initial clean build — a build will be triggered when a new plan is created.

The way in which each build was triggered is listed in the ‘Reason’ column on the Dashboard. Note that build triggering can only be configured by a Bamboo administrator.
Bamboo Development Hub

This page last changed on Mar 14, 2007 by jnolen.

- **Bamboo API**
  - **Bamboo Plugin Developer's Guide**
    - Build Complete Action Module
    - Build Processor Module
    - Builder Plugin Module
    - Getting Started
    - Index Reader Module
    - Notification Condition Module
    - Post Build Index Writer Module
    - Pre Build Action Module
    - Report Module
    - Web Item Module
    - Web Section Module
    - XWork Plugin Module
- **Bamboo Plugin Tutorial**
  - Tutorial 1 - Getting Started with a Simple Post Build Labeller
  - Tutorial 2 - Configurable Regex Labeller
Bamboo API

This page last changed on Feb 14, 2007 by jnolen.

The API documentation is installed on your Bamboo server and can be found at: http://your-bamboo-host/api/index.action (replace 'your-bamboo-host' with your Bamboo server's name).

If you have installed Bamboo locally, you can view the API documentation at: http://localhost:8085/api/index.action.

RELATED TOPICS

Unable to render {children} Page not found: Appendix A. Extending Bamboo
Bamboo Documentation Home
Bamboo Plugin Developer's Guide

This page last changed on May 08, 2007 by rosie@atlassian.com.

Bamboo Plugins Overview

A Bamboo plugin is a single JAR containing code, a plugin descriptor (XML) and usually some Freemaker template files to render HTML.

The plugin descriptor is the only mandatory part of the plugin. It must be called atlassian-plugin.xml and be located in the root of your JAR file.

Each plugin consists of one or more plugin modules. These are of different types (for example a report, or a post-build action) and each has an individual XML element describing it. Each module is described below together with the XML element required for it.

Here is a sample of the descriptor with highlighted elements:

```xml
<!-- the plugin key must be unique, think of it as the 'package' of the plugin -->
<atlassian-plugin key="com.atlassian.plugin.sample" name="Sample Plugin">
  <!-- a short block describing the plugin itself -->
  <plugin-info>
    <description>This is a brief textual description of the plugin</description>
    <!-- the version of the plugin -->
    <version>1.1</version>
    <!-- the versions of the application this plugin is for -->
    <application-version min="3.0" max="3.0"/>
    <!-- details of the plugin vendor -->
    <vendor name="Atlassian Software Systems Pty Ltd" url="http://www.atlassian.com"/>
  </plugin-info>
  ... 1 or more plugin modules ...
</atlassian-plugin>
```

Each plugin has a plugin key which is unique among all plugins (eg "com.atlassian.plugin.sample"). Semantically this equates to the package of a Java class. Each module within the plugin also has a module key which is unique within the plugin (eg "myreport"). Semantically this equates to the class name of a Java class.

The plugin key + module key are combined to make the complete key of the plugin module (combining the examples above, the complete key would be "com.atlassian.plugin.sample:myreport"). Note: a : is used to separate the plugin key from the module key.

Setting up a Bamboo Plugin Project

Refer to the Getting Started guide.

Deploy a Bamboo Plugin

Installing plugins in Bamboo is easy.
Once you have downloaded or created your plugin jar, follow these steps:

1. Shut down Bamboo
2. copy '$MY_COOL_PLUGIN.jar' into '../webapp/WEB-INF/lib/'
3. Start up Bamboo. Your plugin should be automatically installed and activated.
4. Enjoy!

Bamboo Plugin Module Types

The following types of plugin modules are supported by Bamboo

<table>
<thead>
<tr>
<th>Module Type</th>
<th>Since version…</th>
<th>Documentation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>builder</td>
<td>1.0</td>
<td>Builder Plugin Module</td>
<td>Add new builders to Bamboo</td>
</tr>
<tr>
<td>xwork</td>
<td>1.0</td>
<td>XWork Plugin Module</td>
<td>XWork actions and views bundled with the plugin. This enables building generic user interfaces.</td>
</tr>
<tr>
<td>report</td>
<td>1.0</td>
<td>Report Module</td>
<td>Defines a report of build telemetry data.</td>
</tr>
<tr>
<td>preBuildAction</td>
<td>1.1</td>
<td>Pre Build Action Module</td>
<td>Prepends a custom synchronous process to the build. Occurs before the builder has run.</td>
</tr>
<tr>
<td>buildProcessor</td>
<td>1.0</td>
<td>Build Processor Module</td>
<td>Append a custom synchronous process to the build. Occurs after the builder has run.</td>
</tr>
<tr>
<td>buildCompleteAction</td>
<td>1.0</td>
<td>Build Complete Action Module</td>
<td>Add a custom asynchronous action after the build process has completed.</td>
</tr>
<tr>
<td>postBuildIndexWriter</td>
<td>1.0</td>
<td>Post Build Index Writer Module</td>
<td>Writes custom build data into the index for report generation.</td>
</tr>
<tr>
<td>indexReader</td>
<td>1.0</td>
<td>Index Reader Module</td>
<td>Reads out custom index information written by the postBuildIndexWriter from the index.</td>
</tr>
<tr>
<td>web-item</td>
<td>1.0</td>
<td>Web Item Module</td>
<td>Add new links to the Bamboo interface</td>
</tr>
<tr>
<td>web-section</td>
<td>1.0</td>
<td>Web Section Module</td>
<td>Add a new section to the Bamboo interface</td>
</tr>
<tr>
<td>web-resource</td>
<td>1.0</td>
<td>Web Resource Module</td>
<td>Add a new resource to</td>
</tr>
</tbody>
</table>
Built-in Bamboo system plugins

A number of functions and areas within Bamboo are shipped as built in plugins. These can also be useful for plugin developers who want to know more about how to create their own plugins, as they showcase the functionality that can be built.

The system plugins are referenced from the following files (located in /WEB-INF/classes):

- system-builder-plugin.xml - the built in builders, including Ant, Maven, and Maven 2.
- system-clover-plugin.xml - the built in Clover analytics.
- system-jira-plugin.xml - the built in tab view of JIRA issues in a build.
- system-labelling-plugin.xml - the built in automatic build labeller.
- system-reports-plugin.xml - the built reports of builds grouped by time periods under the Reports tab.
- system-webUI-plugin.xml - the built in menu items under the Administration tab as well as the tab menu items on the View Plan page and the View Build Results page.
- system-notifications-plugin.xml - the build in notification conditions, including all builds, failed builds, after X failed builds.

RELATED TOPICS

- Bamboo API
- Bamboo Plugin Developer's Guide
  - Build Complete Action Module
  - Build Processor Module
  - Builder Plugin Module
  - Getting Started
  - Index Reader Module
  - Notification Condition Module
  - Post Build Index Writer Module
  - Pre Build Action Module
  - Report Module
  - Web Item Module
  - Web Section Module
  - XWork Plugin Module
- Bamboo Plugin Tutorial
  - Tutorial 1 - Getting Started with a Simple Post Build Labeller
  - Tutorial 2 - Configurable Regex Labeller

Bamboo Documentation Home
Build Complete Action Module

This page last changed on May 08, 2007 by rosie@atlassian.com.

Description

Like the Build Processor Module, this allows you to specify a custom action to take place. However, the difference is that this will run after the full build result has been registered. The build is deemed to have completed before the BuildCompleteAction is fired. Hence, build complete actions cannot impact the state of the build.

Sample Module Descriptor Element

```xml
<buildCompleteAction key="autoLabeller" name="Build Automatic Labeller"
    class="com.atlassian.bamboo.labels.AutoLabelBuildCompleteAction">
    <resource type="freemarker" name="edit"
        location="templates/plugins/buildCompleteAction/autoLabellingEdit.ftl"/>
    <resource type="freemarker" name="view"
        location="templates/plugins/buildCompleteAction/autoLabellingView.ftl"/>
    <description>An automatic labelling plugin.</description>
</buildCompleteAction>
```
Build Processor Module

This page last changed on Feb 21, 2007 by edwin@atlassian.com.

Description

The BuildProcessor module allows you to define a custom process that runs during the build.

This will occur immediately after the builder has completed execution and the test results have been captured. The BuildProcessor forms part of build execution run and the result of the execution is only registered after your custom BuildProcessor has completed. This means that your plugin has the capability to affect the the final BuildState of your build (i.e. success/failure).

Sample Module Descriptor Element

```xml
<buildProcessor key="cloverResultCollector" name="Clover Results Collector"
   class="com.atlassian.bamboo.builder.coverage.CloverBuildProcessor">
   <skipIfFailed>true</skipIfFailed>
   <resource type="freemarker" name="edit"
      location="templates/plugins/buildProcessor/cloverProcessorEdit.ftl"/>
   <resource type="freemarker" name="view"
      location="templates/plugins/buildProcessor/cloverProcessorView.ftl"/>
   <description>A Clover report parser and data collector</description>
</buildProcessor>
```
Builder Plugin Module

This page last changed on Feb 21, 2007 by edwin@atlassian.com.

Description

A plugin module which defines a builder in Bamboo, such as Maven, Maven2, or Ant.

Sample Module Descriptor Element

```xml
<!-- the module class must implement com.atlassian.bamboo.builder.Builder -->
<builder key="mvn2" name="Maven 2.x Builder"
    class="com.atlassian.bamboo.builder.Maven2Builder">
    <description>A Maven 2.x Builder</description>
    <resource type="freemarker" name="edit"
        location="templates/plugins/builder/mavenBuilderEdit.ftl"/>
    <resource type="freemarker" name="view"
        location="templates/plugins/builder/mavenBuilderView.ftl"/>
</builder>
```
Getting Started

This page last changed on Mar 15, 2007 by jnolen.

Requirements

- You will need to have Maven 2 installed. You can download Maven 2 here.
- A copy of Bamboo 1.0, either built from source or a binary distribution, so that you can test your plugin when you develop.
- We strongly recommend that you build with the Bamboo Plugin Development Kit, available here.
- These instructions assume your IDE is IDEA. You will need to ensure your dependencies are set up correctly if you use any other IDE.

Setting up the project

Inside the Bamboo Development kit, you will need to change the pom.xml file to correctly setup your project. Within this file, you will need to change the following xml elements:

- <groupId> - this is the group identifier for your plugin. It is typically something similar to a Java package name.
- <artifactId> - this defines the file name of your plugin JAR file.
- <version> - this defines the version of your plugin.
- <name> - this defines the name of your plugin.
- <scm> - this defines your source repository URL

Once this is done, you can run the command mvn idea:idea which will download your dependencies (including the bamboo libraries) and build an IDEA project file $MY_PLUGIN_NAME.ipr. To begin development, simply launch the IDEA project file created.

Once IDEA is up, you will also need to modify the file /src/main/resources/atlassian-plugin.xml to give your plugin a name and a plugin key. You should also fill in your plugin meta-data.

That's it, you should now be ready to start coding your Bamboo plugin.
Index Reader Module

This page last changed on Feb 21, 2007 by edwin@atlassian.com.

Description

Written in conjunction with Post Build Index Writer Module, the IndexReader will translate the fields in the index and re-insert the information into a BuildResultSummary object, which has a specially designated customBuildData map for this purpose.

Sample Module Descriptor Element

```xml
<indexReader key="cloverIndexReader" name="Reads Clover result values from index"
    class="com.atlassian.bamboo.builder.coverage.CloverIndexReader">
    <description>Reads the clover result from an index document and populates into build result summary</description>
</indexReader>
```
Notification Condition Module

This page last changed on May 08, 2007 by bmccoy.

Description

A plugin module which allows you to define your own notification condition

You can use this plugin to

- Set conditions as to whether or not to send a notification
- Generate customised notification content

Example

```xml
<!-- the module class must implement com.atlassian.bamboo.notification.NotificationCondition -->
<notification-condition key="buildCompleted.XFailedBuilds" name="After X Failed Builds Completed" class="com.atlassian.bamboo.notification.conditions.AfterXFailedBuildsCondition">
  <description>Send Notification After X Failed Builds</description>
  <resource type="freemarker" name="edit" location="templates/plugins/notifications/afterXFailedEdit.ftl"/>
  <resource type="freemarker" name="view" location="templates/plugins/notifications/afterXFailedView.ftl"/>
</notification-condition>
```

Other Information regarding the NotificationCondition class

Several of the interface methods for this class accept an Event object as a parameter. Current functionality only allows this to be a BuildCompletedEvent. This event will contain the Build object, BuildResults object and BuildResultsSummary object for you to use.
Post Build Index Writer Module

This page last changed on Feb 21, 2007 by edwin@atlassian.com.

Description

The PostBuildIndexWriter allows you to write your custom data for a build into the index, which allows for future retrieval in your custom Report Module. The PostBuildIndexWriter will be invoked in three places in Bamboo: when a build completes and it indexes, operations which requires a re-index of a particular build (result), and when you run the re-index all action under the Administration tab.

The PostBuildIndexWriter should always be written in conjunction with a Index Reader Module which will be able to retrieve the data in the index.

Sample Module Descriptor Element

```xml
<postBuildIndexWriter key="cloverIndexWriter" name="Write Clover Result to Index"
  class="com.atlassian.bamboo.builder.coverage.CloverPostBuildIndexWriter">
  <description>Writes the clover result in a build results to an index document</description>
</postBuildIndexWriter>
```
Pre Build Action Module

This page last changed on May 08, 2007 by edwin@atlassian.com.

Description

The PreBuildAction module allows you to define a custom process that runs before your build begins.

This will occur immediately before the builder begins execution. The PreBuildAction will have access to the BuildResults object which contains the information for the build.

Sample Module Descriptor Element

```xml
<preBuildAction key="vcsVersion" name="VCS Version Collector"
    class="com.atlassian.bamboo.vcsversion.VCSVersionReader">
    <description>A custom action that reads the identifier of a source repository version and stores it into the custom data map of a build.</description>
    <resource type="freemarker" name="edit"
        location="templates/plugins/preBuildAction/vcsVersionReaderEdit.ftl"/>
    <resource type="freemarker" name="view"
        location="templates/plugins/preBuildAction/vcsVersionReaderView.ftl"/>
</preBuildAction>
```
Report Module

This page last changed on Feb 21, 2007 by edwin@atlassian.com.

Description

This defines a report module. A report module will appear under the Reports tab.

A report typically consists of two objects:

- A ReportCollector object implementing the
  `com.atlassian.bamboo.reports.collector.ReportCollector` interface. This takes in a list of
  builds and generates a DataSet.
- A ReportLineChart object extending the
  `com.atlassian.bamboo.reports.charts.BambooReportLineChart` class. This chart will be
  responsible for rendering the dataset results generated by the ReportCollector. Charts in Bamboo
  are generated via `jFreeChart`

Sample Module Descriptor Element

```
<report key="ratioOfSuccess" name="Percentage of Successful Builds"
  class="com.atlassian.bamboo.reports.collector.RatioOfSuccessCollector">
  <description>Comparing success percentages gives you an idea of how stable a build is
  compared to one another.
  100% means your build is always rock solid. 0% means something is seriously
  wrong.</description>
  <chartClass>com.atlassian.bamboo.reports.charts.BuildSummarySuccessRatioLineChart</chartClass>
</report>
```
Web Item Module

This page last changed on Mar 05, 2007 by bmccoy.

Description

The WebItem allows you to define a link in the Bamboo system. (Usually in some form of menu).

Currently, you can use the web-item to add links to three locations

- The Administration Menu
- The Plan Sub Menu (tabs on the View Plan page)
- The Results Sub Menu (tabs on the View Build Results page)

Sample Module Descriptor Element

```xml
<web-item key="pipelineConfig" name="Build Queues" section="system.admin/builds" weight="20">
  <link>/admin/configurePipeline!default.action</link>
  <condition class="com.atlassian.bamboo.plugins.web.conditions.AdminPermissionCondition" />
</web-item>
```

Module Components

- **key** - this is the unique identifier of the web-item, it is also used by Bamboo to give the link an id.
- **name** - in the plan sub menu and results sub menu this is used to determine if the current link (tab) is active
- **section** - the section is made of of the parent section's location followed buy the name of the parent section. In Bamboo this is used to retrieve the appropriate web-items for the menu. (see [Web Section Module](#))
- **weight** - this is used to determine the order of the items on the page
- **label** - this will be displayed on the screen and can be plain text or a property key
- **link** - the link is the url the link will point to. It can be absolute or relative to Bamboo's context path
- **condition** - by implementing the com.atlassian.plugin.web.Condition class you can add rules to determine whether the link will be displayed or not.

Both the **link** and the **id** can make use of parameters passed to the page. For example:

```xml
<link>/build/viewBuildFiles.action?buildKey=${buildKey}</link> where ${buildKey} is the parameter name.
```
Web Section Module

This page last changed on Mar 05, 2007 by bmccoy.

Description

The WebSection module is used to provide a section or grouping of Web Item Module.

Currently, Web Sections are used to group Web Items for the Administration Menu, Plan Sub Menu, Results Sub Menu

Sample Module Descriptor Element

```
<web-section key="builds" name="Builds" location="system.admin" weight="100">
  <label key="websections.system.admin.build"/>
  <icon height="16" width="16">
    <link>/images/icons/icon_spanner.gif</link>
  </icon>
</web-section>
```

Notes

The section is only displayed on the Administration Menu but it is required for all locations as Bamboo uses it to place the web items.

Which menu the item gets placed in is determined by the location attribute. The following table indicates the location attribute required for each menu:

<table>
<thead>
<tr>
<th>Menu</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Administration Menu</td>
<td>system.admin</td>
</tr>
<tr>
<td>Plan Sub Men</td>
<td>build.subMenu</td>
</tr>
<tr>
<td>Build Results Sub menu</td>
<td>results.subMenu</td>
</tr>
</tbody>
</table>
XWork Plugin Module

This page last changed on Feb 21, 2007 by edwin@atlassian.com.

Description

Each XWork module is deployed as a plugin module of type xwork and contains one of more XWork package elements.

Here is an example atlassian-plugin.xml file containing a single XWork module:

The xwork plugin module allows you to define your own xwork package and actions that you can access.

To build the action into the system, you will typically need to add a Web Item Module to link to your action.

Sample Module Descriptor Element

```
<xwork key="viewCloverResult" name="View Clover Result">
  <package name="cloverPlugin" extends="buildView">
    <result name="success" type="freemarker" file="/plugins/clover-plugin/viewCloverResult.ftl">
    </result>
    <result name="error" type="freemarker" file="/error.ftl">
    </result>
  </package>
</xwork>
```
Bamboo Plugin Tutorial

This page last changed on Feb 27, 2007 by jnolen.

Introduction

The purpose of this tutorial is to demonstrate how you can add custom functionality to Bamboo via plugins. The tutorial aims to give you a good starting point for Bamboo plugin development, and how the different Bamboo plugin modules can work together. In this tutorial, we will run through the development of a plugin derived from a real use-case requirement.

Bamboo Labeller Plugin

There are many cases when builds in Bamboo fail because of particular, recurring errors. For example, a functional test in Confluence may periodically fail because of an OutOfMemoryError when things get hectic. It would be useful for developers to keep track of these particular builds, so they can look into it further. For this to happen, Bamboo will need to:

- Parse the error logs after a failed build.
- Look for the text `java.lang.OutOfMemoryError` in the log
- If found, tag the build with a label, say `out_of_memory`

Once that's done, the developer can set up a RSS feed on the `out_of_memory` tag. They will then be able to keep track of the builds which fail with an OutOfMemoryError. We can even extend this concept further, by replacing the search for `java.lang.OutOfMemoryError` with any regular expression, tagging it with a label of choice.

Contents

- Tutorial 1 - Getting Started with a Simple Post Build Labeller
- Tutorial 2 - Configurable Regex Labeller
Tutorial 1 - Getting Started with a Simple Post Build Labeller

This page last changed on May 08, 2007 by rosie@atlassian.com.

The source code to the plugin used in this tutorial is available on the Atlassian public source repository. You can check out the source code here.

Step 1 - Setting up the project.

The first thing you need to do is to set up your Bamboo Plugin project and source directories. The instructions for how to do this are available here.

In the atlassian-plugin.xml located under /src/main/resources/, you will need to give the plugin a unique key, as well as some meta information about this plugin. As our plugin simply labels, we have called it "labeller". Below is the atlassian-plugin.xml for our labelling plugin:

```
<atlassian-plugin key="com.atlassian.bamboo.plugin.labeller" name="Build Labeller">
  <plugin-info>
    <description>Bamboo Labeller</description>
    <version>1.0</version>
    <application-version min="1.0" max="1.0"/>
    <vendor name="Atlassian Software Systems Pty Ltd" url="http://www.atlassian.com"/>
  </plugin-info>
</atlassian-plugin>
```

Now we are ready to move onto writing some code to make our plugin do something.

Step 2 - Adding the first Build Complete Labeller Module

In this plugin, we want Bamboo to perform a custom action immediately after a build has completed. To do this, we write a Build Complete Action Module. You can see all the available Bamboo module types here.

To start things off, we would like to keep our custom action pretty simple and make sure things work. Our first cut of the BuildLabeller will simply label the build as "out_of_memory" if the "OutOfMemoryError" was found in the logs.

```
public class BuildLabeller implements CustomBuildCompleteAction {
    private static final Logger log = Logger.getLogger(BuildLabeller.class);

    /**
     * Dependency on labelManager. Bamboo's Spring IOC will automatically inject manager into this class via the setter.
     */
    private LabelManager labelManager;

    /**
     * This action will run after a build has completed.
     * The build will be labelled with "out_of_memory" if the "OutOfMemoryError" was detected in the logs.
     */
```
* @param build
* @param buildResults
*/
public void run(Build build, BuildResults buildResults)
{
    List logs = buildResults.getBuildLog();
    for (Iterator iterator = logs.iterator(); iterator.hasNext();)
    {
        SimpleLogEntry log = (SimpleLogEntry) iterator.next();
        if(log.getLog().indexOf("OutOfMemoryError") != -1)
        {
            getLabelManager().addLabel("out_of_memory", buildResults, null);
            break;
        }
    }
}
/**
 * This method is used to validate a build configuration for a build plan
 * This is used if the CustomBuildCompleteAction needs to have configuration stored
 * against the build plan.
 * @param buildConfiguration
 * @return
 */
public ErrorCollection validate(BuildConfiguration buildConfiguration)
{
    return null;
}
-----------------------------------------------------------------------------------------------
Getters & Setters
public LabelManager getLabelManager()
{
    return labelManager;
}
public void setLabelManager(LabelManager labelManager)
{
    this.labelManager = labelManager;
}

Our custom module must implement the CustomBuildCompleteAction interface, which defines a run method and a validate method.

The run method is what gets called when a build completes. Our run method in this plugin is fairly simple. It loops through each line of the build logs and searches for the exact string - "OutOfMemoryError". Once found, it stops looping and labels the build.

In the run method, we make use of the services of the LabelManager (a dependency), which is responsible for tagging of a build. Dependencies in plugins are automatically handled by Bamboo Spring container. As long as the plugin has the correct "setter" method, the dependency will be automatically injected.

You may notice that the other method defined by the CustomBuildCompleteAction interface: validate currently doesn't do anything. We will return to this in the next tutorial.

Step 3 - Registering the Build Complete Labeller Module

Once you have written your labeller module, we must now register the plugin module into our plugin
Step 4 - Build and Test

That's it. We now need to test our code. To do this, we can build our plugin by returning to the command line in the root directory of your source directory, and run the command: `mvn package`. This created a `bamboo-labeller-plugin-1-1.0.jar`. We can now drop this into Bamboo (/webapp/WEB-INF/lib), and see it in action.

Here is what our plugin produced after we ran a build with a `OutOfMemoryError`:

![Bamboo Build Result](image)

Next Steps

So we have made our first basic plugin. Right now, it's not very configurable, and runs for every build. In the next tutorial, we will introduce configurability to our Labeller.
Tutorial 2 - Configurable Regex Labeller

This page last changed on Mar 01, 2007 by mike@atlassian.com.

In the previous tutorial, we have made our plugin label the build whenever the logs had the words "OutOfMemoryError". This, however, is not very useful for the other builds which don't have this memory problem. Also, it is not very useful to only be able to tag with "out_of_memory". In this tutorial, we will extend on the plugin module so that we can configure when to label, and what to label a build with.

The source code to the plugin used in this tutorial is available on the Atlassian public source repository. You can check out the source code here.

Step 1 - Adding configuration views

To do this, we must first add the views for configuring the labeller. The BuildCompleteAction module type comes with the capability to accept Freemaker templates which allows you to edit and view custom configuration in the Build Plan Configuration page, under the Post Action tab.

The Freemaker template to edit our Labeller configuration is below (regexLabellerEdit.ftl):

```freemaker
[@ui.bambooSection title='Pattern matching labelling.']
[@ww.textfield name='custom.bamboo.labeller.regex' label='Regex Pattern'
   description='The regular expression for which to match the log files on.' ]
[@ww.textfield name='custom.bamboo.labeller.label' label='Label(s)'
   description='The label(s) for the build if it matches the specified regex pattern.' ]
[/@ui.bambooSection ]
```

Here, we define a section with a title 'Pattern matching labelling.' Inside our configuration section are two text fields, one for the regex expression for matching against the logs, and one for the label(s) that we want to tag a build with if the regex expression matches.

We have named our two text fields `custom.bamboo.labeller.regex` and `custom.bamboo.labeller.label`. These are the keys to your custom configuration property stored in Bamboo.

Please note that these keys must start with "custom." for Bamboo to recognize and store within the plan's configuration. You may also notice that the keys are "namespaced". This is a good idea to prevent a clash of custom configuration properties.

Display Configuration View

We also define a Freemaker view for viewing the configuration (read-only). The display configuration view is below (regexLabellerView.ftl):

```freemaker
[#if build.buildDefinition.customConfiguration.get('custom.bamboo.labeller.regex')?has_content
```
Here we simply build display the configuration by retrieving your custom properties via the same keys we used in the edit view.

Registering the views in the Plugin Descriptor

We need to register these two Freemarker templates as part of our BuildCompleteAction module. We do this by adding <resource> tags with the file path of the templates within the module descriptor definition.

```xml
<buildCompleteAction key="labeller" name="Build Labeller class=com.atlassian.bamboo.plugins.labeller.BuildLabeller">
    <description>An automatic labelling plugin.</description>
    <resource type="freemarker" name="edit" location="templates/buildCompleteAction/regexLabellerEdit.ftl"/>
    <resource type="freemarker" name="view" location="templates/buildCompleteAction/regexLabellerView.ftl"/>
</buildCompleteAction>
```

Once that's done, we can see the templates in action.

Under the edit configuration page:

<table>
<thead>
<tr>
<th>Pattern matching labelling.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Regex Pattern:</strong></td>
</tr>
<tr>
<td><strong>Label(s):</strong></td>
</tr>
</tbody>
</table>

The regular expression for which to match the log files.

The label(s) for the build if it matches the specified regex pattern.

And under the view configuration page:

<table>
<thead>
<tr>
<th>Pattern Matching Labelling</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Regex Pattern:</strong></td>
</tr>
<tr>
<td><strong>Labels:</strong></td>
</tr>
</tbody>
</table>

Step 2 - Adding validation

Inserting the templates has allowed us to view and edit custom plan configuration properties. However, we should validate the input we provide for the BuildLabeller, to catch invalid labels or regex patterns.

This is where we use the `validate` method within our BuildLabeller class, which we have previously
left to return null in the first tutorial. Bamboo will run this validate method before trying to save custom configuration properties.

```java
/**
 * This method is used to validate a build configuration for a build plan
 * This is used if the CustomBuildCompleteAction needs to have configuration stored against the build plan.
 * @param buildConfiguration
 * @return
 */
public ErrorCollection validate(BuildConfiguration buildConfiguration)
{
    // Check the label values to see if they have any invalid characters
    ErrorCollection errors = new SimpleErrorCollection();
    String labelInput = buildConfiguration.getString("custom.bamboo.labeller.label");
    List labels = LabelParser.split(labelInput);
    for (Iterator iterator = labels.iterator(); iterator.hasNext();)
    {
        String label = (String) iterator.next();
        boolean validLabel = LabelParser.isValidLabelName(label);
        if (!validLabel)
        {
            errors.addError("custom.bamboo.labeller.label", label + " contains invalid characters " + LabelParser.getInvalidCharactersAsString());
        }
    }
    // See if the regex is a valid one by trying to compile it
    String regex = buildConfiguration.getString("custom.bamboo.labeller.regex");
    try
    {
        Pattern.compile(regex);
    }
    catch(PatternSyntaxException e)
    {
        errors.addError("custom.bamboo.labeller.regex", regex + " is not a valid regex pattern.");
    }
    return errors;
}
```

The `BuildConfiguration` object passed to the validation method is the in-memory version of the build plan configuration. You can get your custom property by simply calling `getString` on the object, providing the custom property key that you used in the Freemarker templates.

### Step 3 - Applying the configuration

At this stage, we can edit, validate, and view our custom configuration for this plugin module. We now need to modify our original `run` method within the `BuildLabeller` to read the custom configuration properties.

```java
/**
 * This action will run after a build has completed.
 * The build will be tagged with a specified set of labels if the logs matches the specified regex pattern.
 * @param build
 * @param buildResults
 */
```
public void run(Build build, BuildResults buildResults)
{
    // grab the custom configuration object
    Map customConfiguration = build.getBuildDefinition().getCustomConfiguration();
    if (customConfiguration != null)
    {
        if (customConfiguration.containsKey("custom.bamboo.labeller.label"))
        {
            List logs = buildResults.getBuildLog();

            String pattern = (String) customConfiguration.get("custom.bamboo.labeller.regex");
            Pattern regexPattern = Pattern.compile(pattern);

            // Go through the logs
            for (Iterator iterator = logs.iterator(); iterator.hasNext();)
            {
                SimpleLogEntry log = (SimpleLogEntry) iterator.next();
                Matcher matcher = regexPattern.matcher(log.getLog());

                // Use a matcher to see if the logs contained the specified regex
                if (matcher.find())
                {
                    String labelsInput = (String) customConfiguration.get("custom.bamboo.labeller.label");

                    // Our configuration also allows for multiple labels.
                    List labels = LabelParser.split(labelsInput);
                    for (Iterator iterator2 = labels.iterator(); iterator2.hasNext();)
                    {
                        String label = (String) iterator2.next();
                        getLabelManager().addLabel(label, buildResults, null);
                    }
                    break;
                }
            }
        }
    }
}