Space Details

<table>
<thead>
<tr>
<th>Key</th>
<th>BAMBOO10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name:</td>
<td>Bamboo 1.0</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>admin (May 04, 2007)</td>
</tr>
</tbody>
</table>

Available Pages

- Bamboo 1.0 Documentation Home 🏡
  - Bamboo 1.0 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
    - 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
      - 5.1 Creating a User
      - 5.2 Deleting a User
      - 5.3 Creating a Group
• 5.4 Deleting a Group
• 5.5 Adding Users to and removing them from Groups
• 5.6 Granting Administration Privileges to a User
• 5.7 Enabling or disabling Public Signup
• 5.8 Enabling or disabling Anonymous Access
• 5.9 Working with External User Repositories
  • 5.9.1 Integrating Bamboo with Crowd
• 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 1.0 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 1.0 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 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 Editing your User Details
  • 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
  • label
  • plan
  • project
  • triggering
• 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 Upgrade Guide
- Bamboo 1.0 Development Hub
- Bamboo API
- Bamboo Plugin Developer's Guide
  - Build Complete Action Module
  - Build Processor Module
  - Builder Plugin Module
• Getting Started
• Index Reader Module
• Post Build Index Writer 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.

<table>
<thead>
<tr>
<th>Installing/upgrading Bamboo 1.0</th>
<th>Using/administering Bamboo 1.0</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>

Recently Updated

by Rosie Jameson (17 mins ago)
Bamboo 1.0 Administrator's Guide
by Rosie Jameson (04 May)
Bamboo 1.0 Documentation Home

This documentation applies to Bamboo version 1.0.x. The latest Bamboo documentation is available via this link.
Bamboo 1.0 Administrator’s Guide

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

Download

You can download the Bamboo Administrator’s Guide in PDF, HTML or XML formats.

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 1.0 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

- 5.1 Creating a User
- 5.2 Deleting a User
- 5.3 Creating a Group
- 5.4 Deleting a Group
- 5.5 Adding Users to and removing them from Groups
- 5.6 Granting Administration Privileges to a User
- 5.7 Enabling or disabling Public Signup
- 5.8 Enabling or disabling Anonymous Access
- 5.9 Working with External User Repositories
  - 5.9.1 Integrating Bamboo with Crowd

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
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

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

### 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

This page last changed on May 04, 2007 by admin.

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 1.0 Documentation Home
1.2 Creating a Plan

This page last changed on May 04, 2007 by admin.

To create a new plan,

1. Click the 'Create Plan' link in the top navigation bar.
2. Enter the required information in the 5 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 5 screens as described below. On screens 2-5, 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'

Unable to render content due to system error: null

- 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.

Note that 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'
2. 'Source Repository':

- '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:
  - **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.
  - **Subversion**:
    - 'Repository URL' — The location of subversion repository (e.g. `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.
  - **Perforce**:
    - '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 Perforce depot that contains the source-code files for this plan. 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.
- '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'

Unable to render content due to system error: null

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 test (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.
- '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'

Unable to render content due to system error: null

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. 'Post Actions'

Unable to render content due to system error: null

5. 'Post Actions':

- 'When to send notification' — Please see 6.1 Enabling or disabling Notifications for a Plan. Depending on the relative importance of this plan, you can choose 'Only send notifications for failed builds and associated fixed' or 'All builds'. If you do not want this plan to generate any notifications, simply deselect both the 'Send an email' and the 'Send a message to an IM server' check-boxes.
  - Select 'Send an email' if people need to receive email notifications about this plan's build results (note: first please see 6.2 Configuring Bamboo to send SMTP Email).
° Select 'Send a message to an IM server' if people need to receive IM notifications about this plan's build results (note: first please see 6.3 Configuring Bamboo to use Instant Messaging (IM)).

° '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 1.0 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. Unable to render content due to system error: null
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 1.0 Documentation Home
1.4 Specifying Labels for a Plan's Build Results

This page last changed on May 04, 2007 by admin.

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 1.0 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'

Unable to render content due to system error: null

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

Unable to render content due to system error: null

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 1.0 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. Unable to render content due to system error: null

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.

Unable to render content due to system error: null

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
1.6.1 Deleting a Build Result

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. Unable to render content due to system error: null

Unable to render content due to system error: null

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 1.0 Documentation Home
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
2.1 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).

Unable to render content due to system error: null

RELATED TOPICS

- 2.1 Configuring a new Builder
- 2.2 Configuring a new JDK

Bamboo 1.0 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.

Unable to render content due to system error: null

RELATED TOPICS

- 2.1 Configuring a new Builder
- 2.2 Configuring a new JDK

Bamboo 1.0 Documentation Home
03. Triggering a Build

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

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 particlarly 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 1.0 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.

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 1.0 Documentation Home
3.2.1 Polling the Repository for Code Changes

Unable to render content due to system error: null

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

This page last changed on May 04, 2007 by admin.

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 CVSPROJECT 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.

Unable to render content due to system error: null

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 1.0 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 1.0 Documentation Home
3.3.1 Scheduling a Single Daily Build

Unable to render content due to system error: null

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 1.0 Documentation Home
3.3.2 Specifying a Cron-based Schedule

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.

Unable to render content due to system error: null

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 1.0 Documentation Home
3.4 Triggering a Build when another Build finishes

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.

Unable to render content due to system error: null

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 1.0 Documentation Home
3.5 Triggering a Build Manually

To start a manual build,

1. Click 'Home' to go to the [Dashboard](#).
2. Unable to render content due to system error: null

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'.

Unable to render content due to system error: null

**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 1.0 Documentation Home](#)
04. Managing Build Queues

This page last changed on Feb 11, 2007 by rosie@atlassian.com.
4.1 Creating a Build Queue

This page last changed on May 04, 2007 by admin.

One default queue is created when you install Bamboo.

Unable to render content due to system error: null

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 1.0 Documentation Home
4.2 Assigning a Plan to a Build 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.

Unable to render content due to system error: null

⚠️ 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.

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 1.0 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'

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 1.0 Documentation Home
4.4 Stopping an Active Build

This page last changed on May 04, 2007 by admin.
4.5 Disabling or deleting a Build Queue
5. Managing Users and Security

- 5.1 Creating a User
- 5.2 Deleting a User
- 5.3 Creating a Group
- 5.4 Deleting a Group
- 5.5 Adding Users to and removing them from Groups
- 5.6 Granting Administration Privileges to a User
- 5.7 Enabling or disabling Public Signup
- 5.8 Enabling or disabling Anonymous Access
- 5.9 Working with External User Repositories
  - 5.9.1 Integrating Bamboo with Crowd
5.1 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.

Unable to render content due to system error: null

RELATED TOPICS

- 5.1 Creating a User
- 5.2 Deleting a User
- 5.3 Creating a Group
- 5.4 Deleting a Group
- 5.5 Adding Users to and removing them from Groups
- 5.6 Granting Administration Privileges to a User
- 5.7 Enabling or disabling Public Signup
- 5.8 Enabling or disabling Anonymous Access
- 5.9 Working with External User Repositories
  - 5.9.1 Integrating Bamboo with Crowd

Bamboo 1.0 Documentation Home
5.2 Deleting a User

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.

Unable to render content due to system error: null

RELATED TOPICS

- 5.1 Creating a User
- 5.2 Deleting a User
- 5.3 Creating a Group
- 5.4 Deleting a Group
- 5.5 Adding Users to and removing them from Groups
- 5.6 Granting Administration Privileges to a User
- 5.7 Enabling or disabling Public Signup
- 5.8 Enabling or disabling Anonymous Access
- 5.9 Working with External User Repositories
  - 5.9.1 Integrating Bamboo with Crowd

Bamboo 1.0 Documentation Home
5.3 Creating a Group

This page last changed on May 04, 2007 by admin.

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

!Bamboo-group-create.PNG!

Related Topics

- 5.1 Creating a User
- 5.2 Deleting a User
- 5.3 Creating a Group
- 5.4 Deleting a Group
- 5.5 Adding Users to and removing them from Groups
- 5.6 Granting Administration Privileges to a User
- 5.7 Enabling or disabling Public Signup
- 5.8 Enabling or disabling Anonymous Access
- 5.9 Working with External User Repositories
  - 5.9.1 Integrating Bamboo with Crowd

Bamboo 1.0 Documentation Home
5.4 Deleting a Group

This page last changed on May 04, 2007 by admin.

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

Unable to render content due to system error: null

RELATED TOPICS

- 5.1 Creating a User
- 5.2 Deleting a User
- 5.3 Creating a Group
- 5.4 Deleting a Group
- 5.5 Adding Users to and removing them from Groups
- 5.6 Granting Administration Privileges to a User
- 5.7 Enabling or disabling Public Signup
- 5.8 Enabling or disabling Anonymous Access
- 5.9 Working with External User Repositories
  - 5.9.1 Integrating Bamboo with Crowd

Bamboo 1.0 Documentation Home
5.5 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

Unable to render content due to system error: null

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
5.1 Creating a User
5.2 Deleting a User
5.3 Creating a Group
5.4 Deleting a Group
5.5 Adding Users to and removing them from Groups
5.6 Granting Administration Privileges to a User
5.7 Enabling or disabling Public Signup
5.8 Enabling or disabling Anonymous Access
5.9 Working with External User Repositories
   5.9.1 Integrating Bamboo with Crowd

Bamboo 1.0 Documentation Home
5.6 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 1.0 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

- 5.1 Creating a User
- 5.2 Deleting a User
- 5.3 Creating a Group
- 5.4 Deleting a Group
- 5.5 Adding Users to and removing them from Groups
- 5.6 Granting Administration Privileges to a User
- 5.7 Enabling or disabling Public Signup
- 5.8 Enabling or disabling Anonymous Access
- 5.9 Working with External User Repositories
  - 5.9.1 Integrating Bamboo with Crowd

Bamboo 1.0 Documentation Home
5.7 Enabling or disabling Public Signup

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.

Unable to render content due to system error: null

RELATED TOPICS

- 5.1 Creating a User
- 5.2 Deleting a User
- 5.3 Creating a Group
- 5.4 Deleting a Group
- 5.5 Adding Users to and removing them from Groups
- 5.6 Granting Administration Privileges to a User
- 5.7 Enabling or disabling Public Signup
- 5.8 Enabling or disabling Anonymous Access
- 5.9 Working with External User Repositories
  - 5.9.1 Integrating Bamboo with Crowd

Bamboo 1.0 Documentation Home
5.8 Enabling or disabling Anonymous Access

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 1.0 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

- 5.1 Creating a User
- 5.2 Deleting a User
- 5.3 Creating a Group
- 5.4 Deleting a Group
- 5.5 Adding Users to and removing them from Groups
- 5.6 Granting Administration Privileges to a User
- 5.7 Enabling or disabling Public Signup
- 5.8 Enabling or disabling Anonymous Access
- 5.9 Working with External User Repositories
  - 5.9.1 Integrating Bamboo with Crowd

Bamboo 1.0 Documentation Home
5.9 Working with External User Repositories

This page last changed on May 04, 2007 by admin.
5.9.1 Integrating Bamboo with Crowd

This page last changed on May 04, 2007 by admin.

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

- 5.1 Creating a User
- 5.2 Deleting a User
- 5.3 Creating a Group
- 5.4 Deleting a Group
- 5.5 Adding Users to and removing them from Groups
- 5.6 Granting Administration Privileges to a User
- 5.7 Enabling or disabling Public Signup
- 5.8 Enabling or disabling Anonymous Access
- 5.9 Working with External User Repositories
  - 5.9.1 Integrating Bamboo with Crowd

Bamboo 1.0 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, i.e.:

- Send notifications for all builds — this option is recommended for any plans for which it is critical that people are always informed about the latest build activity.
- Only send notifications for failed builds (and associated fixed builds) — this is recommended if you are concerned about sending people too many notifications.

You can specify different recipients and different circumstances for each plan. 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 any plans can send notifications.

Unable to render content due to system error: null

To disable notifications for a plan,

Follow steps 1-6 above, then:

1. To disable email notifications, deselect the ‘Send an email’ check-box.
2. To disable IM notifications, deselect the ‘Send a message to an IM server’ check-box.
3. Click the ‘Save’ button.

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 1.0 Documentation Home
6.2 Configuring Bamboo to send SMTP Email

This page last changed on May 04, 2007 by admin.
6.3 Configuring Bamboo to use Instant Messaging (IM)

This page last changed on May 04, 2007 by admin.

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'

Unable to render content due to system error: null

⚠️ Next step

Now that you have configured Bamboo's IM capability, you can specify notifications for a plan.
- 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 1.0 Documentation Home
6.3.1 Configuring Bamboo to use Google Talk for Instant Messaging

This page last changed on May 04, 2007 by admin.

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. atlassianbambou NOT atlassianbambou@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. atlassianbambou@gmail.com NOT atlassianbambou) 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 1.0 Documentation Home
07. Managing Data and Backups

This page last changed on Feb 06, 2007 by rosie@atlassian.com.
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 1.0 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.

Unable to render content due to system error: null

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 1.0 Documentation Home
7.3 Importing Data from Backup

This page last changed on May 04, 2007 by admin.

⚠️ 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.

Screenshot: Import
!Bamboo-import.PNG!

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 1.0 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'
!Bamboo-buildexpiry.PNG!

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 1.0 Documentation Home
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 04, 2007 by admin.

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 1.0 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/builfs/ sub-directories.**)
- **webapp/** *— This directory contains all the Bamboo server application files.
  - **webapp/plugins/** *— This directory contains the Bamboo plugins.
- **webapp/WEB-INF/lib/** *— This directory is used when deploying Bamboo plugins.
- **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 GZIIP Compression**
- **8.7 Enabling Bamboo's Remote API**

**Bamboo 1.0 Documentation Home**
8.1.1 Specifying Bamboo's Working Directory

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 1.0 Documentation Home
8.2 Viewing Bamboo’s System Information

This page last changed on May 04, 2007 by admin.
8.3 Updating your Bamboo License Details
8.4 Specifying Bamboo's URL

This page last changed on May 04, 2007 by admin.

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 1.0 Documentation Home
8.5 Enabling JIRA integration

This page last changed on May 04, 2007 by admin.

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 1.0 Documentation Home
8.6 Enabling GZIP Compression

This page last changed on May 04, 2007 by admin.

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 1.0 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 1.0 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 1.0 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 1.0 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 1.0 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

This page last changed on Feb 27, 2007 by rosie@atlassian.com.

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.
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

   <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/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/

   <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-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

```
<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>
```

Unable to render content due to system error: null
Plan Summary Graphs

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/](http://opensource.bamboo.atlassian.com/)

```html
<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 1.0 Installation Guide

This page last changed on May 04, 2007 by admin.

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 09, 2007 00:14  Page 92
Running the Setup Wizard

Upgrade Guide
Release Notes

Bamboo 1.0 Documentation Home
Bamboo EAR-WAR Installation Guide

This page last changed on May 04, 2007 by admin.

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](http://docs.oracle.com/javase/6/docs/technotes/guides/visual/index.html).

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](http://host:port/bamboo)

### Step 4. Configure Bamboo

See [Running the Setup Wizard](http://confluence.atlassian.com/display/BAMBOO10/Running+the+Setup+Wizard).

**RELATED TOPICS**

- [Bamboo 1.0 Installation Guide](http://confluence.atlassian.com/display/BAMBOO10/Bamboo+1.0+Installation+Guide)
- [Bamboo 1.0 Documentation Home](http://confluence.atlassian.com/display/BAMBOO10/Bamboo+1.0+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.

- `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:


### 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/.

Step 3. Configuring Bamboo

See Running the Setup Wizard.
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/](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

Unable to render content due to system error: null

- '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.

Unable to render content due to system error: null

Step 3. Server Configuration

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

Unable to render content due to system error: null

Next...

1.1 About Projects, Plans and Builds
1.2 Creating a Plan
Bamboo 1.0 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

Index

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
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 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 Editing your User Details
- 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
<table>
<thead>
<tr>
<th>Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>build queue</td>
</tr>
<tr>
<td>build result</td>
</tr>
<tr>
<td>build telemetry</td>
</tr>
<tr>
<td>builder</td>
</tr>
<tr>
<td>label</td>
</tr>
<tr>
<td>plan</td>
</tr>
<tr>
<td>project</td>
</tr>
<tr>
<td>triggering</td>
</tr>
</tbody>
</table>
01. Getting Started

This page last changed on Feb 05, 2007 by rosie@atlassian.com.

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

This page last changed on May 04, 2007 by admin.
1.2 Viewing Bamboo's Current Activity

Sometimes you may want to see which plans are currently being built, and which plans (if any) are
waiting in a build 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

Unable to render content due to system error: null

- click a plan name (e.g. 'Crowd - Main Build') to view the plan details.
- click the !build_log.gif 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.

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 1.0 Documentation Home
1.3 Viewing your Latest Build Results

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'

Unable to render content due to system error: null

⚠️ 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 1.0 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 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.

Unable to render content due to system error: null

✓ 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

- About Projects and Plans
- About Builds and Build Results
- Adding a Plan to your Favourites
- Removing a Plan from your Favourites
- Bamboo Glossary

Return to Bamboo 1.0 Documentation Home
1.4.1 Adding a Plan to your Favourites

This page last changed on May 04, 2007 by admin.

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: !star_grey.gif!
5. Click the 'My Bamboo' tab.
6. Verify that the plan is now 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 1.0 Documentation Home
1.4.2 Removing a Plan from your Favourites

This page last changed on May 04, 2007 by admin.

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. Unable to render content due to system error: null
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 1.0 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 1.0 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 1.0 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:

Unable to render content due to system error: null

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.

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.

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).

Plans can only be configured by an administrator. For details please see the Bamboo 1.0 Administrator's Guide.
• 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 1.0 Documentation Home
2.3 Viewing a Plan's Activity Log

This page last changed on May 04, 2007 by admin.
03. Working with Build Results

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

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 JIRA Issues for a Build Result
3.1 About Builds and Build Results

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.

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 JIRA Issues for a Build Result

Bamboo Glossary
Return to Bamboo 1.0 Documentation Home
3.2 Viewing a Build Result

This page last changed on May 04, 2007 by admin.
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.

Unable to render content due to system error: null

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 JIRA Issues for a Build Result

Bamboo Glossary
Return to Bamboo 1.0 Documentation Home
3.4 Viewing the Code Changes that triggered a Build

This page last changed on May 04, 2007 by admin.
3.5 Viewing Clover Code-Coverage for a Build Result

This page last changed on May 04, 2007 by admin.
3.6 Viewing JIRA Issues for a Build Result

This page last changed on May 04, 2007 by admin.

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.

Unable to render content due to system error: null

![Warning](https://example.com/warning.png)

This will only be available if Bamboo-JIRA communication has been enabled by your Bamboo administrator. For details please refer to the Bamboo 1.0 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.

Unable to render content due to system error: null

**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 JIRA Issues for a Build Result](#)

**Bamboo Glossary**
Return to [Bamboo 1.0 Documentation Home](#)
04. Working with Tests

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

Unable to render content due to system error: null

Screenshot 2: Successful Tests for a Build

!bamboo-buildresult-successfultests.PNG!

ℹ️ 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 1.0 Documentation Home
4.2 Viewing a Test's History

This page last changed on May 04, 2007 by admin.

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. Unable to render content due to system error: null
5. The 'Test History' will be displayed as shown below.

Screenshot: Test History

Unable to render content due to system error: null

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 1.0 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
!bamboo-test1.png

To view a test's history, click the test name.

Screenshot 2: Top 10 Longest to Fix Tests

Unable to render content due to system error: null

Unable to render content due to system error: null

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 1.0 Documentation Home
5.1 Viewing Build Statistics for a Plan

This page last changed on May 04, 2007 by admin.

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'

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.

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:

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 1.0 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 1.0 Administrator's Guide](https://confluence.atlassian.com/display/BB/1.0+Administrator%27s+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.

Screenshot: 'Report Parameters--Build Plans'
!bamboo-report-parameters.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
- 'Percentage of Successful Builds per Plan' Report
- 'Time to Fix per Plan' Report

Bamboo Glossary
Return to Bamboo 1.0 Documentation Home
'Build Duration per Plan' Report

This page last changed on May 04, 2007 by admin.

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'

Unable to render content due to system error: null

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 1.0 Documentation Home
'Clover Code Coverage per Plan' Report

This page last changed on May 04, 2007 by admin.

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 1.0 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'

Unable to render content due to system error: null

## 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 1.0 Documentation Home
'Clover Lines of Code 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.

Unable to render content due to system error: null

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 1.0 Documentation Home
'Number of Build Failures per Plan' Report

This page last changed on May 04, 2007 by admin.

You can choose the plan(s) and time period on which you want to report.

Sample Report: 'Number of Build Failures per Plan'

!bamboo-report-buildfailures.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
Return to Bamboo 1.0 Documentation Home
'Number of Tests per Plan' Report

This page last changed on May 04, 2007 by admin.

You can choose the plan(s) and time period on which you want to report.

Sample Report: 'Number of Tests per Plan'

!bamboo-report-numbertests.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
Return to Bamboo 1.0 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'

Unable to render content due to system error: null

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 1.0 Documentation Home
'Time to Fix per Plan' Report

You can choose the plan(s) and time period on which you want to report.

Sample Report: 'Time to Fix per Plan'

Unable to render content due to system error: null

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 1.0 Documentation Home
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).

Unable to render content due to system error: null

✔ 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 1.0 Documentation Home
6.2 Viewing Build Results for an Author

This page last changed on May 04, 2007 by admin.
6.3 Generating Reports on selected 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 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 1.0 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'

Unable to render content due to system error: null

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 1.0 Documentation Home
'Build Activity per Author' Report

This page last changed on May 04, 2007 by admin.
'Number of Build Failures per Author' Report

This page last changed on May 04, 2007 by admin.

You can select the author(s) on whom you want to report.

Sample Report: 'Number of Build Failures per Author'

!bamboo-authors-report-buildfailures.PNG!

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 1.0 Documentation Home
'Number of Builds Broken per Author' Report

This page last changed on May 04, 2007 by admin.

You can select the author(s) on whom you want to report.

Sample Report: 'Number of Builds Broken per Author'

Unable to render content due to system error: null

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 1.0 Documentation Home
'Number of Builds Fixed per Author' Report

This page last changed on May 04, 2007 by admin.

You can select the author(s) on whom you want to report.

Sample Report: 'Number of Builds Fixed per Author'
!bamboo-authors-report-fixed.PNG!

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 1.0 Documentation Home
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 1.0 Documentation Home
7.2 Commenting about a Build Result

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.

Unable to render content due to system error: null

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 1.0 Documentation Home
7.3 Viewing Comments about a Build Result

To view comments about a particular build result,

1. Unable to render content due to system error: null

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. Unable to render content due to system error: null

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 1.0 Documentation Home
7.4 Viewing Code Check-in Comments

This page last changed on May 04, 2007 by admin.
08. Working with Labels

This page last changed on Feb 04, 2007 by rosie@atlassian.com.

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 1.0 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. Unable to render content due to system error: null
3. Unable to render content due to system error: null
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 1.0 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. Unable to render content due to system error: null
3. Unable to render content due to system error: null
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 1.0 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 1.0 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'

Unable to render content due to system error: null

✔ 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 1.0 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: !09. Subscribing to RSS Feeds^bamboo-rss.PNG!
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 1.0 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: !09. Subscribing to RSS Feeds^bamboo-rss.PNG!
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 1.0 Documentation Home

Return to Bamboo 1.0 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: !09. Subscribing to RSS Feeds^bamboo-rss.PNG!
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 1.0 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: !09. Subscribing to RSS Feeds^bamboo-rss.PNG!
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 1.0 Documentation Home
9.5 Subscribing to an RSS Feed for Labelled Build Results

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: !bamboo-rss-label.PNG!
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 1.0 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 04, 2007 by admin.
10.2 Labelling a Build Result via IM

This page last changed on May 04, 2007 by admin.
10.3 Commenting about a Build Result via IM

This page last changed on May 04, 2007 by admin.
11. Editing your User Profile

11. Editing your User Profile

- 11.1 Changing your Password
- 11.2 Editing your User Details
- 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 Editing your User Details
- 11.3 Associating your Author Name with your User Profile

Bamboo Glossary
Return to Bamboo 1.0 Documentation Home
11.2 Editing your User Details

This page last changed on May 04, 2007 by admin.

To edit your User Details,

1. Click the 'Profile' link in the top right corner of the screen. This will display the 'User Profile' screen.
2. Unable to render content due to system error: null
3. Update your details as appropriate:
   'Email' — the address at which you will receive emails such as 'Forgot password'.
   'Jabber address' — the address at which you will receive Instant Messaging (IM) notifications about build results, if IM notifications are enabled for a plan.
   'Full Name' — the name that is displayed in build results and Author reports.
   'Source Repository Alias' — your repository login name (Author Name). For details see Associating your Author Name with your User Profile.
4. Click the 'Save' button.

⚠️ Only Bamboo administrators can enable notifications for a plan. For details please see the Bamboo 1.0 Administrator's Guide.

RELATED TOPICS

- 11.1 Changing your Password
- 11.2 Editing your User Details
- 11.3 Associating your Author Name with your User Profile

Bamboo Glossary
Return to Bamboo 1.0 Documentation Home
11.3 Associating your Author Name with your User Profile

This page last changed on May 04, 2007 by admin.

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. Unable to render content due to system error: null
4. Click the 'Save' button.

RELATED TOPICS

- 11.1 Changing your Password
- 11.2 Editing your User Details
- 11.3 Associating your Author Name with your User Profile

Bamboo Glossary
Return to Bamboo 1.0 Documentation Home
Appendix A. Bamboo Glossary

This page last changed on May 04, 2007 by admin.

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.
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.

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.

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.

Return to Bamboo 1.0 Documentation Home
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.
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**

This page last changed on Jan 29, 2007 by rosie@atlassian.com.

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.
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 Feb 05, 2007 by rosie@atlassian.com.
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'.

This page last changed on Jan 28, 2007 by rosie@atlassian.com.
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.
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

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

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".
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.
Bamboo Knowledge Base

Answers to commonly raised questions about configuring and using Bamboo:

**Installation FAQ**

**Integration FAQ**

**Usage FAQ**

- 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.

Do you have a question, or need help with Bamboo? Please create a support request.
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 0.1 Release Notes**

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

Unable to render {include} Couldn't find a page to include called: __newreleaseBamboo

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.

Unable to render \{include\} Couldn't find a page to include called: __newreleaseBamboo

⚠️ 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

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

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 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.

Unable to render {include} Couldn't find a page to include called: __newreleaseBamboo

⚠️ 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 summmary 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>Assigned</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>BAM-221</td>
<td>View CVS URL is optional - mark as such</td>
<td>Ben Kuo</td>
<td>Riaz Khanmohamed</td>
<td></td>
<td></td>
<td>FIXED</td>
<td>Resolved</td>
<td>Sep 14, 2006</td>
<td>Sep 15, 2006</td>
<td></td>
</tr>
<tr>
<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></td>
<td>FIXED</td>
<td>Resolved</td>
<td>Sep 14, 2006</td>
<td>Oct 05, 2006</td>
<td></td>
</tr>
<tr>
<td>BAM-213</td>
<td>Grammar on Log page</td>
<td>Ben Kuo</td>
<td>Riaz Khanmohamed</td>
<td></td>
<td></td>
<td>FIXED</td>
<td>Resolved</td>
<td>Sep 14, 2006</td>
<td>Sep 14, 2006</td>
<td></td>
</tr>
<tr>
<td>BAM-211</td>
<td>Keep refresh option for logs</td>
<td>Ben Kuo</td>
<td>Riaz Khanmohamed</td>
<td></td>
<td></td>
<td>FIXED</td>
<td>Resolved</td>
<td>Sep 14, 2006</td>
<td>Sep 15, 2006</td>
<td></td>
</tr>
<tr>
<td>BAM-210</td>
<td>Log page refreshing</td>
<td>Ben Kuo</td>
<td>Riaz Khanmohamed</td>
<td></td>
<td></td>
<td>FIXED</td>
<td>Resolved</td>
<td>Sep 14, 2006</td>
<td>Sep 15, 2006</td>
<td></td>
</tr>
<tr>
<td>BAM-207</td>
<td>WebUrl link should link to the diff of the file</td>
<td>Mark Chaimungkalanont</td>
<td>Mark</td>
<td></td>
<td></td>
<td>FIXED</td>
<td>Resolved</td>
<td>Sep 14, 2006</td>
<td>Sep 14, 2006</td>
<td></td>
</tr>
<tr>
<td>BAM-196</td>
<td>Adding Visual Studio as</td>
<td>Mark MattyJ Chaimungkalanont</td>
<td></td>
<td></td>
<td></td>
<td>FIXED</td>
<td>Resolved</td>
<td>Sep 06, 2006</td>
<td>Sep 06, 2006</td>
<td></td>
</tr>
</tbody>
</table>

[Atlassian JIRA](#) (11 issues)
<table>
<thead>
<tr>
<th>Issue</th>
<th>Description</th>
<th>Assignee 1</th>
<th>Assignee 2</th>
<th>Resolution</th>
<th>Start Date</th>
<th>End Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>BAM-194</td>
<td>Implement Lucene for indexing of data</td>
<td>Edwin Wong</td>
<td>Edwin Wong</td>
<td>FIXED</td>
<td>Sep 05, 2006</td>
<td>Sep 19, 2006</td>
</tr>
<tr>
<td>BAM-161</td>
<td>Fix grammar on build log screen</td>
<td>Ben Kuo</td>
<td>Matt Ryall</td>
<td>FIXED</td>
<td>Aug 25, 2006</td>
<td>Sep 14, 2006</td>
</tr>
<tr>
<td>BAM-74</td>
<td>Cloning projects &amp; the changeDetection field</td>
<td>Ben Kuo</td>
<td>Ben Kuo</td>
<td>FIXED</td>
<td>Jul 18, 2006</td>
<td>Jan 21, 2007</td>
</tr>
</tbody>
</table>
Bamboo 0.4 Upgrade Guide

This page last changed on Jan 17, 2007 by rosie@atlassian.com.
Bamboo 0.5 Release Notes

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

Unable to render {include} Couldn't find a page to include called: __newreleaseBamboo

⚠ 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.

Unable to render content due to system error: null

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?

Unable to render content due to system error: null

Revised Project Editing

Unable to render content due to system error: null

Screenshots

Unable to render content due to system error: null

Other updates & 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-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>Chaimungkalanont</td>
</tr>
<tr>
<td>BAM-245</td>
<td>Build Summary Page improvements Bread Crumbs</td>
<td></td>
<td>Resolved</td>
<td>Mark</td>
</tr>
<tr>
<td>BAM-244</td>
<td>properties file location incorrect</td>
<td></td>
<td>Resolved</td>
<td>Chaimungkalanont</td>
</tr>
<tr>
<td>BAM-242</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 &quot;Topup EJ Bs&quot;</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>-------</td>
<td>------------------------------------------------------------</td>
<td>---------</td>
<td>------------------</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.

Unable to render {include} Couldn't find a page to include called: __newreleaseBamboo

⚠ 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

Unable to render `<include>` Couldn't find a page to include called: __newreleaseBamboo

⚠️ Upgrading? Please see the Bamboo 0.6 Upgrade Guide.

New in Release 0.6

JIRA issue integration

Unable to render content due to system error: null

Authors

Unable to render content due to system error: null

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.

ℹ️ 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>Issue</td>
<td>Description</td>
<td>Status</td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>------------------------------------------------------------------------------</td>
<td>--------</td>
<td></td>
</tr>
<tr>
<td>BAM-376</td>
<td>Confirmation message not given when deleting multiple builds</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-368</td>
<td>Inconsistencies on case-sensitive filesystems</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-361</td>
<td>Perforce provider stops working after Bamboo restart</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-353</td>
<td>default base url is invalid</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-351</td>
<td>build timer incorrect</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-349</td>
<td>side menubar issues</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-348</td>
<td>RSS feed link invalid</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-344</td>
<td>create build link wrong on dash</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-343</td>
<td>Build Queues should be able to choose which projects it would accept</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-341</td>
<td>Cannot change version repositories</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-333</td>
<td>An issue view page for Build Results</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-332</td>
<td>Retrieve issue information through RPC from JIRA</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-331</td>
<td>Setup JIRA instance administration</td>
<td>Resolved</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 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 DecimalFormat &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>
</tbody>
</table>
**BAM-208**
Build table doesn't always refresh correctly
Resolved

**BAM-314**
Double slash ("//") in link
Resolved

**BAM-290**
Bamboo checklist page doesn't give example of bamboo.home property
Resolved

**BAM-271**
Error page should link back to the home page
Resolved
Bamboo 0.7 Release Notes

Unable to render {include} Couldn't find a page to include called: _newreleaseBamboo

⚠ Upgrading? Please see the Bamboo 0.7 Upgrade Guide.

New in Release 0.7

Cross Build Reports

Unable to render content due to system error: null

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.

Unable to render content due to system error: null

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.

ℹ️ 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</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>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-7</td>
<td>Use project IDs instead of names in all URLs</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-469</td>
<td>Build Reports</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-456</td>
<td>Code change filenames from CVS branch have &quot;Attic&quot; in the path</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-454</td>
<td>Improve page titles</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-453</td>
<td>Use profile names on View Authors page</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-452</td>
<td>Log the user who manually initiated or cancelled a build</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-446</td>
<td>New CVS builds build twice</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-444</td>
<td>Re-adding deleted builds (with same name/key) throws 500 error</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-438</td>
<td>Forgot Password email includes context path twice</td>
<td>Resolved</td>
<td></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</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>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>
Bamboo 0.7 Upgrade Guide

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

⚠ 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.

Unable to render {include} Couldn't find a page to include called: __newreleaseBamboo

⚠ 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

Could not generate thumbnail: Attachment file not found

Labels Labels Labels

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

Unable to render content due to system error: null

Bored of labelling everything yourself? You can even get Bamboo to automatically label for you!

Unable to render content due to system error: null

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.
 unemployed 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!

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 build - 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.

Unable to render content due to system error: null

### 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></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-521</td>
<td>Typo in report name</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-519</td>
<td>Ability to select Favourite Builds (Plans)</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-516</td>
<td>Add tooltips to the build results navigator</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-512</td>
<td>Internal Server Error on editing build for Confluence (Postgre, Tomcat)</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-510</td>
<td>UI Improvements</td>
<td></td>
<td>Resolved</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></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-502</td>
<td>Additional Build Strategies</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-494</td>
<td>Wrong title on 'Test Running Duration' graph</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-487</td>
<td>Server error creating build</td>
<td></td>
<td>Resolved</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-existant 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 Expiry 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>BAM-99</td>
<td>XML Export</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-578</td>
<td>Bamboo not displaying</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-528</td>
<td>webapp hangs on refresh occasionally</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-508</td>
<td>Back button when Creating a Build</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-501</td>
<td>Project/Build Terminology Confusing</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-488</td>
<td>configuring artifacts - links that do nothing and navigation</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-457</td>
<td>Manual build does not pick up changes</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-397</td>
<td>link to latest live build</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-371</td>
<td>Poll at specific times</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-143</td>
<td>Update Quartz usage to utilise Spring</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-48</td>
<td>Allow renaming of projects</td>
<td>Resolved</td>
<td></td>
</tr>
</tbody>
</table>
Bamboo 0.8 Upgrade Guide

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

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 Upgrade Guide

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

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

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</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-873</td>
<td>All Projects table shows the 'little hand' icon over rows that can't be expanded</td>
<td>Resolved</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 exception</td>
<td>Resolved</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>Issue</td>
<td>Description</td>
<td>Status</td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------</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>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.

Unable to render \{include\} Couldn't find a page to include called: __newreleaseBamboo

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</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-162</td>
<td>Previously selected values</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>BAM-570</td>
<td>Testing for mail and IM servers should be more</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>Ticket</td>
<td>Description</td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>-------------</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>

Resolved
Bamboo 1.0-RC1 Release Notes

Unable to render (include) Couldn't find a page to include called: __newreleaseBamboo

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

Unable to render content due to system error: null

"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.

Unable to render content due to system error: null

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

Unable to render content due to system error: null

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

!changes.gif

"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.

Unable to render content due to system error: null

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.

Unable to render content due to system error: null

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.

!FavouriteBuildsPlugin.JPG!

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.

Unable to render content due to system error: null

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.

!bamboo-im-integration.gif!

## 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 Keq</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-795</td>
<td>Project keys should be able to have 2 letters only</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-786</td>
<td>No tab initially selected on dashboard</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-765</td>
<td>Stacktrace if session times out in middle of build configuration wizard</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-759</td>
<td>Display build trigger IP address on 'view plan configuration' page</td>
<td></td>
<td>Resolved</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></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-751</td>
<td>Exception while cancelling build configuration / build artifacts</td>
<td></td>
<td>Resolved</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></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-733</td>
<td>A succesfull building maven 2 multi-project build is failing due to missing test results</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-726</td>
<td>Re-insert missing bamboo-init.properties file</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-666</td>
<td>Improve email notifications</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-626</td>
<td>Ajaxify Dashboard</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-618</td>
<td>AJAXify favourites</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-613</td>
<td>Display project, plan hierarchy on dashboard</td>
<td></td>
<td>Resolved</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>---------</td>
<td>-----------------------------------------------</td>
<td>----------</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>Rogue 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.

Unable to render \{include\} Couldn't find a page to include called: __newreleaseBamboo

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><img src="https://www.atlassian.com" alt="Error" /></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-826</td>
<td>JDK1.4 Incompatible Thread.UncaughtExceptionHandler</td>
<td><img src="https://www.atlassian.com" alt="Error" /></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><img src="https://www.atlassian.com" alt="Error" /></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-799</td>
<td>Security filter should not apply to REST calls</td>
<td><img src="https://www.atlassian.com" alt="Error" /></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-753</td>
<td>Standalone Jetty starts</td>
<td><img src="https://www.atlassian.com" alt="Error" /></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-828</td>
<td>without configuration file</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>---------------------------</td>
<td>----------</td>
<td></td>
</tr>
<tr>
<td>BAM-818</td>
<td>Sitck: client-side browser memory leaks</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-809</td>
<td>Contact Administrators should not be protected with security</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-805</td>
<td>Remove author related tabs from individual user's profile page.</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-804</td>
<td>Firefox left on a Bamboo page leaks memory and hogs CPU</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-801</td>
<td>No program-start menu entry created on installation with the Windows installer</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-797</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-773</td>
<td>The Expand and Collapse link on the View Build Page</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-712</td>
<td>Cannot pass java opts to standalone version</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-459</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>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-724</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-700</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

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.

Unable to render {include} Couldn't find a page to include called: __newreleaseBamboo

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></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-937</td>
<td>Importing data doesn't guarantee unique ids</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-935</td>
<td>SVN Repository doesn't use default authentication</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-958</td>
<td>Links to source is broken</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-939</td>
<td>Need to convert build level plugins to use web fragments</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-908</td>
<td>Standalone Bamboo cannot start in certain Linux environments</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>Issue Number</td>
<td>Description</td>
<td>Resolution</td>
<td></td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
<td>------------</td>
<td></td>
</tr>
<tr>
<td>BAM-889</td>
<td>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 values</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-916</td>
<td>Subversion Event Handler is not all that Null safe</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-869</td>
<td>Error creating new build plan</td>
<td>Resolved</td>
<td></td>
</tr>
</tbody>
</table>
Bamboo 1.0.1 Upgrade Guide

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

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

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>Atlassian JIRA</th>
<th>(18 issues)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Key</td>
<td>Summary</td>
</tr>
<tr>
<td>BAM-982</td>
<td>Bamboo fails to start under JDK 1.4</td>
</tr>
<tr>
<td>BAM-1098</td>
<td>No page associated with this URI</td>
</tr>
<tr>
<td>BAM-1026</td>
<td>Links Are Incorrect When Using 'latest' as the build in the URL</td>
</tr>
<tr>
<td>BAM-1023</td>
<td>Internal error when deleting plans with dependencies</td>
</tr>
<tr>
<td>BAM-1006</td>
<td>Cannot view logs of latest build</td>
</tr>
<tr>
<td>BAM-1002</td>
<td>Perforce commands need better logging</td>
</tr>
<tr>
<td>BAM-994</td>
<td>Internal server error when trying to view TestData history</td>
</tr>
<tr>
<td>BAM-989</td>
<td>Duplicate JARs in classpath</td>
</tr>
<tr>
<td>BAM-988</td>
<td>Error when entered License Key</td>
</tr>
<tr>
<td>BAM-987</td>
<td>FreeMarker template error in plan summary</td>
</tr>
<tr>
<td>BAM-983</td>
<td>FreeMarker template error when viewing 'latest' builds</td>
</tr>
<tr>
<td>BAM-962</td>
<td>Provide ability to point to a CVS tag instead of HEAD/Branch</td>
</tr>
<tr>
<td>BAM-852</td>
<td>Internal error after &quot;Specify Source Repository&quot;</td>
</tr>
<tr>
<td>BAM-806</td>
<td>Support for SSH private key authentication (possible using jsch)</td>
</tr>
<tr>
<td>ID</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>BAM-791</td>
<td>syn+ssh support for private key auth</td>
</tr>
<tr>
<td>BAM-691</td>
<td>CVS Client should use the CVS_RSH environment variable if available</td>
</tr>
<tr>
<td>BAM-1031</td>
<td>Clicking on latest build from home screen does not render the Artifacts or JIRA tabs</td>
</tr>
<tr>
<td>BAM-980</td>
<td>Clicking on Tests tab in Build Result Summary renders empty screen</td>
</tr>
</tbody>
</table>
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

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>BAM-862</td>
<td>Java error on startup</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>Ticket</td>
<td>Description</td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
<td></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></td>
<td></td>
</tr>
<tr>
<td>BAM-710</td>
<td>Internal server error: org.springframework.dao.DataIntegrityViolationException when deleting projects</td>
<td></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

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><strong>Failed to get the build source code: svn: report aborted</strong></td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-974</td>
<td><strong>Bamboo penetrated perforce server with repeated requests on plan creation</strong></td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-750</td>
<td><strong>Perforce changes are not displayed when a manual build is executed</strong></td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1113</td>
<td><strong>Perforce modifications not causing build</strong></td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1096</td>
<td><strong>Change the way bamboo detects changes in perforce</strong></td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1085</td>
<td><strong>Subversion code refresh failing to pick up new revisions</strong></td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1078</td>
<td><strong>BuildChangeDetector continuously polling Perforce repository</strong></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-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 1.0.5 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>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>BAM-1139</td>
<td>Locked externals in SVN causes infinite building loop</td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1137</td>
<td>Authentication always fails for subversion repository</td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1108</td>
<td>Removing last build queue blocks use</td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1100</td>
<td>Cannot log into Bamboo</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>Resolved</td>
</tr>
<tr>
<td>BAM-1050</td>
<td>Null pointer when relogging in after session has died</td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-911</td>
<td>Cannot authenticate with Svn repository</td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1125</td>
<td>Project Creation Fails with Self Signed SSL Certificate for SVN</td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1118</td>
<td>FishEye link from Perforce project causes exception</td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1022</td>
<td>Login link on comment page broken</td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-891</td>
<td>Error when logging in to open source project</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 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

Unable to render content due to system error: null

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 vxzf 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.

\texttt{bamboo.home=\textit{BAMBOO_HOME}}

**Warning:** 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 \texttt{C:dev\bamboo-home} use

\texttt{bamboo.home=C:\dev\bamboo-home}

**Tip:** 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 \texttt{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.

**Warning:** 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 1.0 Development Hub

- Bamboo API
- Bamboo Plugin Developer's Guide
  - Build Complete Action Module
  - Build Processor Module
  - Builder Plugin Module
  - Getting Started
  - Index Reader Module
  - Post Build Index Writer 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 May 04, 2007 by admin.

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 1.0 Documentation Home
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>
```
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

```
<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>
```
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

```xml
<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
<wbml:web-item key="pipelineConfig" name="Build Queues" section="system.admin/builds" weight="20">
  <wbml:link>/admin/configurePipeline!default.action</wbml:link>
  <wbml:condition class="com.atlassian.bamboo.plugins.web.conditions.AdminPermissionCondition"/>
</wbml: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 by 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.
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

```xml
<xwork key="view Clover Result" name="View Clover Result">
  <package name="cloverPlugin" extends="buildView">
    <action name="view Clover Result" class="com.atlassian.bamboo.build.ViewBuildResults">
      <result name="success" type="freemarker">/plugins/clover-plugin/viewCloverResult.ftl</result>
      <result name="error" type="freemarker">/error.ftl</result>
    </action>
  </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

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:

```xml
<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.

```java
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
descriptor (atlassian-plugin.xml).

```xml
<buildCompleteAction key="labeller" name="Build Labeller"
class="com.atlassian.bamboo.plugins.labeller.BuildLabeller">
  <description>An automatic labelling plugin.</description>
</buildCompleteAction>
```

### 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`:

Unable to render content due to system error: null

### 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.

Next >
Tutorial 2 - Configurable Regex Labeller

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 Freemarker templates which allows you to edit and view custom configuration in the Build Plan Configuration page, under the Post Action tab.

Edit Configuration View

The Freemarker template to edit our Labeller configuration is below (regexLabellerEdit.ftl):

```freemarker
[@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 Freemarker view for viewing the configuration (read-only). The display configuration view is below (regexLabellerView.ftl):

```freemarker
[#if build.buildDefinition.customConfiguration.get('custom.bamboo.labeller.regex')?has_content
```

Document generated by Confluence on May 09, 2007 00:14
Page 283
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.

Unable to render content due to system error: null

Unable to render content due to system error: null

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
 */
```
/**
 * 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");
            if (Pattern.matches(pattern, logs.get(0)))
            {
                String label = "custom.bamboo.labeller.label";
                errors.addError(label + " contains invalid characters " + PatternSyntaxException
            }
        }
    }
    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.
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;
    }
}

So that's it! We have now completed a Bamboo plugin containing one BuildCompleteAction module which will match the output logs against a regular expression, and tag it with a set of label(s).