# Space Details

<table>
<thead>
<tr>
<th>Key:</th>
<th>BAMBOO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name:</td>
<td>Bamboo 1.2</td>
</tr>
<tr>
<td>Description:</td>
<td>Documentation for Bamboo 1.2</td>
</tr>
<tr>
<td>Creator (Creation Date):</td>
<td><a href="mailto:edwin@atlassian.com">edwin@atlassian.com</a> (Jul 31, 2006)</td>
</tr>
<tr>
<td>Last Modifier (Mod. Date):</td>
<td><a href="mailto:rosie@atlassian.com">rosie@atlassian.com</a> (Sep 06, 2007)</td>
</tr>
</tbody>
</table>

## Available Pages
- Bamboo Documentation Home
  - Bamboo Administrator's Guide
    - 01. Configuring Projects and Plans
      - 1.1 About Projects, Plans and Builds
      - 1.2 Creating a Plan
      - 1.3 Editing a Plan
      - 1.4 Disabling or deleting a Plan
        - 1.4.1 Deleting a Build Result
        - 1.4.2 Deleting a Plan's Working Files
      - 1.5 Moving a Plan to a different Project
      - 1.6 Renaming a Plan or Project
      - 1.7 Specifying a Plan's Source Repository
        - 1.7.1 CVS
        - 1.7.2 Subversion
        - 1.7.3 Perforce
      - 1.8 Specifying a Plan's Build Artifacts
      - 1.9 Specifying Labels for a Plan's Build Results
    - 02. Configuring Build Resources
      - 2.1 Specifying a Plan's Build Resources
        - 2.1.1 Using Variables
      - 2.2 Configuring a new Builder
      - 2.3 Configuring a new JDK
      - 2.4 Defining Global Variables
    - 03. Triggering a Build
      - 3.1 About Build Triggering
      - 3.2 Triggering a Build when Code is Updated
        - 3.2.1 Polling the Repository for Code Changes
        - 3.2.2 Triggering a Build on Code Check-in
      - 3.3 Triggering a Build on Schedule
        - 3.3.1 Scheduling a Single Daily Build
        - 3.3.2 Specifying a Cron-based Schedule
      - 3.4 Triggering a Build when another Build finishes
      - 3.5 Triggering a Build Manually
    - 04. Managing Build Queues
      - 4.1 Creating a Build Queue
      - 4.2 Assigning a Plan to a Build Queue
4.3 Removing a Plan from a Build Queue
4.4 Stopping an Active Build
4.5 Disabling or deleting a Build Queue

05. Managing Users and Security
5.01 Creating a User
5.02 Changing a User's Password or Details
5.03 Deleting or deactivating a User
5.04 Creating a Group
5.05 Deleting a Group
5.06 Adding Users to and removing them from Groups
5.07 Granting Plan Permissions to Users or Groups
   5.07.01 'Bulk Editing' Plan Permissions
5.08 Granting Global Permissions to Users or Groups
5.09 Granting Administration Rights to a User
5.10 Allowing Anonymous Users to access Bamboo
5.11 Enabling or disabling Contact Details Display
5.12 Enabling or disabling Public Signup
5.13 Working with External User Repositories
   5.13.1 Integrating Bamboo with Crowd
   5.13.2 Integrating Bamboo with LDAP

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

07. Managing Data and Backups
7.1 Locating Important Directories and Files
7.2 Specifying Bamboo's Working Directory
7.3 Viewing your Database Connection Details
7.4 Moving your Bamboo Data to a different Database
7.5 Optimising or Re-indexing Data
7.6 Enabling Expiry of Build Results
   7.6.1 specifying Expiry for a Plan's Build Results
7.7Specifying a Backup Schedule
7.8 Exporting Data for Backup
7.9 Importing Data from Backup

08. Configuring System Settings
8.1 Viewing Bamboo's System Information
8.2 Updating your Bamboo License Details
8.3 Specifying Bamboo's Title
8.4 Specifying Bamboo's URL
8.5 Enabling GZIP Compression
• 8.6 Enabling Bamboo's Remote API
• 8.7 Enabling JIRA integration

• 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 Development Hub
  • Bamboo API
  • Bamboo Plugin Developer's Guide
    • Build Complete Action Module
    • Build Processor Module
    • Builder Plugin Module
    • Getting Started
    • Index Reader Module
    • Notification Condition Module
    • Post Build Index Writer Module
    • Pre Build Action Module
    • Report Module
    • Source Repository 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

• Setting up Bamboo Development Environment in IDEA

• Bamboo Installation & Upgrade Guide
  • Bamboo Installation Guide
    • Bamboo EAR-WAR Installation Guide
    • Bamboo Standalone Installation Guide (Linux)
    • Bamboo Standalone Installation Guide (Mac)
    • Bamboo Standalone Installation Guide (Windows)
  • Running the Setup Wizard
    • 1. Using Bamboo's embedded HSQL database
    • 2. Connecting Bamboo to an external database
      • 2.1 MySQL 4.1 and 5.0
- 2.2 Postgres 8+
- 2.3 Unsupported databases

- Bamboo Release Notes
  - Bamboo 1.0 Release Notes
    - Bamboo 1.0 Upgrade Guide
    - Bamboo 1.0-Beta Release Notes
      - Bamboo 1.0-Beta Upgrade Guide
    - Bamboo 1.0.1 Release Notes
      - Bamboo 1.0.1 Upgrade Guide
    - Bamboo 1.0.2 Release Notes
      - Bamboo 1.0.2 Upgrade Guide
    - Bamboo 1.0.3 Release Notes
      - Bamboo 1.0.3 Upgrade Guide
    - Bamboo 1.0.4 Release Notes
      - Bamboo 1.0.4 Upgrade Guide
    - Bamboo 1.0.5 Release Notes
      - Bamboo 1.0.5 Upgrade Guide
  - Bamboo 1.1 Release Notes
    - Bamboo 1.1 Upgrade Guide
    - Bamboo 1.1.1 Release Notes
      - Bamboo 1.1.1 Upgrade Guide
    - Bamboo 1.1.2 Release Notes
      - Bamboo 1.1.2 Upgrade Guide
  - Bamboo 1.2 Release Notes
    - Bamboo 1.2 Plugin Interface Changes
    - Bamboo 1.2 Upgrade Guide
    - Bamboo 1.2.1 Release Notes
      - Bamboo 1.2.1 Upgrade Guide
    - Bamboo 1.2.2 Release Notes
      - Bamboo 1.2.2 Upgrade Guide
    - Bamboo 1.2.3 Release Notes
      - Bamboo 1.2.3 Upgrade Guide
    - Bamboo 1.2.4 Release Notes
      - Bamboo 1.2.4 Upgrade Guide
  - Bamboo pre-1.0 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-RC1 Release Notes
  • Bamboo 1.0-RC1 Upgrade Guide
• Bamboo 1.0-RC2 Release Notes
  • Bamboo 1.0-RC2 Upgrade Guide
• Bamboo Upgrade Guide
• Bamboo Knowledge Base
• Installation FAQ
  • Adding MIME types to Bamboo Standalone
  • Bamboo 1.2.2 on Tomcat 5
  • Can I have Bamboo.home pointing to a resource on a network share?
  • Changing Bamboo's port from the default 8085
  • Changing the Root Context Path
  • Configuring Bamboo on start-up
  • Getting Bamboo Standalone to use the jetty.xml file
  • Hardware sizing considerations
  • Installation notes for Bamboo on JBoss 4.x
  • Running Bamboo over HTTPS
  • Setting up JNDI on Jetty
  • Supported databases
• Usage FAQ
  • Backing up Bamboo instances over 4GB
  • Binding Bamboo to one IP address
  • Can Bamboo build and test non-Java projects?
  • Can multiple plans share a common 3rd-party directory?
  • Connecting to Subversion repositories
  • Deactivating a Bamboo user
  • Fixing OutOfMemory Errors in Bamboo
  • Problems running Bamboo under Sun JDK 1.4
  • Restoring passwords to recover admin users
  • Testing LDAP or Active Directory connectivity with Paddle
  • Troubleshooting an SVN connection in Bamboo
  • Using Bamboo with Clover
  • Working with Sun JAVA libraries
• Bamboo User's Guide
  • 01. Getting Started
• 1.1 Using the Bamboo Dashboard
• 1.2 Viewing Bamboo’s Current Activity
• 1.3 Viewing your Latest Build Results
• 1.4 Working with Favourites
  • 1.4.1 Adding a Plan to your Favourites
  • 1.4.2 Removing a Plan from your Favourites
• 1.5 Displaying a Build Monitor

• 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 the Code Changes that triggered a Build
  • 3.4 Viewing a Build’s Artifacts
  • 3.5 Viewing a Build Log
  • 3.6 Viewing the Metadata for a Build Result
  • 3.7 Viewing the Clover Code-Coverage for a Build Result
  • 3.8 Viewing the 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 ofSuccessful Builds per Author' Report

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

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

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

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

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

• Glossary
  • activity log
  • artifact
  • author
  • build
  • build activity
  • build duration
  • build log
  • build plan
  • build queue
  • build result
  • build telemetry
  • builder
  • committer
  • favourites
  • global permission
- label
- permission
- plan
- plan permission
- project
- triggering
- watcher
Bamboo Documentation Home

This page last changed on Oct 22, 2007 by rosie@atlassian.com.

Bamboo 1.2.4 has just been released!

- Check out Bamboo's features and try it out while it's hot!
- Read the 1.2.4 Release Notes

<table>
<thead>
<tr>
<th>Installing/upgrading Bamboo 1.2</th>
<th>Resources</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Installation Guide</strong></td>
<td><strong>If you have a question about using Bamboo, please feel free to contact us at support. You may also want to check out the mailing list forums:</strong></td>
</tr>
<tr>
<td><strong>Upgrade Guide</strong></td>
<td><strong>• Bamboo Announcements</strong></td>
</tr>
<tr>
<td><strong>Release Notes</strong></td>
<td><strong>• Bamboo General Forum</strong></td>
</tr>
<tr>
<td><strong>Download Bamboo</strong></td>
<td><strong>• Bamboo Developers Forum</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Using/administering Bamboo 1.2</th>
<th>Resources</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>User's Guide</strong></td>
<td><strong>Other handy links:</strong></td>
</tr>
<tr>
<td><strong>Administrator's Guide</strong></td>
<td>• Knowledge Base</td>
</tr>
<tr>
<td>**Development Hub</td>
<td>Bamboo Extensions**</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>About</th>
<th>Resources</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bamboo is a continuous integration (CI) server. Bamboo assists software development teams by providing:</td>
<td><strong>Previous Versions</strong></td>
</tr>
<tr>
<td>• automated building and testing of software source-code status.</td>
<td><strong>Bamboo 1.1 Documentation Home</strong></td>
</tr>
<tr>
<td>• updates on successful/failed builds.</td>
<td><strong>Bamboo 1.0 Documentation Home</strong></td>
</tr>
<tr>
<td>• reporting tools for statistical analysis.</td>
<td></td>
</tr>
</tbody>
</table>

Error formatting macro: recently-updated: java.lang.NullPointerException
Bamboo Administrator’s Guide

This page last changed on Aug 09, 2007 by rosie@atlassian.com.

---

About

Bamboo is a continuous integration (CI) server. Bamboo assists software development teams by providing:

- automated building and testing of software source-code status.
- updates on successful/failed builds.
- reporting tools for statistical analysis.

The Bamboo Administrator’s Guide provides information about configuring builds and administering your Bamboo system.

For more documentation please visit Bamboo Documentation Home.

---

Download

You can download the Bamboo documentation in PDF, HTML or XML formats.

---

Search the Administrator's Guide

---

Getting Help

Support Request | Feature Request | Forums | Knowledge Base

---

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 Disabling or deleting a Plan
  - 1.4.1 Deleting a Build Result
  - 1.4.2 Deleting a Plan’s Working Files
- 1.5 Moving a Plan to a different Project
- 1.6 Renaming a Plan or Project
- 1.7 Specifying a Plan’s Source Repository
  - 1.7.1 CVS
  - 1.7.2 Subversion
  - 1.7.3 Perforce
- 1.8 Specifying a Plan’s Build Artifacts
- 1.9 Specifying Labels for a Plan’s Build Results

02. Configuring Build Resources

- 2.1 Specifying a Plan’s Build Resources
  - 2.1.1 Using Variables
- 2.2 Configuring a new Builder
- 2.3 Configuring a new JDK
- 2.4 Defining Global Variables

03. Triggering a Build

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

04. Managing Build Queues

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

05. Managing Users and Security

• 5.01 Creating a User
• 5.02 Changing a User's Password or Details
• 5.03 Deleting or deactivating a User
• 5.04 Creating a Group
• 5.05 Deleting a Group
• 5.06 Adding Users to and removing them from Groups
• 5.07 Granting Plan Permissions to Users or Groups
  ° 5.07.01 'Bulk Editing' Plan Permissions
• 5.08 Granting Global Permissions to Users or Groups
• 5.09 Granting Administration Rights to a User
• 5.10 Allowing Anonymous Users to access Bamboo
• 5.11 Enabling or disabling Contact Details Display
• 5.12 Enabling or disabling Public Signup
• 5.13 Working with External User Repositories
  ° 5.13.1 Integrating Bamboo with Crowd
  ° 5.13.2 Integrating Bamboo with LDAP

06. Configuring Email and Instant Messaging Notifications

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

07. Managing Data and Backups

• 7.1 Locating Important Directories and Files
• 7.2 Specifying Bamboo's Working Directory
• 7.3 Viewing your Database Connection Details
• 7.4 Moving your Bamboo Data to a different Database
• 7.5 Optimising or Re-indexing Data
• 7.6 Enabling Expiry of Build Results
  ° 7.6.1 Specifying Expiry for a Plan's Build Results
• 7.7 Specifying a Backup Schedule
• 7.8 Exporting Data for Backup
• 7.9 Importing Data from Backup

08. Configuring System Settings

• 8.1 Viewing Bamboo's System Information
• 8.2 Updating your Bamboo License Details
• 8.3 Specifying Bamboo's Title
• 8.4 Specifying Bamboo's URL
• 8.5 Enabling GZIP Compression
• 8.6 Enabling Bamboo's Remote API
• 8.7 Enabling JIRA integration

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
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 Disabling or deleting a Plan
  - 1.4.1 Deleting a Build Result
  - 1.4.2 Deleting a Plan's Working Files
- 1.5 Moving a Plan to a different Project
- 1.6 Renaming a Plan or Project
- 1.7 Specifying a Plan's Source Repository
  - 1.7.1 CVS
  - 1.7.2 Subversion
  - 1.7.3 Perforce
- 1.8 Specifying a Plan's Build Artifacts
- 1.9 Specifying Labels for a Plan's Build Results
1.1 About Projects, Plans and Builds

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

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

Every plan belongs to a project.

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

Note that creating a new project only requires defining the Project Name and Project Key, which is (optionally) done as part of the process of creating a new plan.

A build is one execution of a plan.

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

Each plan's build results are stored in a subdirectory under your 'Build Directory' (see 7.1 Locating Important Directories and Files).

RELATED TOPICS

- 1.1 About Projects, Plans and Builds
- 1.2 Creating a Plan
- 1.3 Editing a Plan
- 1.4 Disabling or deleting a Plan
  - 1.4.1 Deleting a Build Result
  - 1.4.2 Deleting a Plan's Working Files
- 1.5 Moving a Plan to a different Project
- 1.6 Renaming a Plan or Project
- 1.7 Specifying a Plan's Source Repository
  - 1.7.1 CVS
  - 1.7.2 Subversion
  - 1.7.3 Perforce
- 1.8 Specifying a Plan's Build Artifacts
- 1.9 Specifying Labels for a Plan's Build Results

Bamboo Documentation Home
1.2 Creating a Plan

There are two ways of creating a new plan:

To create a new plan,

1. Click the 'Create Plan' link in the top navigation bar.
2. Enter the required information in the seven screens as described below:
   • 1. 'Plan Details'
   • 2. 'Source Repository'
   • 3. 'Builder Configuration'
   • 4. 'Build Artifacts'
   • 5. 'Notifications'
   • 6. 'Post Actions'
   • 7. 'Permissions'
3. 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, containing all existing plans for which you have the 'Clone' and/or 'Admin' plan permission. Select the plan you wish to copy.
4. Enter the required information in the seven screens as described below. On screens 2-7, appropriate information will be copied from the plan you selected; but you will need to complete all fields on screen 1.

1. 'Plan Details'

Screenshot: 'Plan Details'

- Project — When you create a new plan, you can either add it to an existing project or create a new project. Either:
  ° Select the appropriate project from the drop-down list;
  or:
  ° Select 'New Project' and complete the following two fields:
1. Project Name — Type a descriptive name (e.g. 'Issue Tracking Application') that will identify your project on the Dashboard and in reports.

2. Project Key — Type a logical contraction of the Project Name (e.g. 'ITA'). The Project Key will be included in the plan's Build Results keys (e.g. 'ITA-MAIN-179'), so you may want to make it no longer than 3 or 4 characters. The Project Key must be unique within your Bamboo system.

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

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

2. 'Source Repository'

Screenshot: 'Source Repository'

Here you need to specify what type of source-code repository this plan will use, where the repository is located, and what type of build strategy the plan will use. For details, please see:

- **1.7.1 CVS**
- **1.7.2 Subversion**
- **1.7.3 Perforce**

CVS, Subversion and Perforce are supported out-of-the-box. If you need to use a different type of repository, you can write a Source Repository Module plugin to enable Bamboo to connect to your repository.

3. 'Builder Configuration'

Screenshot: 'Builder Configuration'
Here you need to specify which builder Bamboo will use to build this plan. For details please see 2.1 Specifying a Plan's Build Resources.

One builder (Maven) is automatically configured when you install Bamboo. You can also define other builders (see 2.2 Configuring a new Builder) and then configure this plan to use them (see 2.1 Specifying a Plan's Build Resources).

4. 'Build Artifacts'

Screenshot: 'Build Artifacts'

Here you can define the plan's artifacts, e.g. JAR files which you wish to keep after each build.

Artifacts are copied to a subdirectory (/PLAN_KEY/download_data/) under your 'Build Directory' folder (see 7.1 Locating Important Directories and Files). Artifacts which you define in the plan are listed in each build result as User-defined artifacts (see 3.4 Viewing a Build's Artifacts in the Bamboo User's Guide).

For details, please see 1.8 Specifying a Plan's Build Artifacts.

5. 'Notifications'

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

6. 'Post Actions'

Screenshot: 'Post Actions'

Here you can specify:

- what labels (if any) will be automatically applied to the plan's build results. For details please see 1.9 Specifying Labels for a Plan's Build Results.
- this plan's specific schedule for deletion of build results (if different to the default). For details please see 7.6.1 Specifying Expiry for a Plan's Build Results.

7. 'Permissions'

Screenshot: 'Permissions'

You can edit your plan permissions here. Permissions can be granted to specific users or groups.
Here you need to specify which users will be able to access the plan and its build results. For details please see 5.07 Granting Plan Permissions to Users or Groups.
1.3 Editing a Plan

To edit an existing plan,

1. Click 'Home' to go to the Dashboard.
2. Click the 'All Plans' tab.
3. Locate the plan in the list and click the 'Edit' icon:

4. The 'Configuration' tab will be displayed (see below). Click the eight sub-tabs to edit the following:
   - 'Plan Details' — see 1.6 Renaming a Plan or Project.
   - 'Source Repository' — see 1.7 Specifying a Plan's Source Repository.
   - 'Builder Configuration' — see 2.1 Specifying a Plan's Build Resources.
   - 'Build Artifacts' — see 1.8 Specifying a Plan's Build Artifacts.
   - 'Build Notifications' — see 6.1 Enabling or disabling Notifications for a Plan.
   - 'Post Actions' — see 1.9 Specifying Labels for a Plan's Build Results and 7.6.1 Specifying Expiry for a Plan's Build Results.
   - 'Dependencies' — see 3.4 Triggering a Build when another Build finishes.
   - 'Permissions' — see 5.07 Granting Plan Permissions to Users or Groups.
5. Click the 'Save' button.

Note that 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: 'Configuration'
1.4 Disabling or deleting a Plan

This page last changed on Sep 02, 2007 by rosie@atlassian.com.

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. To do this you will require the 'Admin' global permission. Note that deleting a plan will also delete all of the plan's build results, labels and comments.

To delete a plan, see 7.8 Exporting Data for Backup.

To disable a plan,

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

To delete a plan,

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.

Before you begin
If you need to keep a permanent record of the plan's build results, see 4.4 Stopping an Active Build.

Screenshot: 'Remove Plans'
RELATED TOPICS

- 1.1 About Projects, Plans and Builds
- 1.2 Creating a Plan
- 1.3 Editing a Plan
- 1.4 Disabling or deleting a Plan
  - 1.4.1 Deleting a Build Result
  - 1.4.2 Deleting a Plan's Working Files
- 1.5 Moving a Plan to a different Project
- 1.6 Renaming a Plan or Project
- 1.7 Specifying a Plan's Source Repository
  - 1.7.1 CVS
  - 1.7.2 Subversion
  - 1.7.3 Perforce
- 1.8 Specifying a Plan's Build Artifacts
- 1.9 Specifying Labels for a Plan's Build Results

Bamboo Documentation Home
1.4.1 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 7.6 Enabling Expiry of Build Results.

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

To delete a build result,

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

Only people with the 'Admin' global permission or the 'Admin' plan permission can delete build results.

Screenshot: Build Results for a Plan

RELATED TOPICS

• 1.1 About Projects, Plans and Builds
• 1.2 Creating a Plan
• 1.3 Editing a Plan
• 1.4 Disabling or deleting a Plan
  ° 1.4.1 Deleting a Build Result
  ° 1.4.2 Deleting a Plan's Working Files
• 1.5 Moving a Plan to a different Project
• 1.6 Renaming a Plan or Project
• 1.7 Specifying a Plan's Source Repository
  ° 1.7.1 CVS
  ° 1.7.2 Subversion
  ° 1.7.3 Perforce
• 1.8 Specifying a Plan's Build Artifacts
• 1.9 Specifying Labels for a Plan's Build Results

Bamboo Documentation Home
1.4.2 Deleting a Plan's Working Files

If you want to ensure a clean check-out of your source code when Bamboo runs the next build for a particular plan, you will need to delete the plan's current Working Files.

To delete a plan's working files,

1. Click 'Home' to go to the Dashboard, then click the 'All Plans' tab.
2. Locate the plan in the list and click the plan name.
3. Click the 'Files' tab.
4. A list of the plan's working files will be displayed. Scroll down to the bottom of the screen and click the 'Delete all build files' icon:

Only people with the 'Admin' global permission or the 'Admin' plan permission can delete Working Files.

RELATED TOPICS

- 1.1 About Projects, Plans and Builds
- 1.2 Creating a Plan
- 1.3 Editing a Plan
- 1.4 Disabling or deleting a Plan
  - 1.4.1 Deleting a Build Result
  - 1.4.2 Deleting a Plan's Working Files
- 1.5 Moving a Plan to a different Project
- 1.6 Renaming a Plan or Project
- 1.7 Specifying a Plan's Source Repository
  - 1.7.1 CVS
  - 1.7.2 Subversion
  - 1.7.3 Perforce
- 1.8 Specifying a Plan's Build Artifacts
- 1.9 Specifying Labels for a Plan's Build Results

Bamboo Documentation Home
1.5 Moving a Plan to a different Project

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

Moving a plan to a different project will therefore involve changing the plan's Project Key (as well as possibly the Plan Name and/or Plan Key), which will also change the build key for all of the plan's build results.

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

Before you begin

- Note that moving a plan will require Bamboo to re-index all its data, so your Bamboo system may run slowly for a few minutes.
- It is recommended that you backup your Bamboo build results before you move a plan — see 7.8 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.
6. This will display the 'Configure New Plan Details' screen (see Screenshot 2 below).
7. If the destination project already includes a plan with the same Plan Name, or if you want to change the Plan Name for some other reason, overtype the 'New Plan Name' field.
8. If the destination project already includes a plan with the same Plan Key, or if you want to change the Plan Key for some other reason, overtype the 'New Plan Key' field.
9. Click the 'Move' button.

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

**Move builds**

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

You can move a plan to another project with this wizard. Simply select the plan you want to move and the destination project. A build queue is kept in the current project until the queue is deleted. If you select to move the plan to a new project, you will be asked to delete the build queue and set up the build queue in the new project.

Note that because you are changing plan keys, this operation requires some downtime (e.g. indexing of all builds) and may take a few minutes.

<table>
<thead>
<tr>
<th>Build</th>
<th>Source Build (CLOSED)</th>
<th>Source Build (ABANDONED)</th>
<th>Source Build (CLOSED)</th>
<th>Source Build (ABANDONED)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source Code (CLOSED)</td>
<td>Source Build (CLOSED)</td>
<td>Source Build (ABANDONED)</td>
<td>Source Build (CLOSED)</td>
<td></td>
</tr>
<tr>
<td>Source Code (ABANDONED)</td>
<td>Source Build (ABANDONED)</td>
<td>Source Build (CLOSED)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Destination Project:**

- **Project:**
  - **Plan:**
    - **Build Name:**
      - **Build Queue:**
        - **Plan Name:**
          - **Queue Name:**

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

### Configure New Plan Details

The following plans will be moved to project **Plan Name**. You can update the plan names and keys below.

The existing plan for the destination project includes:

- **Main Build (ABANDONED)**

<table>
<thead>
<tr>
<th>Original Project</th>
<th>Original Name</th>
<th>New Name</th>
<th>Original Key</th>
<th>New Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>Atlassian Build</td>
<td>Atlassian Build</td>
<td>Atlassian Build</td>
<td>Atlassian Build</td>
<td>Atlassian Build</td>
</tr>
</tbody>
</table>

**Move** **Cancel**

### RELATED TOPICS

- **1.1 About Projects, Plans and Builds**
- **1.2 Creating a Plan**
- **1.3 Editing a Plan**
- **1.4 Disabling or deleting a Plan**
  - **1.4.1 Deleting a Build Result**
  - **1.4.2 Deleting a Plan's Working Files**
- **1.5 Moving a Plan to a different Project**
- **1.6 Renaming a Plan or Project**
- **1.7 Specifying a Plan's Source Repository**
  - **1.7.1 CVS**
  - **1.7.2 Subversion**
  - **1.7.3 Perforce**
- **1.8 Specifying a Plan's Build Artifacts**
- **1.9 Specifying Labels for a Plan's Build Results**

[Link to Bamboo Documentation Home]
1.6 Renaming a Plan or Project

To rename a plan or a project,

1. Click 'Home' to go to the Dashboard.
2. Click the 'All Plans' tab.
3. Locate the plan in the list and click the 'Edit' icon:

4. The 'Configuration' tab will displayed. Click the 'Plan Details' sub-tab.
5. The 'Plan Details' sub-tab will be displayed (see below).
6. In the 'Project Name' field, type the project's new name.
7. In the 'Build Plan Name' field, type the plan's new name
8. Click the 'Save' button.

Note that 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: 'Edit Plan Details'

RELATED TOPICS

• 1.1 About Projects, Plans and Builds
• 1.2 Creating a Plan
• 1.3 Editing a Plan
• 1.4 Disabling or deleting a Plan
  ° 1.4.1 Deleting a Build Result
  ° 1.4.2 Deleting a Plan's Working Files
• 1.5 Moving a Plan to a different Project
• 1.6 Renaming a Plan or Project
• 1.7 Specifying a Plan's Source Repository
  ° 1.7.1 CVS
  ° 1.7.2 Subversion
  ° 1.7.3 Perforce
• 1.8 Specifying a Plan's Build Artifacts
• 1.9 Specifying Labels for a Plan's Build Results

Bamboo Documentation Home
1.7 Specifying a Plan's Source Repository

For each plan, you need to specify what type of source-code repository the plan will use, where the repository is located, and what type of build strategy the plan will use. Please see:

- 1.7.1 CVS
- 1.7.2 Subversion
- 1.7.3 Perforce

CVS, Subversion and Perforce are supported out-of-the-box. If you need to use a different type of repository, you can write a Source Repository Module plugin to enable Bamboo to connect to your repository.

RELATED TOPICS

- 1.1 About Projects, Plans and Builds
- 1.2 Creating a Plan
- 1.3 Editing a Plan
- 1.4 Disabling or deleting a Plan
  - 1.4.1 Deleting a Build Result
  - 1.4.2 Deleting a Plan's Working Files
- 1.5 Moving a Plan to a different Project
- 1.6 Renaming a Plan or Project
- 1.7 Specifying a Plan's Source Repository
  - 1.7.1 CVS
  - 1.7.2 Subversion
  - 1.7.3 Perforce
- 1.8 Specifying a Plan's Build Artifacts
- 1.9 Specifying Labels for a Plan's Build Results

Bamboo Documentation Home
1.7.1 CVS

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

For each plan, you need to specify what type of source-code repository the plan will use, where the repository is located, and what type of build strategy the plan will use.

To specify a CVS repository,

If you are creating a new plan, start at step 5.

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:

   ![Icon](image)

4. The 'Configuration' tab will be displayed. Click the 'Source Repository' sub-tab.
5. The 'Source Repository' sub-tab will be displayed (see screenshot below). Complete the fields as follows:

   - **Repository** — 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. Note that you can use global variables in this field (see 2.1.1 Using Variables).
   - **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. Currently Bamboo has limited support for CVS ampersand modules. To use an ampersand module, you will need to define a regular module with the same name as the ampersand module (since Bamboo expects there to be a directory with the specified checkout module name). For example:
     1. Create a module (e.g. `allbuilds`).
     2. Define an ampersand module with the same name. (The ampersand module can be empty.)
     3. In the 'Module' field, enter the following: `allbuilds allbuilds &project2 &project2 &project3`
   - **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. Note that you can use global variables in this field (see 2.1.1 Using Variables).
   - **Web Repository URL** — (Optional) You can specify the URL of the plan's browsable repository. If you specify a Web Repository URL, then links to relevant files will be displayed in the 'Code Changes' section of a build result.
   - **Include/Exclude Files** — (Optional) You can specify a particular inclusion or exclusion pattern for file changes to be detected.
   - **File Pattern** — (Optional) The regular expression for file changes which you wish to include/exclude.
   - **Web Repository Module** — (Optional) The plan's repository name, if the above Web Repository URL points to multiple repositories.
   - **Build Strategy** — The default value, 'Poll the repository for changes', is a convenient option that requires no additional configuration. A number of other options are available; for details, please see 03. Triggering a Build. You can change the Build Strategy over time as required. The rest of the fields on this tab will vary depending on which Build Strategy you select.
• Click the 'Save' button if you are editing an existing plan; or if you are creating a new plan, click the 'Next' button and go to 2.1 Specifying a Plan's Build Resources.

**Screenshot: ‘Source Repository’**

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

**Source Repository**

**Repository:**

**CVS Root:**

The full path to your CVS repository root. Bamboo supports local and remote repositories.

**Authentication Type:**

- **Password**

**Password:**

(Optional) The password required to access the CVS repository.

**Quiet Period:**

- **2**

CVS changes are not atomic. How many seconds should Bamboo wait between checks to determine if the check is complete?

**Module:**

The repository module containing the source code.

**Version of module:**

- **HEAD**

Which version of the module should Bamboo build?

**Include / Exclude Files:**

- **Exclude all changes that matches the following pattern**

**File Pattern:**

- **documentation/**

**Web Repository URL:**

- **http://myserver.atlassian.com/views/repo/atlassianCVS/**

**Web Repository Module:**

- **myrepo**

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

**Build Strategy:**

- **Polling the Repository for changes**

How often Bamboo checks that the source repository has changed?

**Polling Frequency:**

- **160**

How often (in seconds) should Bamboo check the repository for changes?

**RELATED TOPICS**

- **1.1 About Projects, Plans and Builds**
- **1.2 Creating a Plan**
- **1.3 Editing a Plan**
- **1.4 Disabling or deleting a Plan**
  - **1.4.1 Deleting a Build Result**
  - **1.4.2 Deleting a Plan’s Working Files**
- **1.5 Moving a Plan to a different Project**
- **1.6 Renaming a Plan or Project**
- **1.7 Specifying a Plan’s Source Repository**
  - **1.7.1 CVS**
  - **1.7.2 Subversion**
  - **1.7.3 Perforce**
- **1.8 Specifying a Plan’s Build Artifacts**
- **1.9 Specifying Labels for a Plan’s Build Results**

**Bamboo Documentation Home**
1.7.2 Subversion

For each plan, you need to specify what type of source-code repository the plan will use, where the repository is located, and what type of build strategy the plan will use.

To specify a Subversion repository,

1. Click 'Home' to go to the Dashboard.
2. Click the 'All Plans' tab.
3. Locate the plan in the list and click this icon:

4. The 'Configuration' tab will be displayed. Click the 'Source Repository' sub-tab.
5. The 'Source Repository' sub-tab will be displayed. Complete the fields as follows:

- 'Repository' — select 'Subversion'.
- 'Repository URL' — The location of your Subversion repository (e.g. http://svn.collab.net/repos/svn/trunk). Note that you can use global variables in this field (see 2.1.1 Using Variables).
- '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 fields 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 (note that your externals must be in the root of the checkout directory, not in a subdirectory). Please note that you only need to select this check box if you require Bamboo to detect changes in the externals. If your externals references a particular (static) revision, you do not need to check this box.
- 'Web Repository URL' --- (Optional) You can specify the URL of the plan's browsable repository. If you specify a Web Repository URL, then links to relevant files will be displayed in the 'Code Changes' section of a build result.
- 'Include/Exclude Files' — (Optional) You can specify a particular inclusion or exclusion pattern for file changes to be detected.
- 'File Pattern' — (Optional) The regular expression for file changes which you wish to include/exclude.
- 'Web Repository Module' — (Optional) The plan's repository name, if the above Web Repository URL points to multiple repositories.
- 'Build Strategy' — The default value, 'Poll the repository for changes', is a convenient option that requires no additional configuration. A number of other options are available; for details, please see 03. Triggering a Build. You can change the Build Strategy over time as required. The rest of the fields on this tab will vary depending on which Build Strategy you select.

If you are creating a new plan, start at step 5.

If you are creating a new plan, start at step 5.

Related Topics:

- 1.1 About Projects, Plans and Builds
- 1.2 Creating a Plan
- 1.3 Editing a Plan
- 1.4 Disabling or deleting a Plan
1.4.1 Deleting a Build Result
1.4.2 Deleting a Plan’s Working Files
1.5 Moving a Plan to a different Project
1.6 Renaming a Plan or Project
1.7 Specifying a Plan’s Source Repository
   1.7.1 CVS
   1.7.2 Subversion
   1.7.3 Perforce
1.8 Specifying a Plan’s Build Artifacts
1.9 Specifying Labels for a Plan’s Build Results

Bamboo Documentation Home
1.7.3 Perforce

For each plan, you need to specify what type of source-code repository the plan will use, where the repository is located, and what type of build strategy the plan will use.

To specify a Perforce repository,

1. Click 'Home' to go to the Dashboard.
2. Click the 'All Plans' tab.
3. Locate the plan in the list and click this icon:

4. The 'Configuration' tab will be displayed. Click the 'Source Repository' sub-tab.
5. Complete the fields as follows:

   - 'Repository' — select '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 client view of the depot that contains the source-code files for this plan. This is typically in the form //<clientname>/<workspace_mapping>/... 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).
   - 'Password' — (Optional) Type the password required by the Perforce 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 Perforce repository.
   - 'Web Repository URL' — (Optional) You can specify the URL of the plan's browsable repository. If you specify a Web Repository URL, then links to relevant files will be displayed in the 'Code Changes' section of a build result.
   - 'Include/Exclude Files' — (Optional) You can specify a particular inclusion or exclusion pattern for file changes to be detected.
   - 'File Pattern' — (Optional) The regular expression for file changes which you wish to include/exclude.
   - 'Web Repository Module' — (Optional) The plan's repository name, if the above Web Repository URL points to multiple repositories.
   - 'Build Strategy' — The default value, 'Poll the repository for changes', is a convenient option that requires no additional configuration. A number of other options are available; for details, please see G3. 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.

6. Click the 'Save' button if you are editing an existing plan; or if you are creating a new plan, click the 'Next' button and go to 2.1 Specifying a Plan's Build Resources.

RELATED TOPICS
• 1.1 About Projects, Plans and Builds
• 1.2 Creating a Plan
• 1.3 Editing a Plan
• 1.4 Disabling or deleting a Plan
  ° 1.4.1 Deleting a Build Result
  ° 1.4.2 Deleting a Plan’s Working Files
• 1.5 Moving a Plan to a different Project
• 1.6 Renaming a Plan or Project
• 1.7 Specifying a Plan’s Source Repository
  ° 1.7.1 CVS
  ° 1.7.2 Subversion
  ° 1.7.3 Perforce
• 1.8 Specifying a Plan’s Build Artifacts
• 1.9 Specifying Labels for a Plan’s Build Results

Bamboo Documentation Home
1.8 Specifying a Plan's Build Artifacts

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

A plan's artifacts are any reports, websites or files (e.g. JAR files) which you wish to keep after each build. Artifacts are copied to a subdirectory (/PLAN_KEY/download_data/) under your 'Build Directory' folder (see 7.1 Locating Important Directories and Files). Artifacts which you define in the plan are listed in each build result as User-defined artifacts (see 3.4 Viewing a Build's Artifacts in the Bamboo User’s Guide).

To specify a plan's build artifacts,

1. If you are creating a new plan, start at step 5.
2. Click 'Home' to go to the Dashboard.
3. Click the 'All Plans' tab.
4. Locate the plan in the list and click this icon:
5. The 'Configuration' tab will be displayed. Click the 'Build Artifacts' sub-tab.
6. In the 'Artifact Label' field, specify the name which Bamboo will use to describe the artifact.
7. In the 'Artifact Copy Pattern' field, specify the name (or Regular Expression pattern) of the artifact(s) you want to keep.
8. In the 'Source Directory' field, specify the directory (including path) where Bamboo will look for your artifact.
9. Click the 'Save' button if you are editing an existing plan; or if you are creating a new plan, click the 'Next' button and go to 6.1 Enabling or disabling Notifications for a Plan.

For example, if you want to keep the latest version of a JAR you have built, you could specify Artifact Copy Pattern to be "*/.jar" and the Source Directory to be 'target'.

Screenshot: 'Build Artifacts'

RELATED TOPICS

- 1.1 About Projects, Plans and Builds
- 1.2 Creating a Plan
- 1.3 Editing a Plan
- 1.4 Disabling or deleting a Plan
  - 1.4.1 Deleting a Build Result
  - 1.4.2 Deleting a Plan's Working Files
- 1.5 Moving a Plan to a different Project
- 1.6 Renaming a Plan or Project
- 1.7 Specifying a Plan's Source Repository
  - 1.7.1 CVS
  - 1.7.2 Subversion
  - 1.7.3 Perforce
- 1.8 Specifying a Plan's Build Artifacts
- 1.9 Specifying Labels for a Plan's Build Results

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

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

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. The 'Post Actions' tab will be displayed (see screenshot below).
6. In the 'Regex Pattern' field, you can either:
   • Specify a regular expression on which to match the log files. Labels will be applied to the build if the regular expression finds a match (see the examples below).
   • Leave this field blank to label every build.
7. In the 'Labels' field, type the word (or multiple words, separated by commas and/or spaces) with which the plan's build results are to be labelled.
8. Click the 'Save' button if you are editing an existing plan; or if you are creating a new plan, either go to 7.6.1 Specifying Expiry for a Plan's Build Results or click the 'Next' button and go to 5.07 Granting Plan Permissions to Users or Groups.

Regex example:

'There are \d+ results'

In the above regex, '\d+' represents any number with one or more digits. ('\d' means 'any digit', and '+' means 'one or more times'. When combined, they mean 'any sequence of one or more digits'.)

Therefore, positive matches would include:

• 'There are 0 results'
• 'There are 123 results'

Regex example with multiple labels:
You can use "capturing groups" with Bamboo 1.2.1 or later to create different labels for different purposes.

For example,

- Enter the following into the 'Regex Pattern' field:

(\PERFORMANCE\_IMPROVED|\PERFORMANCE\_DETERIORATED)

- Enter the following into the 'Labels' field:

\1

These settings will label your builds with PERFORMANCE\_IMPROVED if "PERFORMANCE\_IMPROVED" appears in the build log, and PERFORMANCE\_DETERIORATED if "PERFORMANCE\_DETERIORATED" appears in the build log. If both strings appear in a log, then both labels are applied to the build.

RELATED TOPICS

- 1.1 About Projects, Plans and Builds
- 1.2 Creating a Plan
- 1.3 Editing a Plan
- 1.4 Disabling or deleting a Plan
  - 1.4.1 Deleting a Build Result
  - 1.4.2 Deleting a Plan's Working Files
- 1.5 Moving a Plan to a different Project
- 1.6 Renaming a Plan or Project
- 1.7 Specifying a Plan's Source Repository
  - 1.7.1 CVS
  - 1.7.2 Subversion
  - 1.7.3 Perforce
- 1.8 Specifying a Plan's Build Artifacts
- 1.9 Specifying Labels for a Plan's Build Results

Bamboo Documentation Home
02. Configuring Build Resources

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

2. Configuring Build Resources

- 2.1 Specifying a Plan's Build Resources
  - 2.1.1 Using Variables
- 2.2 Configuring a new Builder
- 2.3 Configuring a new JDK
- 2.4 Defining Global Variables
2.1 Specifying a Plan's Build Resources

This page last changed on Nov 02, 2007 by rosie@atlassian.com.

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 add more builders of different types as described in 2.2 Configuring a new Builder. Once added, they will appear in the 'Builder' field as described below.

To specify a plan's builder and build resources,

If you are creating a new plan, start at step 5.

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:

![App Icon](image)

4. The 'Configuration' tab will be displayed. Click the 'Builder Configuration' sub-tab.
5. The 'Builder Configuration' sub-tab will be displayed (see screenshot below).
6. In the 'Builder' field, select the builder 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). You can include variables (see 2.1.1 Using Variables).
     - '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 JVM parameters, e.g.: `-Djava.awt.headless=true` (this will pass the parameter 'java.awt.headless' with a value of 'true'). You can also include variables (see 2.1.1 Using Variables).

   - **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 JVM parameters, e.g.: `-Djava.awt.headless=true` (this will pass the parameter 'java.awt.headless' with a value of 'true'). You can also include variables (see 2.1.1 Using Variables).

   - **Custom command**:
     - 'Argument' — Specify the relevant argument to pass to the command. Note that arguments which contain spaces must be quoted. You can include variables (see 2.1.1 Using Variables).

   - **Script**:
     - 'Script' — Specify the location of the script file. This can be either relative to the repository root of the plan, or absolute. You can include variables (see 2.1.1 Using Variables).
     - 'Argument' — Specify the relevant argument to pass to the script. Note that arguments which contain spaces must be quoted. You can include variables (see 2.1.1 Using Variables).

7. 'Build JDK' — If you selected an Ant or Maven builder above, you will need to choose a JDK from the list.

8. 'System Environment Variables' — (Optional) Specify any additional* operating system environment variables you want to pass to your build (e.g. ANT_HOME=/tools/ant); use spaces to separate multiple variables. You can also include Bamboo global or build-specific variables (see 2.1.1 Using Variables).

   * i.e. additional to the existing environment variables (see 8.1 Viewing Bamboo's System Information for a list). Note that existing environment variables are automatically available to the builder, i.e. you don't need to specify them in the 'System Environment Variables' field.

9. '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, as described in 7.1 Locating Important Directories and Files). 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.

10. 'The build has tests' — Select this check-box if you want Bamboo to gather test results data for each
build result. (Note that Bamboo requires test results to be XML files that are compatible with JUnit
XML format) 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. The following field will appear:
      '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. Please specify file path relative to your plan’s root directory
(e.g /home/bamboouser/bamboo-home/xml-data/build-dir/MY_PLAN/), i.e. please do not
specify an absolute path.

11. 'Clover output will be produced' — Select this check-box if you are running Atlassian Clover
and want to view its code-coverage data from within Bamboo (see 3.7 Viewing the Clover Code-
Coverage for a Build Result). The following field will be displayed:
   • 'Clover XML Directory' — Specify the name of the directory (including path) where Bamboo
will look for the XML report output file from Clover. Please specify file path relative to your
plan’s root directory (e.g /home/bamboouser/bamboo-home/xml-data/build-dir/MY_PLAN/),
i.e. please do not specify an absolute path.

12. Click the 'Save' button if you are editing an existing plan; or if you are creating a new plan, click the
'Next' button and go to 1.8 Specifying a Plan’s Build Artifacts.

Screenshot: 'Builder Configuration'

RELATED TOPICS

• 2.1 Specifying a Plan’s Build Resources
  • 2.1.1 Using Variables
• 2.2 Configuring a new Builder
• 2.3 Configuring a new JDK
• 2.4 Defining Global Variables
2.1.1 Using Variables

When configuring a plan, you may want to specify variables to be used in the build process. There are two types of variables available to you:

- Global variables are defined across your entire Bamboo instance, and have the same (static) value for every plan that is built by Bamboo. You can define as many global variables as you wish (see 2.4 Defining Global Variables).
- Build-specific variables are evaluated by Bamboo dynamically at build time. The source of a build-specific variable can either be a Bamboo property or one of the default plugins (assuming they have been enabled). The following build-specific variables are available by default:

<table>
<thead>
<tr>
<th>Build-specific variable</th>
<th>Source</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>buildKey</td>
<td>Bamboo property</td>
<td>The plan key for the build, e.g. BAM-MAIN</td>
</tr>
<tr>
<td>buildNumber</td>
<td>Bamboo property</td>
<td>The Bamboo build number, e.g. 123</td>
</tr>
<tr>
<td>custom.svn.revision.number</td>
<td>Plugin</td>
<td>(For Subversion only) The revision number</td>
</tr>
<tr>
<td>custom.cvs.last.update.time</td>
<td>Plugin</td>
<td>(For CVS only) The last updated timestamp</td>
</tr>
<tr>
<td>custom.p4.revision.number</td>
<td>Plugin</td>
<td>(For Perforce only) The change set number</td>
</tr>
</tbody>
</table>

The usage format for all global and build-specific variables is:

`${bamboo.<variable>}`

Variables can be used in the following fields of your build plan:

<table>
<thead>
<tr>
<th>Field (for Maven builders only) — see 2.1 Specifying a Plan's Build Resources</th>
<th>Available variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>Goal</td>
<td>Global variables</td>
</tr>
<tr>
<td></td>
<td>Build-specific variables</td>
</tr>
<tr>
<td>Build File (for Ant and NAnt builders only) — see 2.1 Specifying a Plan's Build Resources</td>
<td>Global variables</td>
</tr>
<tr>
<td></td>
<td>Build-specific variables</td>
</tr>
<tr>
<td>Target (for Ant and NAnt builders only) — see 2.1 Specifying a Plan's Build Resources</td>
<td>Global variables</td>
</tr>
<tr>
<td></td>
<td>Build-specific variables</td>
</tr>
<tr>
<td>Options (for NAnt builders only) — see 2.1 Specifying a Plan's Build Resources</td>
<td>Global variables</td>
</tr>
<tr>
<td></td>
<td>Build-specific variables</td>
</tr>
<tr>
<td>Script (for Scripts only) — see 2.1 Specifying a Plan's Build Resources</td>
<td>Global variables</td>
</tr>
<tr>
<td></td>
<td>Build-specific variables</td>
</tr>
<tr>
<td>Argument (for Scripts and Custom Commands only) — see 2.1 Specifying a Plan's Build Resources</td>
<td>Global variables</td>
</tr>
<tr>
<td></td>
<td>Build-specific variables</td>
</tr>
<tr>
<td>System Environment Variables — see 2.1 Specifying a Plan's Build Resources</td>
<td>Global variables</td>
</tr>
<tr>
<td></td>
<td>Build-specific variables</td>
</tr>
<tr>
<td>Repository URL (for Subversion repositories only)</td>
<td>Global variables</td>
</tr>
</tbody>
</table>
For example, you may want your Maven 2 version to be determined by Bamboo. In Maven 2 `pom.xml` you may have:

```xml
...<groupId>com.atlassian.boo</groupId>
<artifactId>boo-test</artifactId>
<packaging>jar</packaging>
<version>1.1.$(env.bambooBuildNumber)-SNAPSHOT</version>
...
```

You can then specify the following in the 'Goal' field of your build plan:

```
clean package -DbambooBuildNumber=${bamboo.buildNumber}
```

When the command runs, Bamboo will replace the `buildNumber` with the actual number (e.g. 1102), which will be passed to the underlying Maven build to use. The command will then produce a jar that looks like this: `boo-test-1.1.1102-SNAPSHOT.jar`. 
2.2 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 2.1 Specifying a Plan's Build Resources).

To configure a new builder,

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

Screenshot: Configuring a Builder

<table>
<thead>
<tr>
<th>Label</th>
<th>Type</th>
<th>Path</th>
<th>Is Valid?</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rich</td>
<td>Custom Command</td>
<td>Antivirus</td>
<td>OK</td>
<td>Edit/Cancel</td>
</tr>
<tr>
<td>Make</td>
<td>Custom Command</td>
<td>Antivirus</td>
<td>OK</td>
<td>Edit/Cancel</td>
</tr>
<tr>
<td>Maven 2.0</td>
<td>Maven 2.0</td>
<td>notarisedmars</td>
<td>OK</td>
<td>Edit/Cancel</td>
</tr>
<tr>
<td>Ant</td>
<td>Ant</td>
<td>notarisedmars</td>
<td>OK</td>
<td>Edit/Cancel</td>
</tr>
<tr>
<td>MAVEN_HOME</td>
<td>Maven</td>
<td>notarisedmars</td>
<td>OK</td>
<td>Save</td>
</tr>
</tbody>
</table>

RELATED TOPICS

- 2.1 Specifying a Plan's Build Resources
  - 2.1.1 Using Variables
- 2.2 Configuring a new Builder
- 2.3 Configuring a new JDK
- 2.4 Defining Global Variables

Bamboo Documentation Home
2.3 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 in your Bamboo system, it will appear in the 'Build JDK' drop-down list on the 'Build Resources' tab when you configure a build plan (see 2.1 Specifying a Plan's Build Resources).

To configure a new JDK,

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

Screenshot: JDKs

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1.5 Java</td>
<td>NotJavaHome</td>
<td>OK</td>
<td>Edit/Delete</td>
</tr>
<tr>
<td>1.4 Java</td>
<td>NotJavaHome</td>
<td>OK</td>
<td>Edit/Delete</td>
</tr>
</tbody>
</table>

Available JDKs

You can use this page to view, add and delete system JDKs. The table below shows the JDKs currently configured on this server. The label will appear in the configuration build plan.

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

RELATED TOPICS

- 2.1 Specifying a Plan's Build Resources
  - 2.1.1 Using Variables
- 2.2 Configuring a new Builder
- 2.3 Configuring a new JDK
- 2.4 Defining Global Variables

Bamboo Documentation Home
2.4 Defining Global Variables

When configuring a plan, you may want to specify variables to be used in the build process. For details please see 2.1.1 Using Variables.

Global variables are one type of variable that is available to you. Global variables are defined across your entire Bamboo instance, and have the same value for every plan that is built by Bamboo.

To add a new global variable,

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

Screenshot: Global Variables

RELATED TOPICS

- 2.1 Specifying a Plan's Build Resources
  - 2.1.1 Using Variables
- 2.2 Configuring a new Builder
- 2.3 Configuring a new JDK
- 2.4 Defining Global Variables

Bamboo 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 your source-code repository is configured to fire an event to the Bamboo server.
    See 3.2.2 Triggering a Build on Code Check-in.

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

Also see 4.4 Stopping an Active Build.

RELATED TOPICS

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

Bamboo Documentation Home
3.2 Triggering a Build when Code is Updated

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

- "Pull strategy" — Polling the repository for code changes means that Bamboo will check-out the source-code on a regular basis, and examine it for changes. If Bamboo detects a change, it will trigger a build.
  
  See 3.2.1 Polling the Repository for Code Changes.

- "Push strategy" — Triggering a build on code check-in has the advantage of placing minimal load on your Bamboo server, but requires that your source-code 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 Documentation Home
3.2.1 Polling the Repository for Code Changes

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

To poll the repository for changes,

1. Click 'Home' to go to the Dashboard.
2. Click the 'All Plans' tab.
3. Locate the plan in the list and click this icon:

4. The 'Configuration' tab will be displayed. Click the 'Source Repository' sub-tab.
5. In the 'Build Strategy' field, select 'Poll the repository for changes' (see screenshot below).
6. In the 'Polling Frequency' field, specify how often (in seconds) Bamboo should check-out the source-code and examine it for changes.
7. Click the 'Save' button.

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

RELATED TOPICS

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

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

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

Step 1. Configure your source repository

1. Configure your source-code repository to run post-commit scripts to tell Bamboo whenever a code commit has occurred.
   - For CVS, edit two files in the CVSROOT module: commitinfo and loginfo.
     * For commitinfo add a line like this:
       
       `^jira(/|$) /pathto/preCommit.sh`
       
       where "jira" is your module.
     * For loginfo add a line like this:
       
       `^jira(/|$) /pathto/postCommitBuildTrigger.sh %{} http://bambooserver JIRA-MAIN JIRA-BRANCH`
       
       where JIRA-MAIN and JIRA-BRANCH are the Bamboo plans that you would like to trigger.
   - For Subversion, edit the Subversion repository's hooks/post-commit trigger file with something like:
     
     `/pathto/postCommitBuildTrigger.sh http://bambooserver JIRA-MAIN JIRA-BRANCH`
   - For Perforce, add the script as a change-commit trigger.
     
     `triggerName change-commit //myDepot/... "'/usr/local/bin/postCommitBuildTrigger.sh http://bambooserver/ MYPLAN-DEFAULT"`
2. Copy the scripts to your repository. If you are using Bamboo Standalone, the scripts are located in the /scripts folder of your Bamboo Installation Directory. If you are using Bamboo EAR-WAR you can find them in the /repositoryScripts folder. You can also download the scripts by following this link.
3. Depending on which operating system your repository is running on, you may need to edit the scripts. The scripts assume that 'wget' is in '/usr/bin/'; if this isn't the case for your repository (e.g. Solaris 10 has it in /usr/sfw/bin/), edit the scripts and change '/usr/bin/' to the appropriate location.
4. Ensure that the user which Bamboo is running as has appropriate file permissions to execute the scripts, i.e. the scripts should be executable by non-root user(s).
5. Enable Bamboo's remote API so that the scripts can use Bamboo's REST-style remote API to access Bamboo's data.

Step 2. Configure Bamboo to trigger a build on code check-in

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

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

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

Bamboo Documentation Home
3.3 Triggering a Build on Schedule

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

- **Single Daily Build** —
  A _single daily build_ runs at a time of your choice. This is particularly suitable for builds that take a long time to complete. \ See 3.3.1 Scheduling a Single Daily Build.  

- **Cron-Based Scheduling** —
  A _cron-based schedule_ allows you to schedule builds according to a flexible [cron](http://www.opensymphony.com/quartz/wikidocs/TutorialLesson6.html] expression. For example, "0 0/30 9-19 ? * MON-FRI" would trigger a build every half-an-hour from 9am to 7pm, Monday to Friday. See 3.3.2 Specifying a Cron-based Schedule.

**RELATED TOPICS**

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

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

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

To schedule a single daily build,

1. Click 'Home' to go to the Dashboard.
2. Click the 'All Plans' tab.
3. Locate the plan in the list and click this icon:

4. The 'Configuration' tab will be displayed. Click the 'Source Repository' sub-tab.
5. In the 'Build Strategy' field, select 'Single Daily Build' (see screenshot below).
6. In the 'Build Time' field, specify the time of day at which the build should run. Specify the time in "hh:mm" format, using a 24-hour clock.
7. Click the 'Save' button.

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

RELATED TOPICS

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

Bamboo Documentation Home
3.3.2 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.

To specify a cron-based schedule,

1. Click 'Home' to go to the Dashboard.
2. Click the 'All Plans' tab.
3. Locate the plan in the list and click this icon:

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

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

RELATED TOPICS

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

Bamboo Documentation Home
3.4 Triggering a Build when another Build finishes

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

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

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

See 3.5 Triggering a Build Manually.

To trigger a build when another build finishes,

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

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

RELATED TOPICS

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

Bamboo Documentation Home
3.5 Triggering a Build Manually

To start a manual build,

1. Click 'Home' to go to the Dashboard.
2. Locate the relevant plan and click the 'Check Out and Build' icon:

   ✅ See also 4.4 Stopping an Active Build.

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

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

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

To specify 'manual builds only',

1. Click 'Home' to go to the Dashboard.
2. Click the 'All Plans' tab.
3. Locate the plan in the list and click this icon:

4. The 'Configuration' tab will be displayed. Click the 'Source Repository' sub-tab.
5. In the 'Build Strategy' field, select 'Manual builds only' (see screenshot below).
6. Click the 'Save' button.

Screenshot: 'Plan Configuration--Source Repository--Build Strategy: Manual builds only'
RELATED TOPICS

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

Bamboo Documentation Home
04. Managing Build Queues

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

4. Configuring Build Queues

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

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

One default queue is created when you install Bamboo.

To create a new queue,

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

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

Screenshot: 'Build Queue--Add'

Build Queues

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

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

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

RELATED TOPICS

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

Bamboo Documentation Home
4.2 Assigning a Plan to a Build Queue

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

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

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

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

To assign a plan to a build queue,

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

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

To assign all plans to a build queue,

1. Remove any plans currently assigned to the build queue, as described in 4.3 Removing a Plan from a Build Queue.

When there are no plans explicitly assigned to a build queue, the queue will accept builds from all plans.

Screenshot: 'Build Queue--Edit'

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

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Allowed builds</th>
<th>Queued Builds</th>
<th>Status</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fast Builds</td>
<td>Bamboo - HEAD</td>
<td></td>
<td>0</td>
<td>Disabled</td>
</tr>
<tr>
<td></td>
<td>Crowd - Crowd JDK 1.6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>JIRA-MAIN</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Crowd - Main Build</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>JIRA-MAIN+BIMON</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Crowd - Crowd JDK 1.4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Queue 2</td>
<td>Atlassian Config - Clover Build</td>
<td></td>
<td>0</td>
<td>Disabled</td>
</tr>
<tr>
<td></td>
<td>Atlassian Config - Main Build</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Atlassian Core - Main Build</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Atlassian External - Main Build</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Atlassian External - Main Build</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Atlassian Log Analysis - Default</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Atlassian Maven2 - Main Build</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Atlassian Plugins - Main Build</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Document generated by Confluence on Dec 06, 2007 19:00
RELATED TOPICS

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

Bamboo Documentation Home
4.3 Removing a Plan from a Build Queue

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

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

To remove a plan from a build queue,

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

Screenshot: 'Build Queue--Edit'

Build Queues

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

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Allowed builds</th>
<th>Owned Builds</th>
<th>Status</th>
<th>Operations</th>
</tr>
</thead>
</table>
| Fast Builds | Bamboo - HEAD  
Crond - Crond JSDK 1.6  
JRE WNT  
Crond - Main Build  
JREWNTCH Main  
Crond - Crond JSDK 1.4 | 0            | Disabled | Edit | Delete | Enable |
| Queue 2 | Atlassian Coding - Clover Build  
Atlassian Coding - Main Build  
Atlassian Core - Main Build  
Atlassian Event - Main Build  
Atlassian Extras - Main Build  
Atlassian Log Analysis - Default  
Atlassian Maven2 Base Project - Main Build  
Atlassian Overview - Main Build | 0            | Disabled | Stop | Cancel |
4.4 Stopping an Active Build

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

To stop an active build,

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

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

To stop a plan from submitting builds to a particular queue, see 4.3 Removing a Plan from a Build Queue. Also see 1.4 Disabling or deleting a Plan.

RELATED TOPICS

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

Bamboo Documentation Home
4.5 Disabling or deleting a Build Queue

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

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

Sometimes you might need to stop Bamboo from building any plans. You could achieve this by disabling all queues. Note that you can also delete/disable individual plans — see 1.4 Disabling or deleting a Plan.

To disable a build queue,

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

To delete a build queue,

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

Screenshot: 'Build Queue--Delete or Disable'

**Build Queues**

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

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Allowed Builds</th>
<th>Queued Builds</th>
<th>Status</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fast Build</td>
<td>JIRA: Main Build</td>
<td>0</td>
<td>Enabled</td>
<td>Edit [Disable Build]</td>
</tr>
<tr>
<td>Queue 2</td>
<td>Atlassian Robot: Main Build</td>
<td>1</td>
<td>Enabled</td>
<td>Edit [Disable Build]</td>
</tr>
<tr>
<td></td>
<td>Atlassian Robot: Main Build</td>
<td>0</td>
<td>Enabled</td>
<td>Delete</td>
</tr>
</tbody>
</table>
5. Managing Users and Security

About users and authors

An author is any person who contributes to a build by checking-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 configure Bamboo to grant access to non-users. However, only Bamboo users can:

- view the 'My Bamboo' tab on the Dashboard.
- belong to a group.

About groups

Bamboo groups are used to specify which users will have global permissions and plan permissions. They can also be 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.

A special group called bamboo-admin is automatically created when you install Bamboo. Members of this group have Bamboo administration rights.

About permissions

A plan permission is the ability to perform a particular operation in relation to a build plan. For each plan, different permissions can be granted to particular groups and/or users.

A global permission is the ability to perform a particular operation in relation to Bamboo as a whole.

More about...

- 5.01 Creating a User
- 5.02 Changing a User's Password or Details
- 5.03 Deleting or deactivating a User
- 5.04 Creating a Group
- 5.05 Deleting a Group
- 5.06 Adding Users to and removing them from Groups
- 5.07 Granting Plan Permissions to Users or Groups
  - 5.07.01 'Bulk Editing' Plan Permissions
- 5.08 Granting Global Permissions to Users or Groups
- 5.09 Granting Administration Rights to a User
- 5.10 Allowing Anonymous Users to access Bamboo
- 5.11 Enabling or disabling Contact Details Display
- 5.12 Enabling or disabling Public Signup
- 5.13 Working with External User Repositories
  - 5.13.1 Integrating Bamboo with Crowd
  - 5.13.2 Integrating Bamboo with LDAP
5.01 Creating a User

This page last changed on Sep 20, 2007 by rosie@atlassian.com.

A user is someone who can login to Bamboo. An author is any person who contributes to a build by checking-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 configure Bamboo to grant access to non-users. However, only Bamboo users can:

- view the 'My Bamboo' tab on the Dashboard.
- belong to a group.

To create a Bamboo user,

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

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

• 5.01 Creating a User
• 5.02 Changing a User's Password or Details
• 5.03 Deleting or deactivating a User
• 5.04 Creating a Group
• 5.05 Deleting a Group
• 5.06 Adding Users to and removing them from Groups
• 5.07 Granting Plan Permissions to Users or Groups
  • 5.07.01 'Bulk Editing' Plan Permissions
• 5.08 Granting Global Permissions to Users or Groups
• 5.09 Granting Administration Rights to a User
• 5.10 Allowing Anonymous Users to access Bamboo
• 5.11 Enabling or disabling Contact Details Display
• 5.12 Enabling or disabling Public Signup
• 5.13 Working with External User Repositories
  • 5.13.1 Integrating Bamboo with Crowd
  • 5.13.2 Integrating Bamboo with LDAP

Bamboo Documentation Home
5.02 Changing a User's Password or Details

To change a user's password,

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. To locate the user, type part of their 'Username', 'Full Name' or 'Email' and click the 'Enter' button. This will display a list of matching users.
4. Click the 'Edit' link (in the 'Operations' column) that corresponds to the user.
5. The 'User Details' screen will be displayed. Type the new password in the 'Password' and 'Confirm Password' fields.

   If you have configured SMTP email on your Bamboo server, the user will automatically receive an email containing their new password. The user can easily change their password later.

6. Click the 'Save' button.

To change a user's details (e.g. Email or Full Name),

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. To locate the user, type part of their 'Username', 'Full Name' or 'Email' and click the 'Enter' button. This will display a list of matching users.
4. Click the 'Edit' link (in the 'Operations' column) that corresponds to the user.
5. The 'User Details' screen will be displayed. Enter the new details as described in 5.01 Creating a User.

   If you do not need to change the user's password, simply leave the 'Password' and 'Confirm Password' fields blank.

6. Click the 'Save' button.

Note

Users who have forgotten their passwords can click the 'Forgotten your password?' link on the Bamboo login screen. This will automatically generate a new password and email it to the user (provided the Bamboo server has been configured to send SMTP email).

Logged-in users can also change their own password and details, as described in 11. Editing your User Profile in the Bamboo User's Guide:

- 11.1 Changing your Password
- 11.2 Changing your Notification Preferences
- 11.3 Associating your Author Name with your User Profile

RELATED TOPICS

- 5.01 Creating a User
- 5.02 Changing a User's Password or Details
- 5.03 Deleting or deactivating a User
- 5.04 Creating a Group
- 5.05 Deleting a Group
- 5.06 Adding Users to and removing them from Groups
- 5.07 Granting Plan Permissions to Users or Groups
  - 5.07.01 'Bulk Editing' Plan Permissions
- 5.08 Granting Global Permissions to Users or Groups
- 5.09 Granting Administration Rights to a User
- 5.10 Allowing Anonymous Users to access Bamboo
- 5.11 Enabling or disabling Contact Details Display
- 5.12 Enabling or disabling Public Signup
• 5.13 Working with External User Repositories
  ° 5.13.1 Integrating Bamboo with Crowd
  ° 5.13.2 Integrating Bamboo with LDAP

Bamboo Documentation Home
5.03 Deleting or deactivating a User

Deleting a user removes their Bamboo user account. Deactivating a user revokes their permission to login to Bamboo.

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 may want to deactivate them instead.
- You cannot delete the user account with which you are currently logged in to Bamboo.

To delete a Bamboo user,

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

To deactivate a Bamboo user,

- please see Deactivating a Bamboo user

RELATED TOPICS

- 5.01 Creating a User
- 5.02 Changing a User's Password or Details
- 5.03 Deleting or deactivating a User
- 5.04 Creating a Group
- 5.05 Deleting a Group
- 5.06 Adding Users to and removing them from Groups
- 5.07 Granting Plan Permissions to Users or Groups
  - 5.07.01 'Bulk Editing' Plan Permissions
- 5.08 Granting Global Permissions to Users or Groups
- 5.09 Granting Administration Rights to a User
- 5.10 Allowing Anonymous Users to access Bamboo
- 5.11 Enabling or disabling Contact Details Display
- 5.12 Enabling or disabling Public Signup
- 5.13 Working with External User Repositories
  - 5.13.1 Integrating Bamboo with Crowd
  - 5.13.2 Integrating Bamboo with LDAP

Bamboo Documentation Home
5.04 Creating a Group

This page last changed on Sep 02, 2007 by rosie@atlassian.com.

Bamboo groups are used to specify which users will have global permissions and plan permissions. They can also be 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.

A special group called bamboo-admin is automatically created when you install Bamboo. Members of this group have Bamboo administration rights.

To create a group,

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

Screenshot: Create Group

Create Group

Group Details

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

- Group Name:
- Users in Group:

Screenshot:

RELATED TOPICS

- 5.01 Creating a User
- 5.02 Changing a User's Password or Details
- 5.03 Deleting or deactivating a User
- 5.04 Creating a Group
- 5.05 Deleting a Group
- 5.06 Adding Users to and removing them from Groups
- 5.07 Granting Plan Permissions to Users or Groups
  - 5.07.01 'Bulk Editing' Plan Permissions
- 5.08 Granting Global Permissions to Users or Groups
- 5.09 Granting Administration Rights to a User
- 5.10 Allowing Anonymous Users to access Bamboo
- 5.11 Enabling or disabling Contact Details Display
- 5.12 Enabling or disabling Public Signup
• 5.13 Working with External User Repositories
  ° 5.13.1 Integrating Bamboo with Crowd
  ° 5.13.2 Integrating Bamboo with LDAP

Bamboo Documentation Home
5.05 Deleting a Group

Bamboo groups are used to specify which users will have global permissions and plan permissions. They can also be 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.

A special group called bamboo-admin is automatically created when you install Bamboo. Members of this group have Bamboo administration rights.

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

<table>
<thead>
<tr>
<th>Group</th>
<th>Number of Users</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>bamboo-admin</td>
<td>20</td>
<td>Edit</td>
</tr>
<tr>
<td>casual-developer</td>
<td>5</td>
<td>Edit, Delete</td>
</tr>
</tbody>
</table>

RELATED TOPICS

- 5.01 Creating a User
- 5.02 Changing a User's Password or Details
- 5.03 Deleting or deactivating a User
- 5.04 Creating a Group
- 5.05 Deleting a Group
- 5.06 Adding Users to and removing them from Groups
- 5.07 Granting Plan Permissions to Users or Groups
  - 5.07.01 'Bulk Editing' Plan Permissions
- 5.08 Granting Global Permissions to Users or Groups
- 5.09 Granting Administration Rights to a User
- 5.10 Allowing Anonymous Users to access Bamboo
- 5.11 Enabling or disabling Contact Details Display
- 5.12 Enabling or disabling Public Signup
- 5.13 Working with External User Repositories
  - 5.13.1 Integrating Bamboo with Crowd
  - 5.13.2 Integrating Bamboo with LDAP

Bamboo Documentation Home
Bamboo groups are used to specify which users will have global permissions and plan permissions. They can also be 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.

A special group called bamboo-admin is automatically created when you install Bamboo. Members of this group have Bamboo administration rights.

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.

To remove users from a group,

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

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

- 5.01 Creating a User
- 5.02 Changing a User's Password or Details
- 5.03 Deleting or deactivating a User
- 5.04 Creating a Group
- 5.05 Deleting a Group
- 5.06 Adding Users to and removing them from Groups
- 5.07 Granting Plan Permissions to Users or Groups
  - 5.07.01 'Bulk Editing' Plan Permissions
- 5.08 Granting Global Permissions to Users or Groups
- 5.09 Granting Administration Rights to a User
- 5.10 Allowing Anonymous Users to access Bamboo
- 5.11 Enabling or disabling Contact Details Display
- 5.12 Enabling or disabling Public Signup
- 5.13 Working with External User Repositories
  - 5.13.1 Integrating Bamboo with Crowd
  - 5.13.2 Integrating Bamboo with LDAP

Bamboo Documentation Home
5.07 Granting Plan Permissions to Users or Groups

A plan permission is the ability to perform a particular operation in relation to a build plan. For each plan, different permissions can be granted to particular groups and/or users.

The following plan permissions are available:

<table>
<thead>
<tr>
<th>Plan permission</th>
<th>Description</th>
<th>Can be granted to</th>
</tr>
</thead>
<tbody>
<tr>
<td>'View'</td>
<td>Permission to:</td>
<td>- a particular user</td>
</tr>
<tr>
<td></td>
<td>- view this plan's build results</td>
<td>- a particular group</td>
</tr>
<tr>
<td></td>
<td>- add comments or labels to this plan's build results¹</td>
<td>- all logged-in users</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- anonymous users²</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>'Edit'</td>
<td>Permission to view and edit this plan's configuration, except for the plan's permissions.</td>
<td>- a particular user</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- a particular group</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- all logged-in users</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- anonymous users²</td>
</tr>
<tr>
<td>'Build'</td>
<td>Permission to:</td>
<td>- a particular user</td>
</tr>
<tr>
<td></td>
<td>- manually start/stop a build for this plan.</td>
<td>- a particular group</td>
</tr>
<tr>
<td></td>
<td>- disable/enable this plan from submitting builds to a queue.</td>
<td>- all logged-in users</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- anonymous users²</td>
</tr>
<tr>
<td>'Clone'</td>
<td>Permission to copy this plan when creating a new plan. (Note: only users with the 'Create Plan' global permission can create new plans.)</td>
<td>- a particular user</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- a particular group</td>
</tr>
<tr>
<td>'Admin'</td>
<td>Permission to:</td>
<td>- a particular user</td>
</tr>
<tr>
<td></td>
<td>- edit this plan's permissions.</td>
<td>- a particular group</td>
</tr>
<tr>
<td></td>
<td>- delete this plan's build results and working files.</td>
<td></td>
</tr>
</tbody>
</table>

¹ Only logged-in users (not anonymous users) can label or comment on a build result.
² Anonymous users cannot access Bamboo at all unless they have been granted the 'Access' global permission. See 5.10 Allowing Anonymous Users to access Bamboo.

Anyone with the 'Admin' global permission automatically has all plan permissions for every plan.

The processes for granting and revoking plan permissions are described below. Note that, for ongoing ease of management, it is recommended that you grant permissions to groups rather than to individual users.
To grant plan permissions to a user,

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 'Permissions' tab (see screenshot below).
5. In the 'Grant permission to' list at the bottom of the screen, select 'User'.
6. Type the username into the box, or click the following icon to select from a list of users: 🔄
   Note that the assignment of permissions to LDAP users and groups in Bamboo is case sensitive. For instance, if the username of the LDAP user is 'Bob', you will need to type in 'Bob' (not 'bob' or 'BOB').
7. Click the 'Add' button.
8. The user will be added to the list of users on the 'Permissions' tab. Select the check-box for each permission that you wish to grant to this user.
9. Click the 'Save' button.

To grant plan permissions to a group,

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 'Permissions' tab (see screenshot below).
5. In the 'Grant permission to' list at the bottom of the screen, select 'Group'.
6. Type the group name into the box.
   Note that the assignment of permissions to LDAP users and groups in Bamboo is case sensitive. For instance, if the name of the LDAP group is 'Dev', you will need to type in 'Dev' (not 'dev' or 'DEV').
7. Click the 'Add' button.
8. The group will be added to the list of groups on the 'Permissions' tab. Select the check-box for each permission that you wish to grant to this group.
9. Click the 'Save' button.

To grant plan permissions to all Bamboo users,

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 'Permissions' tab (see screenshot below).
5. Locate 'Logged in users' (under 'Other').
6. Select the check-box for each permission that you wish to grant to all Bamboo users.
7. Click the 'Save' button.

To grant plan permissions to anonymous users,

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 'Permissions' tab (see screenshot below).
5. Locate 'Anonymous users' (under 'Other').
6. Select the check-box for each permission that you wish to grant to all anonymous users.
7. Click the 'Save' button.

To revoke plan permissions,

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 'Permissions' tab (see screenshot below).
5. Locate the relevant user/group/all logged-in users/anonymous users.
6. Deselect the check-box for each permission that you wish to revoke from the user/group/all users/anonymous users.

   If you deselect all permissions for a user or group, that user or group will disappear from the 'Permissions' tab for this plan.
7. Click the 'Save' button.

### Screenshot: Plan Permissions

![Permissions screenshot](image)

#### RELATED TOPICS

- 5.01 Creating a User
- 5.02 Changing a User's Password or Details
- 5.03 Deleting or deactivating a User
- 5.04 Creating a Group
- 5.05 Deleting a Group
- 5.06 Adding Users to and removing them from Groups
- 5.07 Granting Plan Permissions to Users or Groups
  - 5.07.01 'Bulk Editing' Plan Permissions
- 5.08 Granting Global Permissions to Users or Groups
- 5.09 Granting Administration Rights to a User
- 5.10 Allowing Anonymous Users to access Bamboo
- 5.11 Enabling or disabling Contact Details Display
- 5.12 Enabling or disabling Public Signup
- 5.13 Working with External User Repositories
  - 5.13.1 Integrating Bamboo with Crowd
  - 5.13.2 Integrating Bamboo with LDAP

**Bamboo Documentation Home**
5.07.01 'Bulk Editing' Plan Permissions

A plan permission is the ability to perform a particular operation in relation to a build plan. For each plan, different permissions can be granted to particular groups and/or users.

For a description of the various plan permissions ('View', 'Edit', 'Clone', 'Build' and 'Admin'), see 5.07 Granting Plan Permissions to Users or Groups.

How plan permissions are granted:

- People who have the 'Admin' global permission can 'bulk edit' permissions for multiple plans at the same time, as described below. Note that this will overwrite any pre-existing plan permissions.
- People who have the 'Admin' plan permission for one or more plans, but do not have the 'Admin' global permission, can only edit one plan at a time, as described in 5.07 Granting Plan Permissions to Users or Groups.

The processes for granting and revoking permissions across multiple plans are as follows. Note that, for ongoing ease of management, it is recommended that you grant permissions to groups rather than to individual users.

To grant multiple plan permissions to a user,

1. Click the 'Administration' link in the top navigation bar.
2. Click the 'Bulk Edit Plan Permissions' link in the left navigation column.
3. The first screen of the 'Bulk Edit Plan Permissions Wizard' will be displayed (see below). Select the plans whose permissions you wish to edit, then click the 'Next' button (at the bottom of the screen).
4. The second screen of the 'Bulk Edit Plan Permissions Wizard' will be displayed (see below).
5. In the 'Grant permission to' list at the bottom of the screen, select 'User'.
6. Type the username into the box, or click the following icon to select from a list of users:

![User Icon]

7. Click the 'Add' button.
8. The user will be added to the list of users on the screen. Select the check-box for each permission that you wish to grant to this user.
9. Click the 'Save' button.

To grant multiple plan permissions to a group,

1. Click the 'Administration' link in the top navigation bar.
2. Click the 'Bulk Edit Plan Permissions' link in the left navigation column.
3. The first screen of the 'Bulk Edit Plan Permissions Wizard' will be displayed (see below). Select the plans whose permissions you wish to edit, then click the 'Next' button (at the bottom of the screen).
4. The second screen of the 'Bulk Edit Plan Permissions Wizard' will be displayed (see below).
5. In the 'Grant permission to' list at the bottom of the screen, select 'Group'.
6. Type the group name into the box.
7. Click the 'Add' button.
8. The group will be added to the list of groups on the screen. Select the check-box for each permission that you wish to grant to this group.
9. Click the 'Save' button.

To grant multiple plan permissions to all Bamboo users,

1. Click the 'Administration' link in the top navigation bar.
2. Click the 'Bulk Edit Plan Permissions' link in the left navigation column.
3. The first screen of the 'Bulk Edit Plan Permissions Wizard' will be displayed (see below). Select the plans whose permissions you wish to edit, then click the 'Next' button (at the bottom of the screen).
4. The second screen of the 'Bulk Edit Plan Permissions Wizard' will be displayed (see below).
5. Locate 'Logged in users' (under 'Other').
6. Select the check-box for each permission that you wish to grant to all Bamboo users.
7. Click the 'Save' button.

To grant multiple plan permissions to anonymous users,

1. Click the 'Administration' link in the top navigation bar.
2. Click the 'Bulk Edit Plan Permissions' link in the left navigation column.
3. The first screen of the 'Bulk Edit Plan Permissions Wizard' will be displayed (see below). Select the plans whose permissions you wish to edit, then click the 'Next' button (at the bottom of the screen).
4. The second screen of the 'Bulk Edit Plan Permissions Wizard' will be displayed (see below).
5. Locate 'Anonymous users' (under ‘Other’).
   - Anonymous users are people who are not logged in to Bamboo.
6. Select the check-box for each permission that you wish to grant to all anonymous users.
7. Click the 'Save' button.

To revoke multiple plan permissions,

1. Click the 'Administration' link in the top navigation bar.
2. Click the 'Bulk Edit Plan Permissions' link in the left navigation column.
3. The first screen of the 'Bulk Edit Plan Permissions Wizard' will be displayed (see below). Select the plans whose permissions you wish to edit, then click the 'Next' button (at the bottom of the screen).
4. The second screen of the 'Bulk Edit Plan Permissions Wizard' will be displayed (see below).
5. Locate the relevant user/group/all logged-in users/anonymous users.
6. Deselect the check-box for each permission that you wish to revoke from the user/group/all users/anonymous users.
   - If you deselect all permissions for a user or group, that user or group will disappear from the screen.
7. Click the 'Save' button.

Screenshot 1: ‘Bulk Edit Plan Permissions Wizard — Select Plans’
Screenshot 2: 'Bulk Edit Plan Permissions Wizard — Permissions'

RELATED TOPICS

- 5.01 Creating a User
- 5.02 Changing a User's Password or Details
- 5.03 Deleting or deactivating a User
- 5.04 Creating a Group
- 5.05 Deleting a Group
- 5.06 Adding Users to and removing them from Groups
- 5.07 Granting Plan Permissions to Users or Groups
  - 5.07.01 'Bulk Editing' Plan Permissions
- 5.08 Granting Global Permissions to Users or Groups
- 5.09 Granting Administration Rights to a User
- 5.10 Allowing Anonymous Users to access Bamboo
- 5.11 Enabling or disabling Contact Details Display
- 5.12 Enabling or disabling Public Signup
- 5.13 Working with External User Repositories
  - 5.13.1 Integrating Bamboo with Crowd
  - 5.13.2 Integrating Bamboo with LDAP

Bamboo Documentation Home
5.08 Granting Global Permissions to Users or Groups

A global permission is the ability to perform a particular operation in relation to Bamboo as a whole.

You can grant the following global permissions:

<table>
<thead>
<tr>
<th>Global permission</th>
<th>Description</th>
<th>Can be granted to</th>
</tr>
</thead>
</table>
| 'Access'          | Permission to view the Bamboo system. (Note that the ability to view build plans and build results is subject to individual plan permissions.) | - a particular user
                   |             | - a particular group
                   |             | - all logged-in users
                   |             | - anonymous users¹ |
| 'Create Plan'     | Permission to create new build plans. | - a particular user
                   |             | - a particular group
                   |             | - all logged-in users |
| 'Admin'           | Permission to:  
                   | • access the Bamboo 'Administration' menu.  
                   | • delete plans.  
                   | The 'Admin' global permission also includes all plan permissions, for every plan. | - a particular user
                   |             | - a particular group |

¹ i.e. people who are not logged in to Bamboo.

The processes for granting and revoking global permissions are described below.

**To grant global permissions to a user,**

1. Click 'Home' to go to the Dashboard.
2. Click the 'Administration' link in the top navigation bar.
3. Click the 'Global Permissions' link in the left navigation column to display the 'Global Permissions' screen (see screenshot below).
4. In the 'Grant permission to' list at the bottom of the screen, select 'User'.
5. Type the username into the box, or click the following icon to select from a list of users:

![Icon to select users](image)

Note that the assignment of permissions to LDAP users and groups in Bamboo is case sensitive. For instance, if the username of the LDAP user is 'Bob', you will need to type in 'Bob' (not 'bob' or 'BOB').
6. Click the 'Add' button.
7. The user will be added to the list of users on the 'Permissions' tab. Select the check-box for each permission that you wish to grant to this user.
8. Click the 'Save' button.

**To grant global permissions to a group,**

1. Click 'Home' to go to the Dashboard.
2. Click the 'Administration' link in the top navigation bar.
3. Click the 'Global Permissions' link in the left navigation column to display the 'Global Permissions' screen (see screenshot below).
4. In the 'Grant permission to' list at the bottom of the screen, select 'Group'.
5. Type the group name into the box.
   Note that the assignment of permissions to LDAP users and groups in Bamboo is case sensitive.
   For instance, if the name of the LDAP group is 'Dev', you will need to type in 'Dev' (not 'dev' or
   'DEV').
6. Click the 'Add' button.
7. The group will be added to the list of groups on the 'Permissions' tab. Select the check-box for each
   permission that you wish to grant to this group.
8. Click the 'Save' button.

To grant global permissions to all Bamboo users,

1. Click 'Home' to go to the Dashboard.
2. Click the 'Administration' link in the top navigation bar.
3. Click the 'Global Permissions' link in the left navigation column to display the 'Global Permissions'
   screen (see screenshot below).
4. Locate 'All logged in users' (under 'Other').
5. Select the check-box for each permission that you wish to grant to all Bamboo users.
6. Click the 'Save' button.

To grant global permissions to anonymous users,

1. Click 'Home' to go to the Dashboard.
2. Click the 'Administration' link in the top navigation bar.
3. Click the 'Global Permissions' link in the left navigation column to display the 'Global Permissions'
   screen (see screenshot below).
4. Locate 'Anonymous users' (under 'Other').
5. Select the 'Access' check-box.
6. Click the 'Save' button.

To revoke global permissions,

1. Click 'Home' to go to the Dashboard.
2. Click the 'Administration' link in the top navigation bar.
3. Click the 'Global Permissions' link in the left navigation column to display the 'Global Permissions'
   screen (see screenshot below).
4. Locate the relevant user/group/all logged-in users/anonymous users.
5. Deselect the check-box for each permission that you wish to revoke from the user/group/all users/
   anonymous users.
6. Click the 'Save' button.

Screenshot: Global Permissions
GLOBAL PERMISSIONS

You can edit your global application level permissions here. Permissions can be granted to specific users or groups. Please note these are global application permissions. For plan level permissions, please go to the plan configuration page.

Bamboo Application Permissions

<table>
<thead>
<tr>
<th>Groups</th>
<th>Access</th>
<th>Create</th>
<th>Admin</th>
</tr>
</thead>
<tbody>
<tr>
<td>bamboo-admin</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Other</th>
<th>Access</th>
<th>Create</th>
<th>Admin</th>
</tr>
</thead>
<tbody>
<tr>
<td>All logged in users</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Anonymous users</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Grant permission to: <select> <option value='User'>User</option> <option value='Group'>Group</option> <option value='Plan'>Plan</option> </select>

ADD

SAVE CANCEL

RELATED TOPICS

- 5.01 Creating a User
- 5.02 Changing a User's Password or Details
- 5.03 Deleting or deactivating a User
- 5.04 Creating a Group
- 5.05 Deleting a Group
- 5.06 Adding Users to and removing them from Groups
- 5.07 Granting Plan Permissions to Users or Groups
  - 5.07.01 'Bulk Editing' Plan Permissions
- 5.08 Granting Global Permissions to Users or Groups
- 5.09 Granting Administration Rights to a User
- 5.10 Allowing Anonymous Users to access Bamboo
- 5.11 Enabling or disabling Contact Details Display
- 5.12 Enabling or disabling Public Signup
- 5.13 Working with External User Repositories
  - 5.13.1 Integrating Bamboo with Crowd
  - 5.13.2 Integrating Bamboo with LDAP

Bamboo Documentation Home
5.09 Granting Administration Rights to a User

In Bamboo, there are two types of administrators:

- Global administrators — that is, people with the 'Admin' global permission. These people can access the Bamboo 'Administration' menu. They can also administer every plan.
- Plan administrators — that is, people with the 'Admin' and 'Edit' plan permissions. These people can administer a particular plan.

To grant global administration rights to a user,

- Either grant the 'Admin' global permission to the user explicitly (as described in 5.08 Granting Global Permissions to Users or Groups);
- OR:
  - Add the user to a group which has the 'Admin' global permission (as described in 5.06 Adding Users to and removing them from Groups).

To grant plan administration rights to a user,

- Either grant the 'Admin' and 'Edit' plan permissions to the user explicitly (as described in 5.07 Granting Plan Permissions to Users or Groups);
- OR:
  - Add the user to a group which has the 'Admin' and 'Edit' plan permissions (as described in 5.06 Adding Users to and removing them from Groups).

RELATED TOPICS

- 5.01 Creating a User
- 5.02 Changing a User's Password or Details
- 5.03 Deleting or deactivating a User
- 5.04 Creating a Group
- 5.05 Deleting a Group
- 5.06 Adding Users to and removing them from Groups
- 5.07 Granting Plan Permissions to Users or Groups
- 5.07.01 'Bulk Editing' Plan Permissions
- 5.08 Granting Global Permissions to Users or Groups
- 5.09 Granting Administration Rights to a User
- 5.10 Allowing Anonymous Users to access Bamboo
- 5.11 Enabling or disabling Contact Details Display
- 5.12 Enabling or disabling Public Signup
- 5.13 Working with External User Repositories
  - 5.13.1 Integrating Bamboo with Crowd
  - 5.13.2 Integrating Bamboo with LDAP

Bamboo Documentation Home
5.10 Allowing Anonymous Users to access Bamboo

Allowing anonymous users to access your Bamboo system means that people who aren't logged in to Bamboo will be able to perform most of the functions described in the Bamboo User's Guide (e.g. generating reports; viewing plans and build results) — subject to individual plan permissions.

Note that people who aren't logged in to Bamboo do not have a 'My Bamboo' tab on their Dashboard.

To allow anonymous users to access Bamboo,

1. Click 'Home' to go to the Dashboard.
2. Click the 'Administration' link in the top navigation bar.
3. Click the 'Global Permissions' link in the left navigation column to display the 'Global Permissions' screen (see screenshot below).
4. Locate 'Anonymous users' (under 'Other').
5. Select the 'Access' check-box.
6. Click the 'Save' button.

Anonymous users will now be able to access your Bamboo system. However, they will only be able to view plans and build results for plans where the 'Access' plan permission has been granted to 'Anonymous users'.

Screenshot: Global Permissions

RELATED TOPICS

- 5.01 Creating a User
- 5.02 Changing a User's Password or Details
- 5.03 Deleting or deactivating a User
- 5.04 Creating a Group
- 5.05 Deleting a Group
- 5.06 Adding Users to and removing them from Groups
- 5.07 Granting Plan Permissions to Users or Groups
  - 5.07.01 'Bulk Editing' Plan Permissions
- 5.08 Granting Global Permissions to Users or Groups
- 5.09 Granting Administration Rights to a User
- 5.10 Allowing Anonymous Users to access Bamboo
- 5.11 Enabling or disabling Contact Details Display
- 5.12 Enabling or disabling Public Signup
- 5.13 Working with External User Repositories
  - 5.13.1 Integrating Bamboo with Crowd
5.13.2 Integrating Bamboo with LDAP

Bamboo Documentation Home
5.11 Enabling or disabling Contact Details Display

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

To enable contact details display,

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

To disable contact details display,

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

RELATED TOPICS

• 5.01 Creating a User
• 5.02 Changing a User's Password or Details
• 5.03 Deleting or deactivating a User
• 5.04 Creating a Group
• 5.05 Deleting a Group
• 5.06 Adding Users to and removing them from Groups
• 5.07 Granting Plan Permissions to Users or Groups
  ° 5.07.01 'Bulk Editing' Plan Permissions
• 5.08 Granting Global Permissions to Users or Groups
• 5.09 Granting Administration Rights to a User
• 5.10 Allowing Anonymous Users to access Bamboo
• 5.11 Enabling or disabling Contact Details Display
• 5.12 Enabling or disabling Public Signup
• 5.13 Working with External User Repositories
  ° 5.13.1 Integrating Bamboo with Crowd
  ° 5.13.2 Integrating Bamboo with LDAP

[Bamboo 1.1 Documentation Home]
5.12 Enabling or disabling Public Signup

If you enable signup for your Bamboo system, visitors can create their own Bamboo user accounts.

To enable signup,

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

To disable signup,

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

Screenshot: 'Signup'

RELATED TOPICS

- 5.01 Creating a User
- 5.02 Changing a User's Password or Details
- 5.03 Deleting or deactivating a User
- 5.04 Creating a Group
- 5.05 Deleting a Group
- 5.06 Adding Users to and removing them from Groups
- 5.07 Granting Plan Permissions to Users or Groups
  - 5.07.01 'Bulk Editing' Plan Permissions
- 5.08 Granting Global Permissions to Users or Groups
- 5.09 Granting Administration Rights to a User
- 5.10 Allowing Anonymous Users to access Bamboo
- 5.11 Enabling or disabling Contact Details Display
- 5.12 Enabling or disabling Public Signup
- 5.13 Working with External User Repositories
  - 5.13.1 Integrating Bamboo with Crowd
  - 5.13.2 Integrating Bamboo with LDAP
5.13 Working with External User Repositories

You can integrate external user repositories with Bamboo:

- 5.13.1 Integrating Bamboo with Crowd
- 5.13.2 Integrating Bamboo with LDAP

RELATED TOPICS

- 5.01 Creating a User
- 5.02 Changing a User’s Password or Details
- 5.03 Deleting or deactivating a User
- 5.04 Creating a Group
- 5.05 Deleting a Group
- 5.06 Adding Users to and removing them from Groups
- 5.07 Granting Plan Permissions to Users or Groups
  - 5.07.01 'Bulk Editing' Plan Permissions
- 5.08 Granting Global Permissions to Users or Groups
- 5.09 Granting Administration Rights to a User
- 5.10 Allowing Anonymous Users to access Bamboo
- 5.11 Enabling or disabling Contact Details Display
- 5.12 Enabling or disabling Public Signup
- 5.13 Working with External User Repositories
  - 5.13.1 Integrating Bamboo with Crowd
  - 5.13.2 Integrating Bamboo with LDAP

Bamboo Documentation Home
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.

Crowd integration with Bamboo

The text on the Security Settings page, under the 'Enable External User Management?' check-box, currently reads "Enable this option if you are delegating your user management to another user management system (e.g. Crowd)."

This is incorrect. Users should not have this box checked when using Crowd.

RELATED TOPICS

- 5.01 Creating a User
- 5.02 Changing a User's Password or Details
- 5.03 Deleting or deactivating a User
- 5.04 Creating a Group
- 5.05 Deleting a Group
- 5.06 Adding Users to and removing them from Groups
- 5.07 Granting Plan Permissions to Users or Groups
  - 5.07.01 'Bulk Editing' Plan Permissions
- 5.08 Granting Global Permissions to Users or Groups
- 5.09 Granting Administration Rights to a User
- 5.10 Allowing Anonymous Users to access Bamboo
- 5.11 Enabling or disabling Contact Details Display
- 5.12 Enabling or disabling Public Signup
- 5.13 Working with External User Repositories
  - 5.13.1 Integrating Bamboo with Crowd
  - 5.13.2 Integrating Bamboo with LDAP

Bamboo Documentation Home
5.13.2 Integrating Bamboo with LDAP

Bamboo supports LDAP integration for authentication and authorisation of LDAP users. However, please note that it is not possible to manage/administer LDAP accounts or user groups from Bamboo. Bamboo will continue to use local users and groups, even when LDAP is enabled. If you need to assign LDAP users to particular groups referenced by Bamboo (e.g. for permissions or notifications), the workaround is to assign your LDAP users to local Bamboo groups, and reference these groups rather than LDAP groups.

Also please note that:

- Once LDAP has been enabled, reverting back to local user management is not possible.
- In Bamboo versions prior to 1.2.2 it was not possible to perform XML backups of your Bamboo instance when integrated with LDAP. In Bamboo version 1.2.2, exports are possible, but user data will not be included in the export.
- Bamboo does not currently support multiple LDAP servers. If you need to connect to multiple LDAP servers, please consider using Crowd.

Before you begin

Confirm this information about your LDAP server:

1. Check your server LDAP version. Supported versions are v2 and v3. Supported LDAP servers include OpenLDAP, Microsoft Active Directory, Novell eDirectory, and any server that uses Java JNDI-LDAP mapping.
2. Your LDAP or Active Directory server must support static groups. This means that the user DNs must be stored against a membership attribute inside an LDAP group. An example of a static group is shown below:

```java
Dn: CN=Sales and Marketing,CN=Users,DC=ad,DC=atlassian,DC=com
objectClass: top; group;
 cn: Sales and Marketing;
distinguishedName: CN=Sales and Marketing,CN=Users,DC=ad,DC=atlassian,DC=com;
name: Sales and Marketing;
...
member: CN=John Smith,CN=Users,DC=ad,DC=atlassian,DC=com
member: CN=Sally Smith,CN=Users,DC=ad,DC=atlassian,DC=com
...
```

The membership attribute in this case is member, but this is not required. Note that the full DNs of John Smith and Sally Smith are listed. If the values against member are not full DNs, but are just usernames, then you need to add the flag `<useUnqualifiedUsernameForMembershipComparison>true</useUnqualifiedUsernameForMembershipComparison>` to your LDAP configuration. Open Directory on OS X uses this configuration.
3. You must not have an LDAP group called 'bamboo-admin'.
4. Make sure you don't have duplicate users on LDAP. In such cases, the LDAP users will take precedence over your local Bamboo users.
5. Make sure you don't have duplicate groups on LDAP as this may cause unpredictable behaviour.

Stage 1 - Backup your data

Please backup your data before attempting LDAP integration.

Stage 2 - Configure Connection Details

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

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

```xml
<ldap key= "ldapRepository"  name= "LDAP Repository@hecate.atlassian.com"  cache= "true">
<host>hecate.atlassian.com</host>
```
3. Please ensure that the following line is also active in your `atlassian-user.xml` (it should be there by default):

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

Stage 3 - Map LDAP Data Tree

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

```
<baseUserNamespace> dc=staff,dc=perftest,dc=atlassian,dc=private </baseUserNamespace>
<baseGroupNamespace> dc=groups,dc=perftest,dc=atlassian,dc=private </baseGroupNamespace>
$usernameAttribute> cn </usernameAttribute>
$userSearchFilter> (objectClass=inetorgperson) </userSearchFilter>
$firstnameAttribute> givenname </firstnameAttribute>
$surnameAttribute> sn </surnameAttribute>
$emailAttribute> mail </emailAttribute>
$groupnameAttribute> cn </groupnameAttribute>
$groupSearchFilter> (objectClass=groupOfNames) </groupSearchFilter>
$membershipAttribute> member </membershipAttribute>
```

Stage 4 - Optional LDAP Settings

The following settings do not appear in the default `atlassian-user.xml` file. Their default values are as follows:

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

If you want to override these default values, you can specify any or all of them by adding them onto the end of the `atlassian-user.xml` file. For example, to add your own value for the `<initSize>` setting, you would add an extra line before the `</ldap>` line shown in 'Stage 3' above:

```
<initSize>20</initSize>
```
It is important that the connection pool timeout value be set to 0, as this will force Atlassian User (via the JNDI layer) to clean up lingering connections that have lived past one request. For more information about LDAP pools please see http://java.sun.com/products/jndi/tutorial/ldap/connect/config.html.

Stage 5 - External user management

Ensure that External User Management is turned on in Bamboo before assigning LDAP users to Bamboo groups.

To enable external user management,

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 External User Management?' check-box.
4. Click the 'Save' button.

Stage 6 - Assigning LDAP Users to Bamboo Groups

Once Bamboo is started with LDAP enabled, you can assign LDAP users to Bamboo groups. Please see 5.06 Adding Users to and removing them from Groups.

Troubleshooting

To check whether atlassian-user.xml file is correctly configured, please run the paddle tool to debug the LDAP configuration in your atlassian-user.xml file. For further reference, please visit the Paddle usage page.

RELATED TOPICS

- 5.01 Creating a User
- 5.02 Changing a User's Password or Details
- 5.03 Deleting or deactivating a User
- 5.04 Creating a Group
- 5.05 Deleting a Group
- 5.06 Adding Users to and removing them from Groups
- 5.07 Granting Plan Permissions to Users or Groups
  - 5.07.01 'Bulk Editing' Plan Permissions
- 5.08 Granting Global Permissions to Users or Groups
- 5.09 Granting Administration Rights to a User
- 5.10 Allowing Anonymous Users to access Bamboo
- 5.11 Enabling or disabling Contact Details Display
- 5.12 Enabling or disabling Public Signup
- 5.13 Working with External User Repositories
  - 5.13.1 Integrating Bamboo with Crowd
  - 5.13.2 Integrating Bamboo with LDAP

Bamboo Documentation Home
6. Configuring Email and Instant Messaging Notifications

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

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

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

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

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

To enable notifications for a plan,

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

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

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

6. Click the 'Add' button.
7. Repeat steps 5 and 6 until you have added all the Notification Triggers that you wish to enable for this plan.
8. Click the 'Done' button if you are editing an existing plan; or if you are creating a new plan, click the 'Next' button and go to 7.6.1 Specifying Expiry for a Plan's Build Results.

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

Screenshot: 'Build Notifications'

To disable notifications for a plan,

1. Click 'Home' to go to the Dashboard.
2. Click the 'All Plans' tab.
3. Locate the plan in the list and click this icon:

4. The 'Configuration' tab will be displayed. Click the 'Build Notifications' tab.
5. For each 'Notification Trigger', click the corresponding 'Remove' link in the 'Operations' column.
6. Click the 'Done' button.

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

RELATED TOPICS

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

Bamboo Documentation Home
6.2 Configuring Bamboo to send SMTP Email

This page last changed on Jun 25, 2007 by rosie@atlassian.com.

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

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

To configure Bamboo to send SMTP email,

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

¹ Note re JNDI: As an alternative to specifying mail details directly in Bamboo, you can configure them in your application server (e.g. in the server.xml file — see 7.1 Locating Important Directories and Files), and then use JNDI to look up a preconfigured mail session. JNDI has the following advantages:

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

Screenshot: 'Email Server Details'
Now that you have configured Bamboo's SMTP email capability, you can specify notifications for a plan.

### RELATED TOPICS

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

[Bamboo Documentation Home](#)
6.3 Configuring Bamboo to use Instant Messaging (IM)

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.

Next step

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

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

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

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'.
   - If your IM Server uses an "@googlemail.com" account, type 'googlemail.com' in the 'Host' field.
5. Leave the 'Port' field blank, Bamboo will perform a DNS lookup to figure out which port to use.
6. In the 'Username' field, type the login name of the Google account from which IM notifications will be sent. Only the account name needs to be included e.g. atlassianbamboo NOT atlassianbamboo@gmail.com.
   - Do not select "Requires an SSL connection"
7. In the 'Password' field, type the password for the account specified in step 6.
8. Type a test IM user's address (e.g. atlassianbamboo@gmail.com NOT atlassianbamboo) in the 'Test Recipient Address' box. (Note: use a different user to the one you specified in step 6.)
9. Click the 'Test' button, and verify that the message is successfully received.
10. Click the 'Save' button.

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

Additional notes about using Google Talk:

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

RELATED TOPICS

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

Bamboo Documentation Home
07. Managing Data and Backups

7. Managing Data and Backups

- 7.1 Locating Important Directories and Files
- 7.2 Specifying Bamboo's Working Directory
- 7.3 Viewing your Database Connection Details
- 7.4 Moving your Bamboo Data to a different Database
- 7.5 Optimising or Re-indexing Data
- 7.6 Enabling Expiry of Build Results
  - 7.6.1 Specifying Expiry for a Plan's Build Results
- 7.7 Specifying a Backup Schedule
- 7.8 Exporting Data for Backup
- 7.9 Importing Data from Backup
7.1 Locating Important Directories and Files

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 7.5 Optimising or 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 8.1 Viewing Bamboo's System Information, and can be changed as described in 7.2 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 7.6 Enabling Expiry of Build Results). The location of this directory was specified via the Setup Wizard, and can be viewed as described in 8.1 Viewing Bamboo's System Information. Its contents can be backed up as per 7.8 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 8.1 Viewing Bamboo's System Information. Its contents can be backed up as per 7.8 Exporting Data for Backup.

**Bamboo installation directory**

- **webapp/WEB-INF/classes.bamboo-init.properties** — This file tells Bamboo where to find the Bamboo home directory. The location of this directory is specified by the Bamboo administrator as described in the Bamboo Installation Guide, and can be viewed as described in 8.1 Viewing Bamboo's System Information.
- **bamboo.sh** — This is the startup file for Bamboo Standalone under Unix/Linux, Solaris and Mac OS.
- **bamboo.bat** — This is the startup file for Bamboo Standalone under Windows.
- **bamboo.pid** — This file, under Linux, contains the Process ID for the running instance of Bamboo.
- **conf/wrapper.conf** — This file provides the means to configure Bamboo on startup, when using the Java Service wrapper under Linux or Windows.
- **scripts/** — This directory contains operational scripts, including scripts for CVS and SVN triggers.
- **wrapper/** — This directory contains the necessary files to start Bamboo using the Java Service wrapper (see the Mac and Linux installation guides).
- **logs/** — This directory contains logs written by the Java Service wrapper. (Note: The Bamboo server logs are written to the root of the installation directory. Build logs are stored in the xml-data/builds/sub-directories.)
- **webapp/** — This directory contains all the Bamboo server application files.
• `webapp/WEB-INF/lib/` — This directory is used when deploying Bamboo plugins. It also contains other libraries required by Bamboo.
• `webapp/WEB-INF/classes/log4j.properties` — This is Bamboo’s logging configuration file.
• `webapp/WEB-INF/classes/jetty.xml`* — This is the configuration file for Jetty, the application server that is bundled with Bamboo Standalone. The configuration format is simply a mapping from XML to Java. With this format you can call the methods defined in the Jetty Javadoc to configure the server.

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

RELATED TOPICS

• 7.1 Locating Important Directories and Files
• 7.2 Specifying Bamboo’s Working Directory
• 7.3 Viewing your Database Connection Details
• 7.4 Moving your Bamboo Data to a different Database
• 7.5 Optimising or Re-indexing Data
• 7.6 Enabling Expiry of Build Results
  • 7.6.1 Specifying Expiry for a Plan’s Build Results
• 7.7 Specifying a Backup Schedule
• 7.8 Exporting Data for Backup
• 7.9 Importing Data from Backup

Bamboo Documentation Home
7.2 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

- 7.1 Locating Important Directories and Files
- 7.2 Specifying Bamboo's Working Directory
- 7.3 Viewing your Database Connection Details
- 7.4 Moving your Bamboo Data to a different Database
- 7.5 Optimising or Re-indexing Data
- 7.6 Enabling Expiry of Build Results
  - 7.6.1 Specifying Expiry for a Plan's Build Results
- 7.7 Specifying a Backup Schedule
- 7.8 Exporting Data for Backup
- 7.9 Importing Data from Backup

Bamboo Documentation Home
7.3 Viewing your Database Connection Details

When you installed Bamboo, you would have set up a database connection by following one of these processes:

- **1. Using Bamboo's embedded HSQL database**
- **2. Connecting Bamboo to an external database**
  - **2.1 MySQL 4.1 and 5.0**
  - **2.2 Postgres 8+**
  - **2.3 Unsupported databases**

Once Bamboo is running, you can view the database configuration details as follows.

To view your database connection details,

1. Click the 'Administration' link in the top navigation bar.
2. Click the 'Database Configuration' link in the left navigation column.

**Screenshot: 'Database Configuration'**

<table>
<thead>
<tr>
<th>Database Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Driver Class Name:</td>
</tr>
<tr>
<td>Database URL:</td>
</tr>
<tr>
<td>User Name:</td>
</tr>
<tr>
<td>Hibernate Dialect:</td>
</tr>
</tbody>
</table>

**RELATED TOPICS**

- **7.1 Locating Important Directories and Files**
- **7.2 Specifying Bamboo's Working Directory**
- **7.3 Viewing your Database Connection Details**
- **7.4 Moving your Bamboo Data to a different Database**
- **7.5 Optimising or Re-indexing Data**
- **7.6 Enabling Expiry of Build Results**
  - **7.6.1 Specifying Expiry for a Plan's Build Results**
- **7.7 Specifying a Backup Schedule**
- **7.8 Exporting Data for Backup**
- **7.9 Importing Data from Backup**

**Bamboo Documentation Home**
7.4 Moving your Bamboo Data to a different Database

When you installed Bamboo, you would have set up a database connection by following one of these processes:

- 1. **Using Bamboo's embedded HSQL database**
- 2. **Connecting Bamboo to an external database**
  - 2.1 MySQL 4.1 and 5.0
  - 2.2 Postgres 8+
  - 2.3 Unsupported databases

You may later wish to use a different database. For example, the embedded HSQL database is suitable for evaluation purposes only — you would typically move to an external database before deploying Bamboo in production.

To move your Bamboo data to a different database,

1. Backup your Bamboo data as described in 7.8 Exporting Data for Backup. Note the filename and path of the exported file for use in Step 8 below.
2. Shut down your old instance of Bamboo.
3. If your old instance of Bamboo was configured to start automatically (e.g. as a Windows service), disable it.
4. Install a new instance of Bamboo as described in the Bamboo Installation Guide. Specify a different Home Directory* and Installation Directory* from the directories used by your old instance of Bamboo. (If you use the same locations, your existing data will be deleted.)
5. Launch your new instance of Bamboo. You will see the Setup Wizard.
6. At Step 1 of the Setup Wizard, ensure that your new Configuration Directory*, Build Data Directory* and Build Working Directory* are in different locations to your old instance of Bamboo.
7. At Step 2 of the Setup Wizard, select your new database and follow the appropriate instructions for your chosen database:
   - 2.1 MySQL 4.1 and 5.0
   - 2.2 Postgres 8+
   - 2.3 Unsupported databases
8. At Step 3 of the Setup Wizard (see screenshot below), select 'Import existing data' and specify the export file created in Step 1 above.
9. Wait while Bamboo imports your data. (You will not need to complete any more steps of the Setup Wizard.)
10. When the data import has finished, restart your new instance Bamboo.
11. Re-index your Bamboo data as described in 7.5 Optimising or Re-indexing Data.
12. Verify that your build results and system settings look the same as before.

*For information about the contents of these directories, please see 7.1 Locating Important Directories and Files

Screenshot: Setup Wizard — Step 3: 'Starting Data'
RELATED TOPICS

- 7.1 Locating Important Directories and Files
- 7.2 Specifying Bamboo's Working Directory
- 7.3 Viewing your Database Connection Details
- 7.4 Moving your Bamboo Data to a different Database
- 7.5 Optimising or Re-indexing Data
- 7.6 Enabling Expiry of Build Results
  - 7.6.1 Specifying Expiry for a Plan's Build Results
- 7.7 Specifying a Backup Schedule
- 7.8 Exporting Data for Backup
- 7.9 Importing Data from Backup

Bamboo Documentation Home
7.5 Optimising or Re-indexing Data

About optimising

You may want to optimise your Bamboo build results data if you notice that search-intensive operations (e.g. reporting) are becoming slow.

Information: Bamboo will still be accessible while the optimisation process is running.

About re-indexing

You will need to re-index your Bamboo build results data whenever you perform a data import. Re-indexing your data can also be helpful if your reports appear to be out-of-sync with your data.

Information: Bamboo will not be accessible while the re-indexing process is running. This may take a few minutes to complete (see System Information for an estimate of how long it will take).

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. The 'Optimise or Re-index Bamboo' screen will appear. Select 'Full re-index' and click the 'Perform' button.

To optimise 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. The 'Optimise or Re-index Bamboo' screen will appear. Select Optimise current index' and click the 'Perform' button.

RELATED TOPICS

- 7.1 Locating Important Directories and Files
- 7.2 Specifying Bamboo's Working Directory
- 7.3 Viewing your Database Connection Details
- 7.4 Moving your Bamboo Data to a different Database
- 7.5 Optimising or Re-indexing Data
- 7.6 Enabling Expiry of Build Results
  - 7.6.1 Specifying Expiry for a Plan's Build Results
- 7.7 Specifying a Backup Schedule
- 7.8 Exporting Data for Backup
- 7.9 Importing Data from Backup

Bamboo Documentation Home
7.6 Enabling Expiry of Build Results

By enabling build expiry, you can choose how much build results data will be kept in your Bamboo system, and for how long it will be kept (e.g. for reporting purposes), before being automatically deleted. If you disable build expiry, your build results will never be automatically deleted from Bamboo.

You can enable/disable build expiry for:

- all plans (as described below). This is generally the easiest way to manage your build expiry. Your settings will apply to all plans that do not have individual expiry settings.
- individual plans (see 7.6.1 Specifying Expiry for a Plan's Build Results). You would generally only do this if there is a specific reason to keep/delete a particular plan’s build results.

Note that you can also build results manually — see 1.4.1 Deleting a Build Result.

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

To enable expiry 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. The 'Build Expiry' screen will be displayed. Click the 'Enable' button.
4. The 'Build Expiry Settings' screen will be displayed (see screenshot below).
5. In the 'What should be expired' section, select what type of build results data you want to delete:
   - 'Artifacts' — select this if you want to delete all user-defined artifacts but keep all other build results data.
   - 'Build results' — select this if you want to delete all build results data (including artifacts).
6. Choose one of these three methods for specifying how much data to keep:
   - To keep all build results up to a certain age,
     - In the 'Expiry period' field, specify the number of months/weeks/days for which you want to keep your build results, e.g. specify '24 months' to keep all build results for the last two years.
     - In the 'Minimum builds to keep' field, specify '0'.
   - To keep a certain number of build results per plan,
     - In the 'Expiry period' field, specify '0'.
     - In the 'Minimum builds to keep' field, specify the number of build results you want to keep, e.g. specify '50' to keep the latest 50 build results for each plan.
   - To keep all build results up to a certain age, and a certain number of build results per plan,
     - In the 'Expiry period' field, specify the number of months/weeks/days for which you want to keep your build results data, e.g. specify '24 months' to keep all build results for the last two years.
     - In the 'Minimum builds to keep' field, specify the number of build results you want to keep, e.g. specify '50' to keep the latest 50 build results for each plan. (This means that, even if all of a plan’s builds are over two years old, the last 50 build results will not be deleted.)
7. In the 'Labels to keep' field, specify any labels for which you always want to keep labelled builds. (If you want to specify more than one label, use spaces to separate them.) For any label(s) that you specify, all builds that have a matching label will never be deleted, regardless of the method you followed in step 6 above.
   - Note: builds can either be labelled:
     - manually, as described in 8.2 Labelling a Build Result in the Bamboo User's Guide; or
     - automatically, as described in 1.9 Specifying Labels for a Plan's Build Results in the Bamboo Administrator's Guide.
8. Click the 'Save' button.

Screenshot: 'Enable Build Expiry'
To disable expiry 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. The 'Build Expiry' screen will be displayed (see screenshot below). Click the 'Disable' button.

Screenshot: 'Disable Build Expiry'

RELATED TOPICS

- 7.1 Locating Important Directories and Files
- 7.2 Specifying Bamboo's Working Directory
- 7.3 Viewing your Database Connection Details
- 7.4 Moving your Bamboo Data to a different Database
- 7.5 Optimising or Re-indexing Data
- 7.6 Enabling Expiry of Build Results
  - 7.6.1 Specifying Expiry for a Plan's Build Results
- 7.7 Specifying a Backup Schedule
- 7.8 Exporting Data for Backup
- 7.9 Importing Data from Backup

Bamboo Documentation Home
7.6.1 Specifying Expiry for a Plan's Build Results

By enabling build expiry, you can choose how much build results data will be kept in your Bamboo system, and for how long it will be kept (e.g. for reporting purposes), before being automatically deleted.

If you disable build expiry, your build results will never be automatically deleted from Bamboo.

You can enable/disable build expiry for:

- all plans (see 7.6 Enabling Expiry of Build Results). This is generally the easiest way to manage your build expiry. Your settings will apply to all plans that do not have individual expiry settings.
- individual plans (as described below). You would generally only do this if there is a specific reason to keep/delete a particular plan's build results.

Note that you can also delete build results manually — see 1.4.1 Deleting a Build Result.

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

To enable expiry 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. The 'Post Actions' tab will be displayed. Under 'Build Expiry', select 'Override global build expiry configuration'.
6. In the 'What should be expired' section, select what type of build results data you want to delete:
   - 'Artifacts' — select this if you want to delete all user-defined artifacts but keep all other build results data.
   - 'Build results' — select this if you want to delete all build results data (including artifacts).
7. Choose one of these three methods for specifying how much data to keep:
   - To keep all build results up to a certain age,
     - In the 'Expiry period' field, specify the number of months/weeks/days for which you want to keep your build results, e.g. specify '24 months' to keep all build results for the last two years.
     - In the 'Minimum builds to keep' field, specify '0'.
   - To keep a certain number of build results per plan,
     - In the 'Expiry period' field, specify '0'.
     - In the 'Minimum builds to keep' field, specify the number of build results you want to keep, e.g. specify '50' to keep the latest 50 build results for each plan.
   - To keep all build results up to a certain age, and a certain number of build results per plan,
     - In the 'Expiry period' field, specify the number of months/weeks/days for which you want to keep your build results, e.g. specify '24 months' to keep all build results for the last two years.
     - In the 'Minimum builds to keep' field, specify the number of build results you want to keep, e.g. specify '50' to keep the latest 50 build results for each plan. (This means that, even if all of a plan's builds are over two years old, the last 50 build results will not be deleted.)
8. In the 'Labels to keep' field, specify any labels for which you always want to keep labelled builds. (If you want to specify more than one label, use spaces to separate them.) For any label(s) that you specify, all builds that have a matching label will never be deleted, regardless of the method you followed in step 7 above.

Note: builds can either be labelled:
- manually, as described in 8.2 Labelling a Build Result in the Bamboo User's Guide; or
automatically, as described in 1.9 Specifying Labels for a Plan's Build Results in the Bamboo Administrator's Guide.

9. Click the 'Save' button if you are editing an existing plan; or if you are creating a new plan, either go to 1.9 Specifying Labels for a Plan's Build Results, or click the 'Next' button and go to 5.07 Granting Plan Permissions to Users or Groups.

Screenshot: 'Post Actions'

To disable expiry for a plan's build results data,

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. The 'Post Actions' tab will be displayed. Under 'Build Expiry', deselect 'Override global build expiry configuration'.
6. Click the 'Save' button.

RELATED TOPICS

- 7.1 Locating Important Directories and Files
- 7.2 Specifying Bamboo's Working Directory
- 7.3 Viewing your Database Connection Details
- 7.4 Moving your Bamboo Data to a different Database
- 7.5 Optimising or Re-indexing Data
- 7.6 Enabling Expiry of Build Results
  - 7.6.1 Specifying Expiry for a Plan's Build Results
- 7.7 Specifying a Backup Schedule
- 7.8 Exporting Data for Backup
- 7.9 Importing Data from Backup

Bamboo Documentation Home
7.7 Specifying a Backup Schedule

You can configure Bamboo to automatically create a backup each night, rather than doing a manual export every time.

Before you begin:

- Bamboo will be unavailable while the backup 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.
- Bamboo will not export if it detects that plans are currently being built. Please ensure that you do not have plans building (see 1.2 Viewing Bamboo's Current Activity).
- There is a bug in Bamboo 1.2 which causes the dependency associations to not be exported along with the plans. If you are using Bamboo 1.2, please upgrade to Bamboo 1.2.1 for a fix.

To specify a backup schedule,

1. Click the 'Administration' link in the top navigation bar.
2. Click the 'Scheduled Backups' link in the left navigation column.
3. The 'Scheduled Backups' screen will be displayed. Click the 'Edit' button.
4. In the 'Backup path' field, specify the directory where you want to store your backups. Each backup will be stored as a single file.
   - Note that backups may require large amounts of disk space. Please make sure you have enough disk space before proceeding.
5. In the 'Backup file prefix' field, specify the first part of the filename for all your backup files.
6. In the 'Backup file date pattern' field, specify the date/time format for identifying your individual backup files. This will be appended to the 'Backup file prefix' to form the complete filename for your backup files, e.g.:
   - if your 'Backup file prefix' is 'bamboo_backup_' and your 'Backup file date pattern' is 'yyyy_MM_dd', then your backup file on 31st July 2007 would be named 'bamboo_backup_2007_07_31'.
7. In the 'Backup Frequency' field, specify how often (in days) you want Bamboo to create a backup, e.g.:
   - specify '1' to create a backup every day.
   - specify '7' to create a backup every week.
8. In the 'Backup Time field, specify the time of day at which you want Bamboo to create a backup. Use 24-hour format, e.g. to create a backup at 8.30 PM you would specify '20:30'.
   - Note that Bamboo will be unavailable for the duration of the backup, so choose a time of day or night when usage is low.
9. Select the 'Backup Artifacts' check-box if you want to backup your build artifacts.
10. Click the 'Save' button.
   - Your first backup will run when your server's clock matches the time specified in the 'Backup Time' field.

To disable a scheduled backup,

1. Click the 'Administration' link in the top navigation bar.
2. Click the 'Scheduled Backups' link in the left navigation column.
3. The 'Scheduled Backups' screen will be displayed, showing your current backup schedule. Click the 'Edit' button.
4. Select the 'Disable scheduled backups' check-box.
   - Your schedule details will be retained, but no automatic backups will be performed.
5. Click the 'Save' button.
Screenshot: 'Scheduled Backups'

RELATED TOPICS

- 7.1 Locating Important Directories and Files
- 7.2 Specifying Bamboo's Working Directory
- 7.3 Viewing your Database Connection Details
- 7.4 Moving your Bamboo Data to a different Database
- 7.5 Optimising or Re-indexing Data
- 7.6 Enabling Expiry of Build Results
  - 7.6.1 Specifying Expiry for a Plan's Build Results
- 7.7 Specifying a Backup Schedule
- 7.8 Exporting Data for Backup
- 7.9 Importing Data from Backup

Bamboo Documentation Home
7.8 Exporting Data for Backup

Before you begin:

- Bamboo will be unavailable while 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.
- Bamboo will not export if it detects that plans are currently being built. Please ensure that you do not have plans building (see 1.2 Viewing Bamboo's Current Activity).
- There is a bug in Bamboo 1.2 which causes the dependency associations to not be exported along with the plans. If you are using Bamboo 1.2, please upgrade to Bamboo 1.2.1 for a fix.
- Bamboo uses the third party TrueZip library to create zip archives. TrueZIP currently implements the ZIP32 specification only. This limits the maximum ZIP file length to 4GB. Unfortunately, Bamboo exports will fail if the resulting ZIP file is over 4GB. As a workaround, please export Bamboo without artifacts.
- Exporting Bamboo doesn't work with Bamboo versions prior to 1.2.2, if LDAP is turned on. If you want to export Bamboo data, please turn off LDAP. Please see 5.13.2 Integrating Bamboo with LDAP for further details.

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, 'c:/Documents and Settings/<me>/bamboo/bamboohome/manual_backups/export.zip'.
4. Select the 'Export Artifacts' check-box if you want to backup your build artifacts.
5. Click the 'Export' button.

Screenshot: Export

Related Topics

- 7.1 Locating Important Directories and Files
- 7.2 Specifying Bamboo's Working Directory
- 7.3 Viewing your Database Connection Details
- 7.4 Moving your Bamboo Data to a different Database
- 7.5 Optimising or Re-indexing Data
- 7.6 Enabling Expiry of Build Results
7.6.1 Specifying Expiry for a Plan's Build Results

- 7.7 Specifying a Backup Schedule
- 7.8 Exporting Data for Backup
- 7.9 Importing Data from Backup

Bamboo Documentation Home
7.9 Importing Data from Backup

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.

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.
7. After the import is complete,
   - check the paths of your builders and JDK.
   - index your data.

Screenshot: Import

RELATED TOPICS

- 7.1 Locating Important Directories and Files
- 7.2 Specifying Bamboo's Working Directory
- 7.3 Viewing your Database Connection Details
- 7.4 Moving your Bamboo Data to a different Database
- 7.5 Optimising or Re-indexing Data
- 7.6 Enabling Expiry of Build Results
7.6.1 Specifying Expiry for a Plan's Build Results

7.7 Specifying a Backup Schedule

7.8 Exporting Data for Backup

7.9 Importing Data from Backup

Bamboo Documentation Home
8. Configuring System Settings

- 8.1 Viewing Bamboo's System Information
- 8.2 Updating your Bamboo License Details
- 8.3 Specifying Bamboo's Title
- 8.4 Specifying Bamboo's URL
- 8.5 Enabling GZIP Compression
- 8.6 Enabling Bamboo's Remote API
- 8.7 Enabling JIRA integration
8.1 Viewing Bamboo's System Information

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

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

This page contains useful information for you to send to Atlassian when requesting support.

Screenshot: 'System Information'

For information about the 'Bamboo Paths' please see 7.1 Locating Important Directories and Files.

RELATED TOPICS

- 8.1 Viewing Bamboo's System Information
- 8.2 Updating your Bamboo License Details
- 8.3 Specifying Bamboo's Title
- 8.4 Specifying Bamboo's URL
- 8.5 Enabling GZIP Compression
- 8.6 Enabling Bamboo's Remote API
- 8.7 Enabling JIRA integration

Bamboo Documentation Home
8.2 Updating your Bamboo License Details

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

To update your Bamboo license key,

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

Licensing Questions?
Please see the Licensing FAQ.

RELATED TOPICS

• 8.1 Viewing Bamboo's System Information
• 8.2 Updating your Bamboo License Details
• 8.3 Specifying Bamboo's Title
• 8.4 Specifying Bamboo's URL
• 8.5 Enabling GZIP Compression
• 8.6 Enabling Bamboo's Remote API
• 8.7 Enabling JIRA integration

Bamboo Documentation Home
8.3 Specifying Bamboo's Title

Bamboo’s name is the displayed title of this installation of Bamboo. It will appear throughout Bamboo (e.g. on the Dashboard), and in the window-title of your users’ browsers.

To specify Bamboo’s title,

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 'Name field, type the display title for your Bamboo server (e.g. "MyCompany’s Bamboo").
4. Click the 'Save' button.

RELATED TOPICS

- [8.1 Viewing Bamboo's System Information](#)
- [8.2 Updating your Bamboo License Details](#)
- [8.3 Specifying Bamboo's Title](#)
- [8.4 Specifying Bamboo's URL](#)
- [8.5 Enabling GZIP Compression](#)
- [8.6 Enabling Bamboo's Remote API](#)
- [8.7 Enabling JIRA integration](#)

[Bamboo Documentation Home](#)
8.4 Specifying Bamboo's URL

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

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 Viewing Bamboo's System Information
• 8.2 Updating your Bamboo License Details
• 8.3 Specifying Bamboo's Title
• 8.4 Specifying Bamboo's URL
• 8.5 Enabling GZIP Compression
• 8.6 Enabling Bamboo's Remote API
• 8.7 Enabling JIRA integration

Bamboo Documentation Home
8.5 Enabling GZIP Compression

You can enable GZIP compression in order to reduce the size of Bamboo's web pages. This is useful if Bamboo is being run over slow networks. There is a slight performance penalty, and note that GZIP may not work for languages other than English.

To enable GZIP Compression,

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

RELATED TOPICS

- 8.1 Viewing Bamboo's System Information
- 8.2 Updating your Bamboo License Details
- 8.3 Specifying Bamboo's Title
- 8.4 Specifying Bamboo's URL
- 8.5 Enabling GZIP Compression
- 8.6 Enabling Bamboo's Remote API
- 8.7 Enabling JIRA integration

Bamboo Documentation Home
8.6 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 Viewing Bamboo's System Information
- 8.2 Updating your Bamboo License Details
- 8.3 Specifying Bamboo's Title
- 8.4 Specifying Bamboo's URL
- 8.5 Enabling GZIP Compression
- 8.6 Enabling Bamboo's Remote API
- 8.7 Enabling JIRA integration

Bamboo Documentation Home
8.7 Enabling JIRA integration

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

Note that this will require your Bamboo server to login to your JIRA server, regardless of whether your JIRA server is running in 'public' or 'private' mode. Ensure that the Remote API option is enabled on your JIRA server.

JIRA integration with versions prior to JIRA 3.7:

• Bamboo offers limited integration with JIRA versions prior to 3.7 - Bamboo will create links to JIRA issue and you should still be able to click through issues in JIRA. However, Bamboo will not retrieve details of JIRA issues and display them alongside JIRA issue numbers within Bamboo.
• When configuring a JIRA server in Bamboo, do not try to validate (and retrieve issues), but simply click the 'save' button and Bamboo will start generating JIRA links for your Bamboo builds.

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 Viewing Bamboo's System Information
• 8.2 Updating your Bamboo License Details
• 8.3 Specifying Bamboo's Title
• 8.4 Specifying Bamboo's URL
• 8.5 Enabling GZIP Compression
• 8.6 Enabling Bamboo's Remote API
• 8.7 Enabling JIRA integration

Bamboo Documentation Home
09. Configuring Plugins

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

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
- 'NAnt Builder' plugin

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

RELATED TOPICS

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

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

If the 'Auto-Favourites' plugin is enabled, then a plan will be automatically added to a user's 'My Bamboo' tab when the user checks-in code to the plan's source-code repository, and the user will become a 'watcher' of that plan (i.e. depending on how the plan's notifications are configured, they may receive notifications about the plan's build results).

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.

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

RELATED TOPICS

- [9.1 About Bamboo Plugins](#)
- [9.2 Enabling the 'Auto-Favourites' Plugin](#)
- [9.3 Installing a new Plugin](#)

Bamboo Documentation Home
9.3 Installing a new Plugin

To install a new plugin,

1. Copy the plugin (i.e. JAR file) into the following directory:
   - If you are using Bamboo Standalone: ../<Bamboo-install>/webapp/WEB-INF/lib/
   - If you are using the Bamboo EAR-WAR distribution: ../<Bamboo-Deploy-Location>/WEB-INF/lib/ (e.g. if you are running Bamboo EAR-WAR on Tomcat, copy your plugin to the ../<Tomcat-Install>/webapps/<Bamboo-Deploy-Location>/WEB-INF/lib/ directory).
2. Restart Bamboo.

A number of plugins are available from the library. You can also create your own as described in the Bamboo Plugin Developer’s Guide.

RELATED TOPICS

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

Bamboo Documentation Home
Appendix A. Embedding Bamboo into Other Applications

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

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

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

To use this widget

1. Include the style sheet in your html document

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

2. Place the following script tag in your html

   • For all plans

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

   • For your favourite plans

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

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

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

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

Example

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

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

   `<script type="text/javascript" src="http://opensource.bamboo.atlassian.com/js/jsBuildSummaryAll.action"></script>`

RELATED TOPICS

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

Bamboo Documentation Home
**Latest Builds**

This page last changed on May 14, 2007 by rosie@atlassian.com.

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/](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>`

**RELATED TOPICS**

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

[Bamboo Documentation Home](#)
Latest Status of a Plan

This page last changed on May 14, 2007 by rosie@atlassian.com.

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>">&gt;
   </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>

RELATED TOPICS

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

Bamboo Documentation Home
My Latest Changes

This page last changed on May 14, 2007 by rosie@atlassian.com.

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

To use this widget

1. Include the style sheet in your html document

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

2. Place the following script tag in your html

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

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

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

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

Example

   <link rel="stylesheet" type="text/css" href="http://localhost:8085/styles/bamboo-widget.css">
   <script type="text/javascript" src="http://localhost:8085/js/myChanges.action?
   os_username=admin&os_password=admin"></script>

<table>
<thead>
<tr>
<th>Build</th>
<th>When</th>
<th>Comments</th>
<th>Test Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUCBUILD-DEF-88</td>
<td>6 minutes ago</td>
<td>28th time lucky!</td>
<td>[pass] 2 passed</td>
</tr>
<tr>
<td>SUCBUILD-DEF-89</td>
<td>14 minutes ago</td>
<td>Trying to stop build from failing (TST-115)</td>
<td>[pass] 4 out of 6 failed!</td>
</tr>
<tr>
<td>SUCBUILD-DEF-82</td>
<td>17 minutes ago</td>
<td>Added test failure (TST-110)</td>
<td>[pass] 4 out of 6 failed!</td>
</tr>
<tr>
<td>SUCBUILD-DEF-81</td>
<td>32 minutes ago</td>
<td>Added failing test to original test suite</td>
<td>[pass] 4 out of 6 failed!</td>
</tr>
<tr>
<td>SUCBUILD-DEF-75</td>
<td>1 hour ago</td>
<td>*** empty log message ***</td>
<td>[pass] 4 out of 6 failed!</td>
</tr>
<tr>
<td>SUCBUILD-DEF-74</td>
<td>3 days ago</td>
<td>Brady's change is related to TST-1</td>
<td>[pass] 4 out of 6 failed!</td>
</tr>
<tr>
<td>SUCBUILD-DEF-73</td>
<td>3 days ago</td>
<td>Trying to break build</td>
<td>[pass] 4 out of 6 failed!</td>
</tr>
<tr>
<td>AMP-DEF-10</td>
<td>1 week ago</td>
<td>*** empty log message ***</td>
<td>[pass] 3 passed</td>
</tr>
<tr>
<td>TEST-DEF-60</td>
<td>1 week ago</td>
<td>*** empty log message ***</td>
<td>[pass] 2 out of 4 failed!</td>
</tr>
<tr>
<td>SUCBUILD-DEF2-14</td>
<td>1 week ago</td>
<td>*** empty log message ***</td>
<td>[pass] 2 passed</td>
</tr>
</tbody>
</table>

RELATED TOPICS

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

Bamboo Documentation Home
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/

<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>

RELATED TOPICS

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

Bamboo Documentation Home
Bamboo Development Hub

This page last changed on May 24, 2007 by rosie@atlassian.com.

The Bamboo Development Hub is for people who are looking to develop their own plugins. For pre-built plugins please visit the library.

- Bamboo API
- Bamboo Plugin Developer's Guide
  - Build Complete Action Module
  - Build Processor Module
  - Builder Plugin Module
  - Getting Started
  - Index Reader Module
  - Notification Condition Module
  - Post Build Index Writer Module
  - Pre Build Action Module
  - Report Module
  - Source Repository 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
- Setting up Bamboo Development Environment in IDEA

See also: 9.3 Installing a new Plugin
Bamboo API

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

- Bamboo API
- Bamboo Plugin Developer's Guide
  - Build Complete Action Module
  - Build Processor Module
  - Builder Plugin Module
  - Getting Started
  - Index Reader Module
  - Notification Condition Module
  - Post Build Index Writer Module
  - Pre Build Action Module
  - Report Module
  - Source Repository 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
- Setting up Bamboo Development Environment in IDEA

Bamboo Documentation Home
Bamboo Plugin Developer's Guide

Bamboo Plugins Overview

A Bamboo plugin is a single JAR containing code, a plugin descriptor (XML) and usually some Freemarker template files to render HTML.

The plugin descriptor is the only mandatory part of the plugin. It must be called atlassian-plugin.xml and be located in the root of your JAR file.

Each plugin consists of one or more plugin modules. These are of different types (for example a report, or a post-build action) and each has an individual XML element describing it. Each module is described below together with the XML element required for it.

Here is a sample of the descriptor with highlighted elements:

```xml
<!-- the plugin key must be unique, think of it as the 'package' of the plugin -->
<atlassian-plugin key= "com.atlassian.plugin.sample" name= "Sample Plugin">
<!-- a short block describing the plugin itself -->
<plugin-info>
<description> This is a brief textual description of the plugin </description>
<!-- the version of the plugin -->
<version> 1.1 </version>
<!-- the versions of the application this plugin is for -->
<application-version min= "3.0" max= "3.0"/>
<!-- details of the plugin vendor -->
<vendor name= "Atlassian Software Systems Pty Ltd" url= "http://www.atlassian.com"/>
</plugin-info>

... 1 or more plugin modules ...
</atlassian-plugin>
```

Each plugin has a plugin key which is unique among all plugins (eg "com.atlassian.plugin.sample"). Semantically this equates to the package of a Java class. Each module within the plugin also has a module key which is unique within the plugin (eg "myreport"). Semantically this equates to the class name of a Java class.

The plugin key + module key are combined to make the complete key of the plugin module (combining the examples above, the complete key would be "com.atlassian.plugin.sample:myreport"). Note: a : is used to separate the plugin key from the module key.

Setting up a Bamboo Plugin Project

Refer to the Getting Started guide.

Deploy a Bamboo Plugin

Installing plugins in Bamboo is easy.

Once you have downloaded or created your plugin jar, follow these steps:

1. Shut down Bamboo
2. copy '$MY_COOL_PLUGIN.jar' into '/.../webapp/WEB-INF/lib/'
3. Start up Bamboo. Your plugin should be automatically installed and activated.
4. Enjoy!

Bamboo Plugin Module Types

The following types of plugin modules are supported by Bamboo

<table>
<thead>
<tr>
<th>Module Type</th>
<th>Since version...</th>
<th>Documentation</th>
<th>Description</th>
</tr>
</thead>
</table>

Document generated by Confluence on Dec 06, 2007 19:00
<table>
<thead>
<tr>
<th>Component</th>
<th>Version</th>
<th>Module(s)</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>builder</td>
<td>1.0</td>
<td>Builder Plugin Module</td>
<td>Add new builders to Bamboo</td>
</tr>
<tr>
<td>xwork</td>
<td>1.0</td>
<td>XWork Plugin Module</td>
<td>XWork actions and views bundled with the plugin. This enables building generic user interfaces.</td>
</tr>
<tr>
<td>report</td>
<td>1.0</td>
<td>Report Module</td>
<td>Defines a report of build telemetry data.</td>
</tr>
<tr>
<td>preBuildAction</td>
<td>1.1</td>
<td>Pre Build Action Module</td>
<td>Prepends a custom synchronous process to the build. Occurs before the builder has run.</td>
</tr>
<tr>
<td>buildProcessor</td>
<td>1.0</td>
<td>Build Processor Module</td>
<td>Append a custom synchronous process to the build. Occurs after the builder has run.</td>
</tr>
<tr>
<td>buildCompleteAction</td>
<td>1.0</td>
<td>Build Complete Action Module</td>
<td>Add a custom asynchronous action after the build process has completed.</td>
</tr>
<tr>
<td>postBuildIndexWriter</td>
<td>1.0</td>
<td>Post Build Index Writer Module</td>
<td>Writes custom build data into the index for report generation.</td>
</tr>
<tr>
<td>indexReader</td>
<td>1.0</td>
<td>Index Reader Module</td>
<td>Reads out custom index information written by the postBuildIndexWriter from the index.</td>
</tr>
<tr>
<td>web-item</td>
<td>1.0</td>
<td>Web Item Module</td>
<td>Add new links to the Bamboo interface</td>
</tr>
<tr>
<td>web-section</td>
<td>1.0</td>
<td>Web Section Module</td>
<td>Add a new section to the Bamboo interface</td>
</tr>
<tr>
<td>web-resource</td>
<td>1.0</td>
<td>[Web Resource Module]</td>
<td>Add a new resource to the Bamboo application (e.g. javascript)</td>
</tr>
<tr>
<td>notificationCondition</td>
<td>1.1</td>
<td>Notification Condition Module</td>
<td>Add new notification condition</td>
</tr>
<tr>
<td>repository</td>
<td>1.1</td>
<td>Source Repository Module</td>
<td>Add a custom source repository</td>
</tr>
</tbody>
</table>

**Built-in Bamboo system plugins**

A number of functions and areas within Bamboo are shipped as built in plugins. These can also be useful for plugin developers who want to know more about how to create their own plugins, as they showcase the functionality that can be built.

The system plugins are referenced from the following files (located in `/WEB-INF/classes`):

- `system-builder-plugin.xml` - the built in builders, including Ant, Maven, and Maven 2.
- `system-clover-plugin.xml` - the built in Clover analytics.
• system-jira-plugin.xml - the built in tab view of JIRA issues in a build.
• system-labelling-plugin.xml - the built in automatic build labeller.
• system-reports-plugin.xml - the built reports of builds grouped by time periods under the Reports tab.
• system-repository-plugin.xml - definition for the CVS, SVN and Perforce repositories shipped with Bamboo.
• system-webUI-plugin.xml - the built in menu items under the Administration tab as well as the tab menu items on the View Plan page and the View Build Results page.
• system-notifications-plugin.xml - the built in notification conditions, including all builds, failed builds, after X failed builds.

RELATED TOPICS

• Bamboo API
  • Bamboo Plugin Developer's Guide
    ° Build Complete Action Module
    ° Build Processor Module
    ° Builder Plugin Module
    ° Getting Started
    ° Index Reader Module
    ° Notification Condition Module
    ° Post Build Index Writer Module
    ° Pre Build Action Module
    ° Report Module
    ° Source Repository 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
  • Setting up Bamboo Development Environment in IDEA

Bamboo Documentation Home
Build Complete Action Module

Description

Like the Build Processor Module, this allows you to specify a custom action to take place. However, the difference is that this will run after the full build result has been registered. The build is deemed to have completed before the BuildCompleteAction is fired. Hence, build complete actions cannot impact the state of the build.

Interface

Build Complete Action modules must implement the com.atlassian.bamboo.build.CustomBuildCompleteAction interface.

Sample Module Descriptor Element

```xml
<buildCompleteAction key= "autoLabeller" name= "Build Automatic Labeller"
class= "com.atlassian.bamboo.labels.AutoLabelBuildCompleteAction">
  <resource type= "freemarker" name= "edit" location= "templates/plugins/buildCompleteAction/autoLabellingEdit.ftl"/>
  <resource type= "freemarker" name= "view" location= "templates/plugins/buildCompleteAction/autoLabellingView.ftl"/>
  <description> An automatic labelling plugin. </description>
</buildCompleteAction>
```

RELATED TOPICS

- Bamboo API
- Bamboo Plugin Developer's Guide
  - Build Complete Action Module
  - Build Processor Module
  - Builder Plugin Module
  - Getting Started
  - Index Reader Module
  - Notification Condition Module
  - Post Build Index Writer Module
  - Pre Build Action Module
  - Report Module
  - Source Repository 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
- Setting up Bamboo Development Environment in IDEA

Bamboo Documentation Home
Build Processor Module

This page last changed on Jun 19, 2007 by ahempe.

Description

The BuildProcessor module allows you to define a custom process that runs during the build.

This will occur immediately after the builder has completed execution and the test results have been captured. The BuildProcessor forms part of build execution run and the result of the execution is only registered after your custom BuildProcessor has completed. This means that your plugin has the capability to affect the the final BuildState of your build (i.e. success/failure).

Interface

Build Processor modules must implement the com.atlassian.bamboo.build.CustomBuildProcessor interface.

Sample Module Descriptor Element

```xml
<buildProcessor key="cloverResultCollector" name="Clover Results Collector" class="com.atlassian.bamboo.builder.coverage.CloverBuildProcessor">
  <skipIfFailed>true</skipIfFailed>
  <resource type="freemarker" name="edit" location="templates/plugins/buildProcessor/cloverProcessorEdit.ftl"/>
  <resource type="freemarker" name="view" location="templates/plugins/buildProcessor/cloverProcessorView.ftl"/>
  <description>A Clover report parser and data collector</description>
</buildProcessor>
```

RELATED TOPICS

- Bamboo API
- Bamboo Plugin Developer's Guide
  - Build Complete Action Module
  - Build Processor Module
  - Builder Plugin Module
  - Getting Started
  - Index Reader Module
  - Notification Condition Module
  - Post Build Index Writer Module
  - Pre Build Action Module
  - Report Module
  - Source Repository 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
- Setting up Bamboo Development Environment in IDEA

Bamboo Documentation Home
Builder Plugin Module

Description

A plugin module which defines a builder in Bamboo, such as Maven, Maven2, or Ant.

Interface

Builder modules must implement the com.atlassian.bamboo.builder.Builder interface.

Sample Module Descriptor Element

```xml
<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>
```

RELATED TOPICS

- Bamboo API
- Bamboo Plugin Developer's Guide
  - Build Complete Action Module
  - Build Processor Module
  - Builder Plugin Module
  - Getting Started
  - Index Reader Module
  - Notification Condition Module
  - Post Build Index Writer Module
  - Pre Build Action Module
  - Report Module
  - Source Repository 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
- Setting up Bamboo Development Environment in IDEA

Bamboo Documentation Home
Getting Started

This page last changed on Aug 21, 2007 by asridhar.

Sun JAVA Libraries

Due to licensing restrictions, we are not allowed to re-distribute native Sun JAVA libraries through our maven2 public repositories. Please visit to our Confluence page on coping with Sun JAVA Libraries for further details.

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 from our public SVN Repository.


• 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.

RELATED TOPICS

• Bamboo API
• Bamboo Plugin Developer’s Guide
  • Build Complete Action Module
  • Build Processor Module
  • Builder Plugin Module
  • Getting Started
  • Index Reader Module
  • Notification Condition Module
  • Post Build Index Writer Module
  • Pre Build Action Module
  • Report Module
  • Source Repository 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
• Setting up Bamboo Development Environment in IDEA

Bamboo Documentation Home
**Index Reader Module**

This page last changed on Jun 19, 2007 by ahempel.

**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.

**Interface**

Index Reader modules must implement the com.atlassian.bamboo.index.CustomIndexReader interface.

**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>
```

**RELATED TOPICS**

- Bamboo API
- Bamboo Plugin Developer's Guide
  - Build Complete Action Module
  - Build Processor Module
  - Builder Plugin Module
  - Getting Started
  - Index Reader Module
  - Notification Condition Module
  - Post Build Index Writer Module
  - Pre Build Action Module
  - Report Module
  - Source Repository 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
- Setting up Bamboo Development Environment in IDEA

[Bamboo Documentation Home](#)
Notification Condition Module

This page last changed on Jun 19, 2007 by ahempel.

Description

A plugin module which allows you to define your own notification condition

You can use this plugin to

- Set conditions as to whether or not to send a notification
- Generate customised notification content

Interface

Notification Condition modules must implement the com.atlassian.bamboo.notification.NotificationCondition interface.

Sample Module Descriptor Element

```xml
<notification-condition key="buildCompleted.XFailedBuilds" name="After X Failed Builds Completed" class="com.atlassian.bamboo.notification.conditions.AfterXFailedBuildsCondition">
  <description>Send Notification After X Failed Builds</description>
  <resource type="freemarker" name="edit" location="templates/plugins/notifications/afterXFailedEdit.ftl"/>
  <resource type="freemarker" name="view" location="templates/plugins/notifications/afterXFailedView.ftl"/>
</notification-condition>
```

Other Information regarding the NotificationCondition class

Several of the interface methods for this class accept an Event object as a parameter. Current functionality only allows this to be a BuildCompletedEvent. This event will contain the Build object, BuildResults object and BuildResultsSummary object for you to use.

RELATED TOPICS

- Bamboo API
- Bamboo Plugin Developer's Guide
  - Build Complete Action Module
  - Build Processor Module
  - Builder Plugin Module
  - Getting Started
  - Index Reader Module
  - Notification Condition Module
  - Post Build Index Writer Module
  - Pre Build Action Module
  - Report Module
  - Source Repository 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
- Setting up Bamboo Development Environment in IDEA

Bamboo Documentation Home
Post Build Index Writer Module

This page last changed on Jun 19, 2007 by ahempel.

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.

Interface

Post Build Index Writer modules must implement the com.atlassian.bamboo.index.CustomPostBuildIndexWriter interface.

Sample Module Descriptor Element

```xml
<postBuildIndexWriter key="cloverIndexWriter"  name= "Write Clover Result to Index" class= "com.atlassian.bamboo.builder.coverage.CloverPostBuildIndexWriter" >
  <description> Writes the clover result in a build results to an index document </description>
</postBuildIndexWriter>
```

RELATED TOPICS

- Bamboo API
- Bamboo Plugin Developer's Guide
  - Build Complete Action Module
  - Build Processor Module
  - Builder Plugin Module
  - Getting Started
  - Index Reader Module
  - Notification Condition Module
  - Post Build Index Writer Module
  - Pre Build Action Module
  - Report Module
  - Source Repository 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
- Setting up Bamboo Development Environment in IDEA

Bamboo Documentation Home
Pre Build Action Module

This page last changed on Jun 19, 2007 by ahempel.

Description

The PreBuildAction module allows you to define a custom process that runs before your build begins. This will occur immediately before the builder begins execution. The PreBuildAction will have access to the BuildResults object which contains the information for the build.

Interface

Pre Build Action modules must implement the com.atlassian.bamboo.build.CustomPreBuildAction interface.

Sample Module Descriptor Element

```xml
<preBuildAction key= "vcsVersion" name= "VCS Version Collector" class= "com.atlassian.bamboo.vcsversion.VCSVersionReader">  
<description> A custom action that reads the identifier of a source repository version and stores it into the custom data map of a build. </description>  
<resource type= "freemarker" name= "edit" location= "templates/plugins/preBuildAction/vcsVersionReaderEdit.ftl"/>  
<resource type= "freemarker" name= "view" location= "templates/plugins/preBuildAction/vcsVersionReaderView.ftl"/> 
</preBuildAction>
```

RELATED TOPICS

- Bamboo API
- Bamboo Plugin Developer's Guide
  - Build Complete Action Module
  - Build Processor Module
  - Builder Plugin Module
  - Getting Started
  - Index Reader Module
  - Notification Condition Module
  - Post Build Index Writer Module
  - Pre Build Action Module
  - Report Module
  - Source Repository 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
- Setting up Bamboo Development Environment in IDEA

Bamboo Documentation Home
Report Module

This page last changed on May 14, 2007 by rosie@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>
```

RELATED TOPICS

- Bamboo API
- Bamboo Plugin Developer's Guide
  - Build Complete Action Module
  - Build Processor Module
  - Builder Plugin Module
  - Getting Started
  - Index Reader Module
  - Notification Condition Module
  - Post Build Index Writer Module
  - Pre Build Action Module
  - Report Module
  - Source Repository 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
- Setting up Bamboo Development Environment in IDEA

Bamboo Documentation Home
Source Repository Module

This page last changed on May 14, 2007 by rosie@atlassian.com.

Description

A plugin module which defines a repository in Bamboo, such as CVS, Subversion, or Perforce.

Sample Module Descriptor Element

```xml
<repository key="svn" name="SVN Repository" class="com.atlassian.bamboo.repository.svn.SvnRepository">
  <description>A Subversion Repository</description>
  <resource type="freemarker" name="edit" location="templates/plugins/repository/svnRepositoryEdit.ftl"/>
  <resource type="freemarker" name="view" location="templates/plugins/repository/svnRepositoryView.ftl"/>
</repository>
```

Notes

Plugins of this type must implement the com.atlassian.bamboo.repository.Repository interface. For (comparative) simplicity, you should use the class AbstractRepository as a starting point and extend from that.

RELATED TOPICS

- **Bamboo API**
- **Bamboo Plugin Developer's Guide**
  - Build Complete Action Module
  - Build Processor Module
  - Builder Plugin Module
  - Getting Started
  - Index Reader Module
  - Notification Condition Module
  - Post Build Index Writer Module
  - Pre Build Action Module
  - Report Module
  - Source Repository 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
- **Setting up Bamboo Development Environment in IDEA**

Bamboo Documentation Home
Web Item Module

This page last changed on May 14, 2007 by rosie@atlassian.com.

Description

The WebItem allows you to define a link in the Bamboo system. (Usually in some form of menu).

Currently, you can use the web-item to add links to three locations

- The Administration Menu
- The Plan Sub Menu (tabs on the View Plan page)
- The Results Sub Menu (tabs on the View Build Results page)

Sample Module Descriptor Element

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

Module Components

- **key** - this is the unique identifier of the web-item, it is also used by Bamboo to give the link an id.
- **name** - in the plan sub menu and results sub menu this is used to determine if the current link (tab) is active.
- **section** - the section is made of of the parent section's location followed buy the name of the parent section. In Bamboo this is used to retrieve the appropriate web-items for the menu. (see Web Section Module)
- **weight** - this is used to determine the order of the items on the page.
- **label** - this will be displayed on the screen and can be plain text or a property key.
- **link** - the link is the url the link will point to. It can be absolute or relative to Bamboo's context path.
- **condition** - by implementing the com.atlassian.plugin.web.Condition class you can add rules to determine whether the link will be displayed or not.

Both the link and the id can make use of parameters passed to the page. For example:

```xml
<link>/build/viewBuildFiles.action?buildKey=${buildKey}</link>
```

where `${buildKey}` is the parameter name.

RELATED TOPICS

- Bamboo API
- Bamboo Plugin Developer's Guide
  - Build Complete Action Module
  - Build Processor Module
  - Builder Plugin Module
  - Getting Started
  - Index Reader Module
  - Notification Condition Module
  - Post Build Index Writer Module
  - Pre Build Action Module
  - Report Module
  - Source Repository 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
- Setting up Bamboo Development Environment in IDEA

Bamboo Documentation Home
Web Section Module

This page last changed on May 14, 2007 by rosie@atlassian.com.

Description

The WebSection module is used to provide a section or grouping of Web Item Module.

Currently, Web Sections are used to group Web Items for the Administration Menu, Plan Sub Menu, Results Sub Menu

Sample Module Descriptor Element

```xml
<web-section key= "builds" name= "Builds" location= "system.admin" weight= "100">
  <label key= "websections.system.admin.build"/>
  <icon height= "16" width= "16">
    <link> /images/icons/icon_spanner.gif </link>
  </icon>
</web-section>
```

Notes

The section is only displayed on the Administration Menu but it is required for all locations as Bamboo uses it to place the web items.

Which menu the item gets placed in is determined by the location attribute.

The following table indicates the location attribute required for each menu:

<table>
<thead>
<tr>
<th>Menu</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Administration Menu</td>
<td>system.admin</td>
</tr>
<tr>
<td>Plan Sub Menu</td>
<td>build.subMenu</td>
</tr>
<tr>
<td>Build Results Sub menu</td>
<td>results.subMenu</td>
</tr>
</tbody>
</table>

RELATED TOPICS

- Bamboo API
- Bamboo Plugin Developer's Guide
  - Build Complete Action Module
  - Build Processor Module
  - Builder Plugin Module
  - Getting Started
  - Index Reader Module
  - Notification Condition Module
  - Post Build Index Writer Module
  - Pre Build Action Module
  - Report Module
  - Source Repository 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
- Setting up Bamboo Development Environment in IDEA

Bamboo Documentation Home
XWork Plugin Module

This page last changed on May 14, 2007 by rosie@atlassian.com.

Description

Each XWork module is deployed as a plugin module of type xwork and contains one or 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="viewCloverResult" name="View Clover Result">
  <package name="cloverPlugin" extends="buildView">
    <action name="viewCloverResult" 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>
```

RELATED TOPICS

- Bamboo API
- Bamboo Plugin Developer's Guide
  - Build Complete Action Module
  - Build Processor Module
  - Builder Plugin Module
  - Getting Started
  - Index Reader Module
  - Notification Condition Module
  - Post Build Index Writer Module
  - Pre Build Action Module
  - Report Module
  - Source Repository 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
- Setting up Bamboo Development Environment in IDEA

Bamboo Documentation Home
Bamboo Plugin Tutorial

This page last changed on May 14, 2007 by rosie@atlassian.com.

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

RELATED TOPICS

- Bamboo API
- Bamboo Plugin Developer's Guide
  - Build Complete Action Module
  - Build Processor Module
  - Builder Plugin Module
  - Getting Started
  - Index Reader Module
  - Notification Condition Module
  - Post Build Index Writer Module
  - Pre Build Action Module
  - Report Module
  - Source Repository 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
- Setting up Bamboo Development Environment in IDEA

Bamboo Documentation Home
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>
<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
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.
 */
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).

<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`:

### 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 >](#)

### RELATED TOPICS

- Bamboo API
- Bamboo Plugin Developer's Guide
  - Build Complete Action Module
  - Build Processor Module
  - Builder Plugin Module
  - Getting Started
  - Index Reader Module
  - Notification Condition Module
  - Post Build Index Writer Module
  - Pre Build Action Module
  - Report Module
  - Source Repository 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
- Setting up Bamboo Development Environment in IDEA

[Bamboo Documentation Home](#)
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 ]
[@ui.bambooInfoDisplay titleKey='Pattern Matching Labelling' float=false height='80px']
[@ww.label label='Regex Pattern' ]
[@ww.param name='value']${build.buildDefinition.customConfiguration.get('custom.bamboo.labeller.regex')?if_exists}[/@ww.param]
[/@ww.label]
[@ww.label label='Labels' ]
[@ww.param name='value']${build.buildDefinition.customConfiguration.get('custom.bamboo.labeller.label')?if_exists}[/@ww.param]
[/@ww.label]
[/@ui.bambooInfoDisplay]
[/#if]
```

Here we simply build display the configuration by retrieving your custom properties via the same keys we used in the edit view.
Registering the views in the Plugin Descriptor

We need to register these two Freemarker templates as part of our BuildCompleteAction module. We do this by adding `<resource>` tags with the file path of the templates within the module descriptor definition.

```xml
<buildCompleteAction key="labeller" name="Build Labeller" class="com.atlassian.bamboo.plugins.labeller.BuildLabeller">
  <description>An automatic labelling plugin.</description>
  <resource type="freemarker" name="edit" location="templates/buildCompleteAction/regexLabellerEdit.ftl"/>
  <resource type="freemarker" name="view" location="templates/buildCompleteAction/regexLabellerView.ftl"/>
</buildCompleteAction>
```

Once that's done, we can see the templates in action.

Under the edit configuration page:

And under the view configuration page:

Step 2 - Adding validation

Inserting the templates has allowed us to view and edit custom plan configuration properties. However, we should validate the input we provide for the BuildLabeller, to catch invalid labels or regex patterns.

This is where we use the validate method within our BuildLabeller class, which we have previously left to return null in the first tutorial. Bamboo will run this validate method before trying to save custom configuration properties.

```java
/**
 * This method is used to validate a build configuration for a build plan
 * This is used if the CustomBuildCompleteAction needs to have configuration stored
 * against the build plan.
 * @param buildConfiguration
 * @return
 */
public ErrorCollection validate(BuildConfiguration buildConfiguration) {

  // Check the label values to see if they have any invalid characters
  ErrorCollection errors = new SimpleErrorCollection();
  String labelInput = buildConfiguration.getString("custom.bamboo.labeller.label");
  List labels = LabelParser.split(labelInput);

  for (Iterator iterator = labels.iterator(); iterator.hasNext();)
  {
    String label = (String) iterator.next();
    boolean validLabel = LabelParser.isValidLabelName(label);
    if (!validLabel)
    {
      errors.addError("custom.bamboo.labeller.label", label + " contains invalid characters " + LabelParser.getInvalidCharactersAsString());
    }
  }

  return errors;
}
```
// See if the regex is a valid one by trying to compile it
String regex = buildConfiguration.getString("custom.bamboo.labeller.regex");
try {
    Pattern.compile(regex);
} catch (PatternSyntaxException e) {
    errors.addError("custom.bamboo.labeller.regex", regex + " is not a valid regex pattern.");
}
return errors;

The BuildConfiguration object passed to the validation method is the in-memory version of the build plan configuration. You can get your custom property by simply calling getString on the object, providing the custom property key that you used in the Freemarker templates.

Step 3 - Applying the configuration

At this stage, we can edit, validate, and view our custom configuration for this plugin module. We now need to modify our original run method within the BuildLabeller to read the custom configuration properties.

/**
 * This action will run after a build has completed.
 * The build will be tagged with a specified set of labels if the logs matches the specified regex pattern.
 * @param build
 * @param buildResults
 */
public void run(Build build, BuildResults buildResults) {
    // grab the custom configuration object
    Map customConfiguration = build.getBuildDefinition().getCustomConfiguration();
    if (customConfiguration != null) {
        if (customConfiguration.containsKey("custom.bamboo.labeller.label")) {
            List logs = buildResults.getBuildLog();
            String pattern = (String) customConfiguration.get("custom.bamboo.labeller.regex");
            Pattern regexPattern = Pattern.compile(pattern);
            // Go through the logs
            for (Iterator iterator = logs.iterator(); iterator.hasNext();)
            {
                SimpleLogEntry log = (SimpleLogEntry) iterator.next();
                Matcher matcher = regexPattern.matcher(log.getLog());
                // Use a matcher to see if the logs contained the specified regex
                if (matcher.find()) {
                    String labelsInput = (String) customConfiguration.get("custom.bamboo.labeller.label");
                    // Our configuration also allows for multiple labels.
                    List labels = LabelParser.split(labelsInput);
for (Iterator iterator2 = labels.iterator(); iterator2.hasNext();)
{
    String label = (String) iterator2.next();
    getLabelManager().addLabel(label, buildResults, null);
    break;
}
}
}
}
}

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).

RELATED TOPICS

- Bamboo API
- Bamboo Plugin Developer's Guide
  - Build Complete Action Module
  - Build Processor Module
  - Builder Plugin Module
  - Getting Started
  - Index Reader Module
  - Notification Condition Module
  - Post Build Index Writer Module
  - Pre Build Action Module
  - Report Module
  - Source Repository 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
- Setting up Bamboo Development Environment in IDEA

Bamboo Documentation Home
Setting up Bamboo Development Environment in IDEA

This page last changed on Aug 09, 2007 by asridhar.

To get bamboo started. You need to acquire a Commercial License to gain access to Bamboo Source code. Once you have downloaded Bamboo source code, please follow the instructions below to set-up Bamboo in IDEA -

1. Install maven 2.0.4
2. Install JDK 1.5 (Doesn't work with 1.4 ).
3. In the Bamboo directory run mvn clean install -Dmaven.test.skip=true

Coping with Sun JAVA libraries

Due to licensing restrictions, we are not allowed to re-distribute native SUN libraries through our maven2 public repositories.

If you are developing plugins for Bamboo or building Bamboo from source, you might need javax.mail and javax.transaction:jta:jar for Bamboo to build successfully. Please visit our confluence page on Coping with Sun JAVA libraries for further details.

4. Run mvn idea:clean:idea -DdownloadSources=true
5. This should generate atlassian-bamboo.* files in your source directory. Open the project with Idea
6. Right click on atlassian-bamboo module (on the left hand panel under Projects) and click on module settings. Click on the dependencies tab and add all other modules except atlassian-bamboo-web-server. Thus, making the atlassian-bamboo module to be dependent on all the other modules except atlassian-bamboo-web-server
7. Do the same for the atlassian-bamboo-web-server but make it depended only on the atlassian-bamboo module.
8. Add the application. Mine looks like..

<table>
<thead>
<tr>
<th>Main Class</th>
<th>com.atlassian.bamboo.server.Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM Params</td>
<td>-Dorg.mortbay.xml.XmlParser.NotValidating=true -Dbamboo.home=home1 -server -Xmx512m -XX:MaxPermSize=126m</td>
</tr>
<tr>
<td>Program Params</td>
<td>8085 pathtobamboo/bamboo-web-app/src/main/webapp/</td>
</tr>
<tr>
<td>Working Dir</td>
<td>D:\dev\src\atlassian\bamboo-home</td>
</tr>
<tr>
<td>Class Path of Module</td>
<td>atlassian-bamboo-web-server</td>
</tr>
</tbody>
</table>

That should be it.
Bamboo Installation & Upgrade Guide

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

- **Bamboo Installation Guide**
  - Bamboo EAR-WAR Installation Guide
  - Bamboo Standalone Installation Guide (Linux)
  - Bamboo Standalone Installation Guide (Mac)
  - Bamboo Standalone Installation Guide (Windows)
  - Running the Setup Wizard
- **Bamboo Release Notes**
  - Bamboo 1.0 Release Notes
  - Bamboo 1.1 Release Notes
  - Bamboo 1.2 Release Notes
  - Bamboo pre-1.0 Release Notes
- **Bamboo Upgrade Guide**
Bamboo Installation Guide

Requirements

1. JDK/JRE 1.4+. For best performance, use Sun JDK 1.5 (Java 5) and above.
   🚨 Warning: Bamboo has incompatibilities when not running under Sun JDK. Please make sure you are using Sun JDK to run your Bamboo installation.
2. A database. NOTE: Bamboo ships with a built-in HSQL database, which is fine for evaluation purposes. For production environments it is recommended that you use an enterprise database, as described in 2. Connecting Bamboo to an external database.
3. Only if you are using the Bamboo EAR-WAR distribution: A servlet container that supports Servlet 2.4 specification. Most modern containers should comply to this.

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

<table>
<thead>
<tr>
<th>Standalone Installation Guide — Windows</th>
<th>or</th>
<th>EAR-WAR Installation Guide</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standalone Installation Guide — Linux</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Standalone Installation Guide — Mac</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

RELATED TOPICS

Running the Setup Wizard
Upgrade Guide
Release Notes

Bamboo Documentation Home
The Bamboo EAR/WAR edition 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 Bamboo Standalone edition.

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 and install Bamboo EAR-WAR

1. The Bamboo WAR file is available for download here.
2. Deploy onto your application server. In Tomcat there are two ways you can do this:
   a. Place the WAR file directly into the webapps folder of Tomcat. When Tomcat starts it will perform all the necessary extractions.
   b. Extract the WAR file to your chosen directory in the webapps folder.
      - Windows users must avoid Win XP's built in unzip as it doesn't extract all the files. Use a third-party zip extractor like WinZip.
      - Solaris users will need to use GNU tar to handle the longfilenames.

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 2. Set Bamboo Home

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. specify an environment variable 'BAMBOO_HOME' which specifies the absolute path to your {BAMBOO_HOME} directory.

Step 3. Set Java OPTs

It is recommended that you set the following Java OPTs on your Application Server.

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

In Tomcat you can set the above Java OPTs as follows

- Windows:
  1. Find the setenv.bat file.
  2. Assign the desired properties to the JAVA_OPTS variable:
set JAVA_OPTS=-server -XX:MaxPermSize=256m -Dbamboo.home=/opt/bamboo/bamboohome -Xmx512m -Djava.awt.headless=true $JAVA_OPTS

• Linux-based systems:
  1. Find the setenv.sh file
  2. Assign the desired properties to the JAVA_OPTS variable:

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

Step 4. Restart Server

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

Step 5. Configure Bamboo

See [Running the Setup Wizard](#).

RELATED TOPICS

- [Bamboo Installation Guide](#)
- [Bamboo Documentation Home](#)
To install Bamboo Standalone on Linux,

**Step 1. Download and installing Bamboo Standalone**

Bamboo Standalone for Linux 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.

Alternatively, you can 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 2. Launch Bamboo Standalone on Linux**

There are two ways you can launch Bamboo on Linux:

1. Launch via `bamboo.sh` startup script

   You can start Bamboo with the default `bamboo.sh` file in your installation root directory. The `bamboo.sh` command accepts the following options (e.g. `./bamboo.sh start`):
   - `start` — this starts Bamboo.
   - `stop` — this stops Bamboo.
   - `restart` — this restarts Bamboo
   - `status` — this provides the current status of Bamboo.

2. Launch via Java Service Wrapper

   Alternatively, you can start Bamboo via a Java Service Wrapper, which provides services such as automatic restarting. To do this, you will need to use the `start-bamboo` command available in the `/wrapper` folder of the Bamboo installation. You will need to fire the command with one of the following options (e.g. `./start-bamboo start`):
   - `console` — this starts Bamboo in a console. The logs will scroll to standard out.
   - `start` — this starts Bamboo.
   - `stop` — this stops Bamboo.
   - `restart` — this restarts Bamboo
   - `status` — this provides the current status of Bamboo.
   - `dump` — stops Bamboo abruptly by killing the process

   Once Bamboo has started, you can access it by going to your web browser and entering the address: [http://localhost:8085/](http://localhost:8085/).
Step 3. Configure Bamboo

See Running the Setup Wizard.
To install Bamboo Standalone on Mac OS,

**Step 1. Download and install Bamboo Standalone**

Bamboo Standalone for Mac OS 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:
  
  `~/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
```

Alternatively, you can 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 2. Launch 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. Configure Bamboo**

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

To install Bamboo Standalone on Windows,

**Step 1. Download and install Bamboo Standalone**

Bamboo Standalone for Windows 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:

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

Alternatively, you can 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 2. Launch 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 (note: if you are running Bamboo as a service, please ensure that the Windows user which the Bamboo service is running as has the necessary privileges, and is logged into Windows).

Bamboo comes with the following batch files:

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

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

Step 3. Configure Bamboo

See Running the Setup Wizard.
Running the Setup Wizard

Step 1. Installation Settings

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

Welcome to Atlassian Bamboo!

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

<table>
<thead>
<tr>
<th>Configuration Directory</th>
<th>C:/bamboo-home/xml-data/configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Build Data Directory</td>
<td>C:/bamboo-home/xml-data/builds</td>
</tr>
<tr>
<td>Build Working Directory</td>
<td>C:/bamboo-home/xml-data/build-dir</td>
</tr>
<tr>
<td>License Key</td>
<td></td>
</tr>
</tbody>
</table>

- '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 find it simplest to keep the default settings for the above three directories. For more information please see 7.1 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 can generate your own Bamboo evaluation license from your Atlassian self-service account here. If you have any problems with this please email sales.

Step 2. Database

Here, you will choose what type of database Bamboo will use:
Choose where Bamboo should store its data

- 'Embedded Database' — choose this for quick and easy first-time installation of Bamboo. Note that the embedded HSQL database is suitable for evaluation purposes only. You should later move to an external database before deploying Bamboo in production, as described in 7.4 Moving your Bamboo Data to a different Database.

OR:
- 'External Database' — if you wish to use an external database, please see 2. Connecting Bamboo to an external database.

Step 3. Starting Data

Here, you will tell Bamboo how to populate the 'home directory' that you setup when you installed Bamboo.

- 'Create new Bamboo home' — choose this if you are performing a normal installation or upgrade.

OR:
- 'Import existing data' — only choose this under exceptional circumstances, e.g. if you are connecting Bamboo to a different database, or moving your pre-existing Bamboo installation to a different server.

Step 4. Bamboo Administrator

Here, you will enter the details of the first registered user to the Bamboo system. This user will have global administrative privileges over the entire installation of Bamboo and should not be removed.
Step 5. Server Configuration

The final page of the Setup Wizard allows you to enter some final configuration data for Bamboo.

- 'Name' — for more details please see 8.3 Specifying Bamboo’s Title.
- 'Base URL' — for more details please see 8.4 Specifying Bamboo’s URL.
- 'Apply gzip compression to reduce the size of Bamboo’s web pages?' — for more details please see 8.5 Enabling GZIP Compression.
- 'Accept remote API calls?' — for more details please see 8.6 Enabling Bamboo’s Remote API.

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

Next...

1.1 About Projects, Plans and Builds
1.2 Creating a Plan
1. Using Bamboo's embedded HSQL database

For quick and easy installation, use the embedded HSQL database that ships with Bamboo.

Note that the embedded HSQL database is suitable for evaluation purposes only. You should move to an external database before deploying Bamboo in production. For details please see 7.4 Moving your Bamboo Data to a different Database.

To use Bamboo's embedded HSQL database,

1. At Step 2 of the Setup Wizard, choose 'Embedded Database'.
2. Click the 'Continue' button.
3. Go to Step 3 of the Setup Wizard.

RELATED TOPICS

• 1. Using Bamboo's embedded HSQL database
• 2. Connecting Bamboo to an external database
  ° 2.1 MySQL 4.1 and 5.0
  ° 2.2 Postgres 8+
  ° 2.3 Unsupported databases

Bamboo Documentation Home
2. Connecting Bamboo to an external database

Bamboo can be connected to an external database. For details and instructions please see:

- 2.1 MySQL 4.1 and 5.0
- 2.2 Postgres 8+
- 2.3 Unsupported databases

Before you begin
Please note: if you are already using Bamboo with the embedded HSQL database (or any other database), and you want to keep your data, please see 7.4 Moving your Bamboo Data to a different Database.

RELATED TOPICS

- 1. Using Bamboo's embedded HSQL database
- 2. Connecting Bamboo to an external database
  - 2.1 MySQL 4.1 and 5.0
  - 2.2 Postgres 8+
  - 2.3 Unsupported databases

Bamboo Documentation Home
2.1 MySQL 4.1 and 5.0

Before you begin
Please ensure that your MySQL database server is set to 'utf8' character encoding, and not 'latin1' character encoding. For details please see:

- MySQL 4.1 documentation: Database Character Set
- MySQL 5 documentation: Database Character Set

First, you need to choose how you will connect to the MySQL database. Please follow the instructions for your chosen method:

- Connecting via JBDC
- Connecting via a datasource

JDBC is generally simpler, and is therefore the recommended method.

Connecting via JBDC

To connect Bamboo to a MySQL database, via JDBC,

1. Put the MySQL JDBC driver jar file ([download here](#)) into your application server’s classpath:
   - For the Bamboo Standalone distribution, copy the jar file into the `webapp/WEB-INF/lib` directory.
   - For the Bamboo EAR-WAR distribution, the location will depend on which application server you are using.

2. At Step 2 of the Bamboo Setup Wizard, choose 'External Database' and select 'MySQL' from the list.

3. The 'Select Database Connection' screen will appear. Select 'Direct JDBC connection'.

4. The 'Setup JDBC Connection' screen will appear as shown in the screenshot below.
   - 'Driver Class Name' — Type the following: `com.mysql.jdbc.Driver`
   - 'Database URL' — Type the URL where Bamboo will access your database. For syntax, please see the MySQL documentation.
     
     ![Screenshot 1: 'Setup JDBC Connection (MySQL)'](https://example.com/screenshot1.png)

     Include the `autoReconnect=true` flag, the `useUnicode=true` flag and the `characterEncoding=utf8` flag, e.g.:

     ```
     jdbc:mysql://localhost/bamboo?
     autoReconnect=true&useUnicode=true&characterEncoding=utf8
     ```

     If the `autoReconnect` flag is not set, the MySQL JDBC driver will eventually time out and Bamboo will no longer be able to communicate with the database.

   - 'User Name' — Type the username that Bamboo will use to access your database.
   - 'Password' — Type the password (if required) that Bamboo will use to access your database.

5. Select the 'Overwrite existing data' checkbox if you wish Bamboo to overwrite any tables that already exist in the database.

6. Go to Step 3 of the Setup Wizard.
Connecting via a datasource

To connect Bamboo to a MySQL database, via a datasource,

1. Configure a datasource in your application server (consult your application server documentation for details). For the syntax of the JDBC URL to use, please see the MySQL documentation.

   In the JDBC URL that you configure in your application server, include the **autoReconnect=true** flag, the **useUnicode=true** flag and the **characterEncoding=utf8** flag, e.g.:

   ```
   jdbc:mysql://localhost/bamboo?
   autoReconnect=true&useUnicode=true&characterEncoding=utf8
   ```

   If the autoReconnect flag is not set, the MySQL JDBC driver will eventually time out and Bamboo will no longer be able to communicate with the database.

2. At Step 2 of the Bamboo Setup Wizard, choose 'External Database' and select 'MySQL' from the list.

3. The 'Select Database Connection' screen will appear. Select 'Connect via a datasource (configured in the application server)'.

4. The 'Setup Datasource Connection' screen will appear as shown in the screenshot below. In the 'JNDI name' field, type the JNDI name of your datasource, as configured in your application server.

   If `java:comp/env/jdbc/DataSourceName` doesn't work, try `jdbc/DataSourceName` (and vice versa).

5. Select the 'Overwrite existing data' checkbox if you wish Bamboo to overwrite any tables that already exist in the database.

6. Go to Step 3 of the Setup Wizard.

Screenshot 2: 'Setup Datasource Connection'

 RELATED TOPICS

- **1. Using Bamboo's embedded HSQL database**
- **2. Connecting Bamboo to an external database**
  - 2.1 MySQL 4.1 and 5.0
  - 2.2 Postgres 8+
  - 2.3 Unsupported databases

Bamboo Documentation Home
2.2 Postgres 8+

Before you begin

If connecting to a remote PostgreSQL server (i.e. if your PostgreSQL server is not installed locally on your Bamboo server host system), please ensure that your `data/postgresql.conf` and `data/pg_hba.conf` files are configured to accept remote TCP connections from the Bamboo server's IP address. Refer to the PostgreSQL documentation for the `listen_addresses` value in the `postgresql.conf` file, as well as documentation for the `pg_hba.conf` file, for enabling TCP connections to your PostgreSQL server. (Note that you will need to restart PostgreSQL once any changes to these files have been made.) See:

- PostgreSQL 8.0 documentation
- PostgreSQL 8.1 documentation
- PostgreSQL 8.2 documentation

First, you need to choose how you will connect to the Postgres database. Please follow the instructions for your chosen method:

- Connecting via JBDC
- Connecting via a datasource

JDBC is generally simpler, and is therefore the recommended method.

Connecting via JBDC

To connect Bamboo to a Postgres database, via JDBC,

1. Put the Postgres JDBC driver jar file ([download here](#)) into your application server's classpath:
   - For the Bamboo Standalone distribution, copy the jar file into the `webapp/WEB-INF/lib` directory.
   - For the Bamboo EAR-WAR distribution, the location will depend on which application server you are using.
2. At Step 2 of the Bamboo Setup Wizard, choose 'External Database' and select 'Postgres' from the list.
3. The 'Select Database Connection' screen will appear. Select 'Direct JDBC connection'.
4. The 'Setup JDBC Connection' screen will appear as shown in the screenshot below.
   - 'Driver Class Name' — Type the following: `org.postgresql.Driver`
   - 'Database URL' — Type the URL where Bamboo will access your database. For syntax, please see the [Postgres JDBC driver documentation](#).
   - 'User Name' — Type the username that Bamboo will use to access your database.
   - 'Password' — Type the password (if required) that Bamboo will use to access your database.
5. Select the 'Overwrite existing data' checkbox if you wish Bamboo to overwrite any tables that already exist in the database.
6. Go to Step 3 of the Setup Wizard.

Screenshot 1: 'Setup JDBC Connection (Postgres)'
Connecting via a datasource

To connect Bamboo to a Postgres database, via a datasource,

1. Configure a datasource in your application server (consult your application server documentation for details). For the syntax of the JDBC URL to use, please see the Postgres JDBC driver documentation.
2. At Step 2 of the Bamboo Setup Wizard, choose 'External Database' and select 'Postgres' from the list.
3. The 'Select Database Connection' screen will appear. Select 'Connect via a datasource (configured in the application server)'.
4. The 'Setup Datasource Connection' screen will appear as shown in the screenshot below. In the 'JNDI name' field, type the JNDI name of your datasource, as configured in your application server. If java:comp/env/jdbc/DataSourceName doesn't work, try jdbc/DataSourceName (and vice versa).
5. Select the 'Overwrite existing data' checkbox if you wish Bamboo to overwrite any tables that already exist in the database.
6. Go to Step 3 of the Setup Wizard.

Screenshot 2: 'Setup Datasource Connection'

RELATED TOPICS

- 1. Using Bamboo's embedded HSQL database
- 2. Connecting Bamboo to an external database
  - 2.1 MySQL 4.1 and 5.0
  - 2.2 Postgres 8+
  - 2.3 Unsupported databases

Bamboo Documentation Home
2.3 Unsupported databases

This page last changed on Oct 31, 2007 by rosie@atlassian.com.

First, you need to choose how you will connect to your database. Please follow the instructions for your chosen method:

- Connecting via JDBC
- Connecting via a datasource

JDBC is generally simpler, and is therefore the recommended method.

Connecting via JDBC

To connect Bamboo to an unsupported database, via JDBC,

1. Put the appropriate JDBC driver jar file into your application server’s classpath:
   - For the Bamboo Standalone distribution, copy the jar file into the webapp/WEB-INF/lib directory.
   - For the Bamboo EAR-WAR distribution, the location will depend on which application server you are using.
2. At Step 2 of the Bamboo Setup Wizard, choose 'External Database' and select 'Unsupported Database' from the list.
3. The 'Select Database Connection' screen will appear. Select 'Direct JDBC connection'.
4. The 'Setup JDBC Connection' screen will appear as shown in the screenshot below.
   - 'Driver Class Name' — Type the classname of your JDBC driver (consult your JDBC driver documentation for details).
   - 'Database URL' — Type the URL where Bamboo will access your database (consult your JDBC driver documentation for details).
   - 'User Name' — Type the username that Bamboo will use to access your database.
   - 'Password' — Type the password (if required) that Bamboo will use to access your database.
   - 'Hibernate Dialect' — Type the Hibernate dialect for your particular database:

        **NOTE:** the databases on this list are not supported by Atlassian. Using these databases is not recommended as there is no guarantee that they will operate correctly with Bamboo. Please consider using a supported database instead.

<table>
<thead>
<tr>
<th>Database</th>
<th>Dialect</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2</td>
<td>net.sf.hibernate.dialect.DB2Dialect</td>
</tr>
<tr>
<td>DB2 AS/400</td>
<td>net.sf.hibernate.dialect.DB2400Dialect</td>
</tr>
<tr>
<td>DB2 OS390</td>
<td>net.sf.hibernate.dialect.DB2390Dialect</td>
</tr>
<tr>
<td>Oracle 9/10g</td>
<td>net.sf.hibernate.dialect.Oracle9Dialect</td>
</tr>
<tr>
<td>Oracle (other versions)</td>
<td>net.sf.hibernate.dialect.OracleDialect</td>
</tr>
<tr>
<td>Sybase</td>
<td>net.sf.hibernate.dialect.SybaseDialect</td>
</tr>
<tr>
<td>Sybase Anywhere</td>
<td>net.sf.hibernate.dialect.SybaseAnywhereDialect</td>
</tr>
<tr>
<td>Microsoft SQL Server</td>
<td>net.sf.hibernate.dialect.SQLServerDialect</td>
</tr>
<tr>
<td>SAP DB</td>
<td>net.sf.hibernate.dialect.SAPDBDialect</td>
</tr>
<tr>
<td>Informix</td>
<td>net.sf.hibernate.dialect.InformixDialect</td>
</tr>
<tr>
<td>Ingres</td>
<td>net.sf.hibernate.dialect.IngresDialect</td>
</tr>
<tr>
<td>Progress</td>
<td>net.sf.hibernate.dialect.ProgressDialect</td>
</tr>
</tbody>
</table>
### Micket SQL

<table>
<thead>
<tr>
<th>Database</th>
<th>Dialect</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mckoi SQL</td>
<td>net.sf.hibernate.dialect.MckoiDialect</td>
</tr>
<tr>
<td>Interbase</td>
<td>net.sf.hibernate.dialect.InterbaseDialect</td>
</tr>
<tr>
<td>Pointbase</td>
<td>net.sf.hibernate.dialect.PointbaseDialect</td>
</tr>
<tr>
<td>FrontBase</td>
<td>net.sf.hibernate.dialect.FrontbaseDialect</td>
</tr>
<tr>
<td>Firebird</td>
<td>net.sf.hibernate.dialect.FirebirdDialect</td>
</tr>
</tbody>
</table>

5. Select the 'Overwrite existing data' checkbox if you wish Bamboo to overwrite any tables that already exist in the database.
6. Go to Step 3 of the Setup Wizard.

### Connecting via a datasource

To connect Bamboo to an unsupported database, via a datasource,

1. Configure a datasource in your application server (consult your application server documentation for details). For the syntax of the JDBC URL to use, please see your JDBC driver documentation.
2. At Step 2 of the Bamboo Setup Wizard, choose 'External Database' and select 'Unsupported Database' from the list.
3. The 'Select Database Connection' screen will appear. Select 'Connect via a datasource (configured in the application server)'.
4. The 'Setup Datasource Connection' screen will appear as shown in the screenshot below. In the 'JNDI name' field, type the JNDI name of your datasource, as configured in your application server. 
   - If `java:comp/env/jdbc/DataSourceName` doesn't work, try `jdbc/DataSourceName` (and vice versa).
5. Select the 'Overwrite existing data' checkbox if you wish Bamboo to overwrite any tables that already exist in the database.
6. Go to Step 3 of the Setup Wizard.

### Screenshot 1: 'Setup JDBC Connection (Unsupported Database)'

### Screenshot 2: 'Setup Datasource Connection'
RELATED TOPICS

• 1. Using Bamboo's embedded HSQL database
• 2. Connecting Bamboo to an external database
  ◦ 2.1 MySQL 4.1 and 5.0
  ◦ 2.2 Postgres 8+
  ◦ 2.3 Unsupported databases

Bamboo Documentation Home
Bamboo Release Notes

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

Latest Version

⚠️ Bamboo 1.2.4 has now been released. Read the full Bamboo 1.2.4 Release Notes and Upgrade Guide.

Previous Releases

- **Bamboo pre-1.0 Release Notes**
  - Bamboo 1.0-RC2 Release Notes
    - Bamboo 1.0-RC2 Upgrade Guide
  - Bamboo 1.0-RC1 Release Notes
    - Bamboo 1.0-RC1 Upgrade Guide
  - Bamboo 0.9 Release Notes
    - Bamboo 0.9 Upgrade Guide
  - Bamboo 0.8 Release Notes
    - Bamboo 0.8 Upgrade Guide
  - Bamboo 0.7 Release Notes
    - Bamboo 0.7 Upgrade Guide
  - Bamboo 0.6 Release Notes
    - Bamboo 0.6 Upgrade Guide
  - Bamboo 0.5 Release Notes
    - Bamboo 0.5 Upgrade Guide
    - Bamboo 0.5.1 Release Notes
    - Bamboo 0.5.1 Upgrade Guide
  - Bamboo 0.4 Release Notes
    - Bamboo 0.4 Upgrade Guide
  - Bamboo 0.3 Release Notes
    - Bamboo 0.3 Upgrade Guide
  - Bamboo 0.2 Release Notes
    - Bamboo 0.2 Upgrade Guide
  - Bamboo 0.1 Release Notes
- **Bamboo 1.2 Release Notes**
  - Bamboo 1.2.4 Release Notes
    - Bamboo 1.2.4 Upgrade Guide
  - Bamboo 1.2.3 Release Notes
    - Bamboo 1.2.3 Upgrade Guide
  - Bamboo 1.2.2 Release Notes
    - Bamboo 1.2.2 Upgrade Guide
  - Bamboo 1.2.1 Release Notes
    - Bamboo 1.2.1 Upgrade Guide
  - Bamboo 1.2 Plugin Interface Changes
  - Bamboo 1.2 Upgrade Guide
- **Bamboo 1.1 Release Notes**
  - Bamboo 1.1.2 Release Notes
    - Bamboo 1.1.2 Upgrade Guide
  - Bamboo 1.1.1 Release Notes
    - Bamboo 1.1.1 Upgrade Guide
  - Bamboo 1.1 Upgrade Guide
- **Bamboo 1.0 Release Notes**
  - Bamboo 1.0.5 Release Notes
    - Bamboo 1.0.5 Upgrade Guide
  - Bamboo 1.0.4 Release Notes
    - Bamboo 1.0.4 Upgrade Guide
  - Bamboo 1.0.3 Release Notes
    - Bamboo 1.0.3 Upgrade Guide
  - Bamboo 1.0.2 Release Notes
    - Bamboo 1.0.2 Upgrade Guide
  - Bamboo 1.0.1 Release Notes
- Bamboo 1.0.1 Upgrade Guide
  ° Bamboo 1.0 Upgrade Guide
  ° Bamboo 1.0-Beta Release Notes
    - Bamboo 1.0-Beta Upgrade Guide
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>✅</td>
</tr>
<tr>
<td>BAM-857</td>
<td>Document our external Javascript widgets</td>
<td>✅</td>
<td>✅</td>
</tr>
<tr>
<td>BAM-821</td>
<td></td>
<td>✅</td>
<td>✅</td>
</tr>
<tr>
<td>JIRA Key</td>
<td>Issue Description</td>
<td>Resolution</td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>-----------------------------------------------------------------------------------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td>BAM-879</td>
<td>Unable to export build configuration - no info on how to repair</td>
<td>Resolution</td>
<td></td>
</tr>
<tr>
<td>BAM-878</td>
<td>Where are my nice build result commit message tool tips?</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-876</td>
<td>NumberFormatException for Test</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-873</td>
<td>Move the Clover plugin to opensource as an example of a Bamboo plugin</td>
<td>Resolved</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 Installation</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>BAM-718</td>
<td>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>Resolution</td>
<td>Status</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></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-16</td>
<td>Ability to externally embed full build status</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-807</td>
<td>Bamboo passes bad parameter diff ViewVC</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-770</td>
<td>Bamboo User ID should also be a repository alias</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-690</td>
<td>Improve validation for CVS :ext</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-499</td>
<td>Use 307 response code instead of meta-refresh when hitting path of bamboo</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-433</td>
<td>exe installer home setting issue</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-683</td>
<td>Allow multiple tabs on dashboard page</td>
<td></td>
<td>Resolved</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 Dec 03, 2007 by rosie@atlassian.com.

⚠️ Bamboo 1.2.4 has now been released. Read the full Bamboo 1.2.4 Release Notes and Upgrade Guide.

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></td>
</tr>
<tr>
<td>BAM-652</td>
<td>Checkboxes dont work properly when removing dependant builds</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-644</td>
<td>Script builder fails for windows</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-616</td>
<td>Capture code changes for dependent and scheduled builds</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-566</td>
<td>The back button on create plan wizard clears previously selected values</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-162</td>
<td>Passwords in plaintext</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-714</td>
<td>Support for SVN File Protocol</td>
<td></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></td>
<td></td>
</tr>
<tr>
<td>BAM-685</td>
<td>Commit comments lose line breaks</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-684</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Issue</td>
<td>Description</td>
<td>Resolution</td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------------</td>
<td></td>
</tr>
<tr>
<td>BAM-676</td>
<td>Can't add builder of type Ant</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Unable to re-index due to locked file</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-674</td>
<td>Build fails to start</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-661</td>
<td>Security and Login Improvements</td>
<td></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></td>
<td></td>
</tr>
<tr>
<td>BAM-647</td>
<td>Split webapp WAR module into a JAR and a WAR module</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-637</td>
<td>Reports build table not sorted</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-635</td>
<td>Auto favourite functionality</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-632</td>
<td>Breadcrumbs should have build numbers</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-631</td>
<td>Reports on top ten longest time to fix for tests</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-630</td>
<td>Test summary page still using the old style</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-625</td>
<td>Redirect after a plan is created</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-624</td>
<td>IE caches ajax response for comments and labels</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-622</td>
<td>Last screen of create build broken on Safari</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-609</td>
<td>error displaying build queue admin page</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-608</td>
<td>too many files open error</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-606</td>
<td>Invalid path to clover throws error</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-595</td>
<td>Adding a comment from the Summary page doesn't work in IE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-572</td>
<td>Need option to disable signups</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-570</td>
<td>Testing for mail and IM servers should be more visible</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-569</td>
<td>Auto report grouping for Tests doesn't seem to work</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-563</td>
<td>Validation for report not being selected</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-543</td>
<td>Minor tweaks of the Admin pages</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-542</td>
<td>Plugin Points for Web Fragments</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-490</td>
<td>Ability to run a Bamboo in 'private mode'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-91</td>
<td>Static files are not cached, increases size of downloads</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-668</td>
<td>JIRA tabs shows up regardless</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-659</td>
<td>Edit configuration needs formatting fixes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-612</td>
<td>Label grammar</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-611</td>
<td>Allow two character plan keys</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Issue Key</td>
<td>Description</td>
<td>Status</td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------</td>
<td></td>
</tr>
<tr>
<td>BAM-603</td>
<td>Accessing /api/index.action throws a freemarker error</td>
<td>Resolved</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>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-658</td>
<td>Project names not ordered in dropdown to create project</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-657</td>
<td>Typo and user interface improvement</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-579</td>
<td>Make top right options clearer</td>
<td>Resolved</td>
<td></td>
</tr>
</tbody>
</table>
Bamboo 1.0-Beta Upgrade Guide

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

Upgrading from Bamboo 0.9 to 1.0-Beta

Please follow the Bamboo Upgrade Guide.

Upgrading from Bamboo 0.8 and earlier

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

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

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

New startup procedures for Mac OS X and Linux distributions

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

Usages for bamboo.sh

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

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

Updates and Issues fixed.

<table>
<thead>
<tr>
<th>Key</th>
<th>Summary</th>
<th>Pr</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>BAM-943</td>
<td>Sessions need to closed in the finally block</td>
<td>![ ]</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>BAM-889</td>
<td>Precedence: bulk mail header causing notifications to be blocked</td>
<td>![ ]</td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-975</td>
<td>Edit Configuratlon broken</td>
<td>![ ]</td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-965</td>
<td>Plan Summary does not render in IE 7 when logged in as a user</td>
<td>![ ]</td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-936</td>
<td>Smack Client does not recognize project/plan keys with numbers</td>
<td>![ ]</td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-933</td>
<td></td>
<td>![ ]</td>
<td></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-925</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-917</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-899</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-897</td>
<td>Unable to Edit Build Configuration</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-851</td>
<td>Null pointer exception creating a build plan that uses svn+ssh</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-848</td>
<td>Bambo cannot run on 64-bit linux machines</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-841</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-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>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

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

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

Updates and Issues fixed

<table>
<thead>
<tr>
<th>Key</th>
<th>Summary</th>
<th>Pr</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>BAM-982</td>
<td>Bamboo fails to start under JDK 1.4</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1098</td>
<td>No page associated with this URI</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1026</td>
<td>Links Are Incorrect When Using 'latest' as the build in the URL</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1023</td>
<td>Internal error when deleting plans with dependencies</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1006</td>
<td>Cannot view logs of latest build</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1002</td>
<td>Perforce commands need better logging</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-994</td>
<td>Internal server error when trying to view TestData history</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-989</td>
<td>Duplicate JARs in classpath</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-988</td>
<td>Error when entered License Key</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-987</td>
<td>FreeMarker template error in plan summary</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-983</td>
<td>Freemaker template error when viewing 'latest' builds</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-962</td>
<td>Provide ability to point to a CVS tag instead of HEAD/Branch</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-960</td>
<td>Builds tend to disable themselves far too frequently</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-852</td>
<td>Internal error after &quot;Specify Source Repository&quot;</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-806</td>
<td>Support for SSH private key authentication (possible using jsch)</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-791</td>
<td>svn+ssh support for private key auth</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-691</td>
<td>CVS Client should use the CVS_RSH environment variable if available</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1031</td>
<td></td>
<td></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-980</td>
<td>Clicking on latest build from home screen does not render the Artifacts or JIRA tabs</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Clicking on Tests tab in Build Result Summary renders empty screen</td>
<td>Resolved</td>
<td></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

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

Bamboo 1.2.4 has now been released. Read the full Bamboo 1.2.4 Release Notes and Upgrade Guide.

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>Atlassian JIRA (14 issues)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BAM-1005</td>
<td>Setup fails</td>
<td>!</td>
</tr>
<tr>
<td>BAM-1017</td>
<td>Never Can Retrieve Changelogs</td>
<td>!</td>
</tr>
<tr>
<td>BAM-1063</td>
<td>Clover doesn't use the checkbox to determine if it should run.</td>
<td>!</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>
</tr>
<tr>
<td>BAM-1000</td>
<td>Standalone Bamboo cannot start on Solaris</td>
<td>!</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>
</tr>
<tr>
<td>BAM-986</td>
<td>Emails should be more intelligent</td>
<td>!</td>
</tr>
<tr>
<td>BAM-976</td>
<td>IM message recipients input accumulates square brackets on form load</td>
<td>!</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>
</tr>
<tr>
<td>BAM-884</td>
<td>Redirection error when browsing to Bamboo pages</td>
<td>!</td>
</tr>
<tr>
<td>BAM-882</td>
<td>Add an option to use SVN Externals</td>
<td>!</td>
</tr>
<tr>
<td>BAM-862</td>
<td>Java error on startup</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>
</tr>
<tr>
<td>BAM-710</td>
<td>Internal server error: org.springframework.dao.DataIntegrityViolationException when deleting projects</td>
<td>!</td>
</tr>
</tbody>
</table>
Bamboo 1.0.3 Upgrade Guide

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.
Atlassian is proud to announce the release of Bamboo 1.0.4! Bamboo 1.0.4 is mainly a bug fix release with over 10 issues resolved.

In this release, the focus has been on resolving connectivity issues with Subversion and Perforce.

**Perforce Improvements**

There have been a few changes in Bamboo's Perforce integration:

- Bamboo will now cache the client root rather than polling the repository continuously to obtain it. This reduces the load on the Perforce server considerably. However, if you change the root in the client definition on Perforce, Bamboo will require a restart to pick up the change.
- Bamboo now uses changelist numbers to detect source code changes rather than a timestamp. This will avoid all sorts of problems that occur when the Bamboo server clock and Perforce server clock are out of sync.
- Bamboo now picks up multi line change descriptions from Perforce.
- Bamboo can now generate web urls for perforce files when using Fisheye.

**Updates and Issues fixed**

<table>
<thead>
<tr>
<th>Key</th>
<th>Summary</th>
<th>Pr</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>BAM-1056</td>
<td>Failed to get the build source code: svn: report aborted</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-974</td>
<td>Bamboo penetrated perforce server with repeated requests on plan creation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-750</td>
<td>Perforce changes are not displayed when a manual build is executed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-1113</td>
<td>Perforce modifications not displayed when a manual build is executed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-1096</td>
<td>Change the way bamboo detects changes in perforce</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-1085</td>
<td>Subversion code refresh failing to pick up new revisions</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-1078</td>
<td>BuildChangeDetector continuously polling Perforce repository</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-1028</td>
<td>Bamboo throws exception when it polls Subversion repository</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-979</td>
<td>Different time zone on Perforce server does not work</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-890</td>
<td>SVN triggered update failing.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-747</td>
<td>Perforce repository polling build plan not building</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-871</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Issue</td>
<td>Description</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td>-------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-829</td>
<td>Manual builds still poll the Perforce server</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-823</td>
<td>Only first line of change description is displayed for Perforce changes</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Web Repository URL is not persisted for Perforce repositories</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Resolved

Resolved

Resolved
Bamboo 1.0.4 Upgrade Guide

Upgrading from Bamboo 1.0.3 to 1.0.4

In this version, an upgrade task has been added to update Perforce plans to use the change list number rather than the timestamp when detecting changes. Please ensure that you have connectivity to the Perforce server before you upgrade.

If Bamboo encounters any errors during the upgrade task it will set the Perforce plan’s last change list number to 0. This means that the next time you build that plan there may be some unusual results (e.g. picking up every single change list). Once this build is complete normal behaviour will resume.

It is strongly recommended that you back up your xml-data directory before proceeding. For full instructions please follow the Bamboo Upgrade Guide.

Upgrading from Bamboo 1.0.2 and earlier

In addition to the above, please read the Upgrade Guide for every version you are skipping during the upgrade. The complete list of Upgrade Guides is available here.
Bamboo 1.0.5 Release Notes

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

Bamboo 1.2.4 has now been released. Read the full Bamboo 1.2.4 Release Notes and Upgrade Guide.

Atlassian is proud to announce the release of Bamboo 1.0.5! Bamboo 1.0.5 is mainly a bug fix release related to subversion connectivity issues.

Updates and Issues fixed

<table>
<thead>
<tr>
<th>Key</th>
<th>Summary</th>
<th>Pr</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>BAM-1139</td>
<td>Locked externals in SVN causes infinite building loop</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1137</td>
<td>Authentication always fails for subversion repository</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1108</td>
<td>Removing last build queue blocks use</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1100</td>
<td>Cannot log into Bamboo</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1060</td>
<td>Bamboo source update problem: &quot;Failed to get the build source code&quot;</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1050</td>
<td>Null pointer when relogging in after session has died</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-911</td>
<td>Cannot authenticate with Svn repository</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1125</td>
<td>Project Creation Fails with Self Signed SSL Certificate for SVN</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1118</td>
<td>FishEye link from Perforce project causes exception</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1022</td>
<td>Login link on comment page broken</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-891</td>
<td>Error when logging in to open source project</td>
<td></td>
<td>Resolved</td>
</tr>
</tbody>
</table>
Upgrading from Bamboo 1.0.4 to 1.0.5

It is strongly recommended that you back up your xml-data directory before proceeding. For full instructions please follow the Bamboo Upgrade Guide.

Upgrading from Bamboo 1.0.4 and earlier

In addition to the above, please read the Upgrade Guide for every version you are skipping during the upgrade. The complete list of Upgrade Guides is available here.
Bamboo 1.1 Release Notes

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

⚠️ Bamboo 1.2.4 has now been released. Read the full Bamboo 1.2.4 Release Notes and Upgrade Guide.

The Atlassian Bamboo team is proud to announce the release of Bamboo 1.1! This release contains a whole host of new features targeted to make your build plans even more powerful and flexible.

⚠️ Want to see Bamboo 1.1 in action? Check out our live opensource instance.

Advanced Notifications

In this release, we have extended Bamboo notifications framework to provide more flexibility, allowing you to select the how, who and when of notifications.

Notification Rules

Rather than having static fields for emails/IM recipients, Bamboo now allows you to define your own notifications for your build plans as a set of rules, giving you greater granularity in controlling exactly which recipient gets notified and when.

<table>
<thead>
<tr>
<th>Notification Trigger</th>
<th>Notification Recipients</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Failed Builds And First Successful</td>
<td>Users: text, admin</td>
<td>Edit Remove</td>
</tr>
<tr>
<td>Notify After X Failed Builds</td>
<td>Roles: Watcher</td>
<td>Edit Remove</td>
</tr>
<tr>
<td>All Completed Builds</td>
<td>Roles: Committer</td>
<td>Edit Remove</td>
</tr>
</tbody>
</table>

Notification Triggers

In release 1.1, we introduce notification triggers, defining exactly when you would like a notification to be sent by Bamboo. By default, you can select a notification to be sent on "all builds completion", "after X failed builds" or "failed builds and first successful build". Want more customised triggers? You can now write your own as a notification condition plugin.

Add Build Notification

Notification Trigger: [All Completed Builds]
Roles: [Failed Builds And First Successful, After X Failed Builds, Watcher - Users who have marked this build as failing]

Notification Preferences

Different users prefer to get notified in different ways. Bamboo now lets you control that, via the new user notification preferences.

Set Your Notification Preference

How would you like bamboo to send you notifications:
- Do not send notifications
- Send an instant message
- Send a text email
- Send an email and an instant message

Please ensure that you have set the appropriate address for the notification type.
Dynamic recipients

Only want to receive a notification when you have committed against the build? Want to opt-in to receive notifications on the build plan that you are keeping an eye on? Bamboo 1.1 introduces two new dynamic recipient roles: committers (those users who have committed to the plan triggering the particular build to execute) and watchers (those users who have marked the build plan as their favourite), which allow you to do just that!

Build Metadata

Every build process is different, and each build will have its own information that you may want to keep track of and use on top of the information that Bamboo stores about your build. This is particularly the case if you run custom plugins in your build process.

Pass them to your build

One way to use your build metadata is to pass it along to your builder as a property or target. To do this, you simply specify your variables in your target (or goal) field in your builder configuration. During build execution, the variables will be substituted with the actual values from your build metadata.

Global Variables

Bamboo 1.1 also allows you the option to specify variables globally. When a build begins, the global variables will be populated to the build's metadata. This is a handy option for you to control many plans in one go.

Global Variables

You can use this page to view, add and delete global variables. Global variables are available on every build run in Bamboo.

View your metadata

Use the "Metadata" tab to keep track of all of your build's metadata.
File Trigger Inclusions/Exclusions

In this release, we also introduce the file trigger inclusion/exclusion filter. Instead of listening and picking up all changes from a repository, you can now use regex patterns to define those files which you do (or don’t) want to trigger builds.

More pluggability

In release 1.1, we have added more plugin points to make Bamboo even more extensible than before. On top of the notification condition plugin point, we have also added pre-build action plugins, as well as repository plugins.

- Repository Plugins Not using SVN, CVS, or Perforce? You can now write a plugin to integrate with your very own source control.
- Pre-build Plugins Similar to the post-build action plugin, the pre-build action plugin will allow you to perform any custom task you may wish. The only difference is, of course, that it occurs before the build execution begins.

Improved Maven 2 error log parsing

Bamboo now intelligently parses the Maven 2 error log for possible errors in the build errors log, giving you a better view of what really went wrong in your build summary.

LDAP and external user management support

In release 1.1, we have improved our user management capability to support externally sourced users and groups, including LDAP, and Crowd.

Performance of Dashboard

With this release, we have also made significant performance improvements to the dashboard, which should see its load times reduce dramatically.

Other updates and bug fixes

On top of these features, we have also made a whole host of bug fixes, with over x bugs fixed since release 1.0.5.
<table>
<thead>
<tr>
<th>Key</th>
<th>Summary</th>
<th>Pr</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>BAM-940</td>
<td>Make Repository pluggable</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1191</td>
<td>IM notifications should allow group to be unselected</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1189</td>
<td>Add global variables to be used in plans</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1183</td>
<td>Tool tip in the dashboard for the plans</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1163</td>
<td>Dashboard very slow</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1155</td>
<td>Error cancelling a user preferences dialog</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1134</td>
<td>&quot;operation not permitted&quot; clicking Completed Builds tab</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1115</td>
<td>Artifact URL fails to escape invalid characters</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1090</td>
<td>Allow pluggable pre-build actions</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1088</td>
<td>Hide User Info From Non-Admins</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1081</td>
<td>Triggering a build only on certain commits</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1080</td>
<td>Lock obtain timed out error</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1073</td>
<td>No way to pass proxy information down to Ant</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1059</td>
<td>Implementation of new style of Notifications</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1058</td>
<td>Error viewing Build: Expression failingSinceBuild.buildResultsSummary is undefined</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1053</td>
<td>ANT_HOME is used for ant</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1032</td>
<td>NullPointerException accessing RSS feed</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1029</td>
<td>LDAP integration</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1020</td>
<td>Inconsistent Test Results between builds</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1016</td>
<td>Allow include/exclude patterns for triggering build</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1011</td>
<td>bamboo.home is not a valid environment variable</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1010</td>
<td>Error summary should parse logs in Maven 2 builds</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-996</td>
<td>FreeMarker template error in Completed Build Results screen</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-952</td>
<td>Bamboo uses older m2 version even when maven 2.0.5 builder configured for the plan</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-950</td>
<td>Out of memory error while processing Clover 2.0a4 results</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-942</td>
<td>Upgrade jfreechart version due to concurrency bug</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-938</td>
<td>Editing LDAP users in Bamboo</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-934</td>
<td>OutOfMemory while checking a build result</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-856</td>
<td>Add a simple batch / shell script to run the standalone</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-847</td>
<td>Remote API for manual checkout and build.</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-819</td>
<td>'Important Files and Directories'</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>BAM-787</td>
<td>add screenshots (when Clover data is available)</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-767</td>
<td>We able to add custom project specific Build Telemetry data</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-729</td>
<td>make terminology consistent (&quot;build&quot; vs. &quot;plan&quot;)</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-641</td>
<td>Ability to ignore file patterns for updates</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-633</td>
<td>Add Recent Activity tab for the dashboard</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-574</td>
<td>The interface used to provide a view of artifacts has issues dealing with subdirectories</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-561</td>
<td>Configure dependencies in the opposite direction</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-517</td>
<td>Email Notifications</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-493</td>
<td>Better handling of large number of unit tests</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-146</td>
<td>Build time graph should use minutes in scale</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-57</td>
<td>Improve configuration of notifications</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-1173</td>
<td>Add an option to use a non Ajax dashboard &quot;hideDashboard&quot;</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-1124</td>
<td>Crowd 1.0.6 + Bamboo 1.0.4</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-915</td>
<td>Time values in Report data table should be more readable</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-792</td>
<td>Pass in bamboo parameters such as project and plan key to builders</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-604</td>
<td>API access to the SCM repository's build identifier for a given build</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-681</td>
<td>Checkboxes for build queues</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-680</td>
<td>Picker for e-mail notifications</td>
<td>Resolved</td>
<td></td>
</tr>
</tbody>
</table>
If you want to check out a live Bamboo instance, take a look at our opensource instance.
Bamboo 1.1 Upgrade Guide

Upgrading from Bamboo 1.0.5 to 1.1

It is strongly recommended that you back up your xml-data directory before proceeding. For full instructions please follow the Bamboo Upgrade Guide.

Please note that the upgrade process may take a while to complete.

Upgrading from Bamboo 1.0.5 and earlier

In addition to the above, please read the Upgrade Guide for every version you are skipping during the upgrade. The complete list of Upgrade Guides is available here.
Bamboo 1.1.1 Release Notes

Atlassian is proud to announce the release of Bamboo 1.1.1! Bamboo 1.1.1 is mainly a bug fix release.

Major fixes include:

- LDAP - Many problems with LDAP integration have been overcome
- IMPORT/EXPORT - Several import fixes were implemented
- CVS - CVS change detection has been improved

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-1244</td>
<td>Error after i have upgraded ldap in Bamboo</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1233</td>
<td>A failed build is shown as a successful build</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1226</td>
<td>Building a project within Bamboo with JDK 1.4 sometimes fails when it does build fine outside of Bamboo</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1215</td>
<td>Import fails if the import file is too large</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1206</td>
<td>Unable to remove build plan dependency</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1151</td>
<td>Create plan input lost on timeout</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1857</td>
<td>Document BAM-1224</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1234</td>
<td>Ant build log appears incorrect through Bamboo which causes build to be successful rather than failed</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1227</td>
<td>Bamboo throws freemarker exception when LDAP accounts don't have full names for users</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1224</td>
<td>Improve the REST api in Bamboo to allow getting details from Bamboo at the project level instead of the plan/build level.</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1216</td>
<td>Logging in as an ldap user causes a db exception</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1200</td>
<td>Unable to change how Bamboo sends notifications to user</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1192</td>
<td>&quot;build.dependency.select.none&quot; shown when there are no dependencies</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1154</td>
<td>Changes Made While Build is In Queue Don't Get Change Logs</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1120</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Issue</td>
<td>Description</td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>-----------------------------------------------------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-1117</td>
<td>Arguments field in bash script/script builders are mandatory</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-1066</td>
<td>Ant builder checks last 5 lines for BUILD SUCCESSFUL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-825</td>
<td>Bambo sometimes doesn't know who did CVS commit</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-505</td>
<td>OutOfMemoryException thrown when importing large zip-files</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-1195</td>
<td>Bambo may pickup wrong java installation in sub-processes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-1165</td>
<td>Wording changes for notifications</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-1165</td>
<td>Text on 'Completed Builds' page doesn't match plan build strategy.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Resolved
Bamboo 1.1.1 Upgrade Guide

This page last changed on May 16, 2007 by bmccoy.

Upgrading from Bamboo 1.1 to 1.1.1

Please follow the Bamboo Upgrade Guide

Upgrading from Bamboo 1.0.5 and earlier

In addition to the above, please read the Upgrade Guide for every version you are skipping during the upgrade. The complete list of Upgrade Guides is available here.
Bamboo 1.1.2 Release Notes

Atlassian is proud to announce the release of Bamboo 1.1.2! Bamboo 1.1.2 is mainly a bug fix release.

Major fixes include:

- Export - Windows Export caused some problems, these are now fixed
- Subversion - We have ungraded to the latest SVNKit to incorporate many of their bug fixes
- Fisheye Integration - The Fisheye links for perforce have been fixed
- Character Encoding - Bamboo now lets you use all Unicode characters
- LDAP - More LDAP fixes!

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-1289</td>
<td>Export not working in Windows</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-1256</td>
<td>Upgrade SVNKit to the latest version</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-1254</td>
<td>Upgrade bundled svnkit version to get bugfixes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-1296</td>
<td>PATH variable gets clobbered in JDK 1.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-1279</td>
<td>User and group browser very slow in LDAP environment</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-1262</td>
<td>Browse user and group pages occasionally throws exception (LDAP)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-1260</td>
<td>Cron triggers that specify multiples (entries with commas) get interpreted as arrays of Strings</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-1238</td>
<td>Improve the CVS trigger doco</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-1232</td>
<td>Update docs after packaging the scripts in the WAR version</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-1231</td>
<td>Include the scripts for triggering CVS and SVN builds in the WAR version</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-1229</td>
<td>Dependency builds incorrectly reported as initial or manual build in email notifications</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-1094</td>
<td>start-bamboo wrapper uses false uname syntax</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-1046</td>
<td>Valid cron expression not really valid</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-971</td>
<td>JIRA issue parser should look for numbers in project key as well.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-745</td>
<td>Encoding (of project name)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-1265</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Issue</td>
<td>Description</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td>-------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-1138</td>
<td>When adding builders - adding a label with space in the beginning of fails to build</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-1065</td>
<td>File Version Number problems with Perforce</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-518</td>
<td>Bamboo reports twice as many unit test</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-1235</td>
<td>Unable to copy build artifacts</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-973</td>
<td>Backslashes in usernames not displayed correctly on tooltip</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-973</td>
<td>Users with non standard characters in their names show up corrupted.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Resolved
Bamboo 1.1.2 Upgrade Guide

Upgrading from Bamboo 1.1.1 to 1.1.2

Please follow the Bamboo Upgrade Guide

Upgrading from Bamboo 1.1 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 1.2 Release Notes

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

Bamboo 1.2.4 has now been released. Read the full Bamboo 1.2.4 Release Notes and Upgrade Guide.

Atlassian Software Systems is proud to announce the release of Bamboo 1.2. This release contains:

- Permissions (global and plan-based)
- External database support
- Perforce triggering support
- Scheduled backups
- New Bundled NAnt plugin
- Lots of minor features and bug fixes

Bamboo 1.2 can be downloaded here, and is of course free to all customers who purchased their Bamboo licence or maintenance after July 9, 2006.

When upgrading, please refer to the Bamboo 1.2 Upgrade Guide.

Want to see Bamboo 1.2 in action? Check out our live opensource instance.

Permissions (global and plan-based)

Different organisations, and different projects, have different security requirements. Some information can be made public, while sensitive information may need to be confined to a particular group of people.

Bamboo 1.2 gives you the ability to set security on individual build plans, as well as on your entire Bamboo system:

- **Plan permissions** allow your chosen users to perform a particular operation in relation to a particular build plan (e.g. view its build results).
- **Global permissions** allow your chosen users to perform a particular operation in relation to Bamboo as a whole.

External database support

Bamboo ships with a built-in HSQL database, which is well suited to evaluation purposes. When deploying Bamboo in production, however, you will probably prefer to connect Bamboo to an enterprise database of your choice.

Bamboo 1.2 now includes support for MySQL and Postgres. If you need to use a different database, generic instructions for connecting Bamboo to an unsupported database are also provided.
Perforce triggering support

We are pleased to announce that Bamboo builds can now be triggered by Perforce repositories (previously only Subversion and CVS repositories were supported).

Scheduled backups

You can now schedule your Bamboo data exports to occur automatically at a convenient time:

New Bundled NAnt plugin

Want to build your .Net projects on Bamboo? Now you can, with the NAnt plugin, which comes bundled by default with Bamboo 1.2.

On the topic of plugins, have you checked out Bamboo Extensions space, home to a whole host of cool Bamboo plugins?

Other updates and bug fixes

On top of these features, Bamboo 1.2 also includes a host of minor new features, improvements and bug fixes:
<table>
<thead>
<tr>
<th>Key</th>
<th>Summary</th>
<th>Pr</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>BAM-1380</td>
<td>Exporting from an instance with large ZIP artifacts may fail</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1298</td>
<td>Bamboo doesn't URL encode &quot;&quot;</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1683</td>
<td>Fix for IndexOutOfBoundsException exceptions in the SVNDeltaReader</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1426</td>
<td>Ability to customize the installation name for a Bamboo instance</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1422</td>
<td>Option to Export/Backup without Artifacts</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1417</td>
<td>Ability to remove artifacts for a build result</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1415</td>
<td>Perforce Build Trigger Scripts</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1406</td>
<td>Perforce change logs not picked up due to update returning too many results.</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1399</td>
<td>Log Output Download Option</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1395</td>
<td>New Jabber command for getting more change logs information</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1384</td>
<td>build expiry should be available on a per project basis</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1374</td>
<td>Delete recovery mechanism in Bamboo</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1373</td>
<td>Bamboo throws NullPointerException while deleting build</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1366</td>
<td>NullPointerException after clicking &quot;1. Plan Details&quot; Tab</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1348</td>
<td>Bamboo shows &quot;null build&quot; in RSS header (for project builds).</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1346</td>
<td>Single quotes in logs are prefixed with a back slash \</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1345</td>
<td>Subscribing to a RSS feed from a plan page leads to a 404 page not found exception</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1340</td>
<td>Deadlock issue while view currently running build</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1337</td>
<td>REST API invalid userid error not handled correctly</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1336</td>
<td>REST API documentation not accurate</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1328</td>
<td>Look at Clover XML Even If Build Fails</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1322</td>
<td>Bamboo error when browsing authors</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>Issue Key</td>
<td>Description</td>
<td>Resolution</td>
<td></td>
</tr>
<tr>
<td>-----------</td>
<td>--------------------------------------------------------------------------------------------------------</td>
<td>------------</td>
<td></td>
</tr>
<tr>
<td>BAM-1282</td>
<td>Bamboo goes through password reminder even if no mail server configured</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-1255</td>
<td>Duplicate Email Notifications</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-1253</td>
<td>Allow for expiry of just the artifacts</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-1250</td>
<td>Null Pointer error in GetReturnURL</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-1152</td>
<td>Bamboo should accept repository triggers for Perforce builds</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-1130</td>
<td>access control to specific plan (based on groups)</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-1077</td>
<td>Change terminology/ function of Perforce Source Repository page</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-1042</td>
<td>Modify bamboo.sh to start the JVM in 'server' mode</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-1030</td>
<td>Ability to download particular build log file</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>BAM-957</td>
<td>Ability to configure a database</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-906</td>
<td>Automate backups with a task scheduler</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-859</td>
<td>JUnit XML Improperly Parsed</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-752</td>
<td>Perforce Configuration should allow passwords</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-701</td>
<td>Native support for NAnt builder</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-525</td>
<td>Restrict access to statistics</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-249</td>
<td>Configurable Permissions for Dashboard</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-248</td>
<td>Permissions for User and Groups</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-58</td>
<td>Bootstrapping process</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-1382</td>
<td>Build Labeller plugin fails to validate regex pattern</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-1309</td>
<td>Edit build notification screen has no title</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-1302</td>
<td>Perforce depot access will fail if the depot contains no workspace mapping</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-1259</td>
<td>deleting a build plan caused NPE</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-1068</td>
<td>New mime types for bamboo artifact downloads</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-798</td>
<td>Sub Menu Tabs move from side to side in IE</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-354</td>
<td>customise site title</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-181</td>
<td>General Configuration, Build Expiry --&gt; change to view and edit screens instead of just edit screen</td>
<td>Resolved</td>
<td></td>
</tr>
</tbody>
</table>
Bamboo 1.2 Plugin Interface Changes

Below are details of plugin interface changes with Bamboo 1.2

Notification Condition

The method getTextEmail has changed from

```
public void getTextEmail(Event event, Email email);
```

to

```
public Email getTextEmail(Event event, Email email);
```

It now requires you to return the email object with the content populated (body, subject mimeType etc)
Bamboo 1.2 Upgrade Guide

Upgrading from Bamboo 1.1.2 to 1.2

It is strongly recommended that you back up your xml-data directory before proceeding. For full instructions please follow the Bamboo Upgrade Guide.

- If you are using plugins, please make sure that your plugins are compile against 1.2 before upgrading.

- If you are using Bamboo with Crowd, please make sure that you upgrade to Crowd 1.1.2 before upgrading Bamboo.

Please note that the upgrade process may take a while to complete.

Upgrading from Bamboo 1.1.1 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.

- If you’re upgrading from Bamboo 1.0.x to Bamboo 1.2, please upgrade to 1.1.2 first. There is an issue with the upgrade process from the 1.0.x series that we’re currently looking into
Atlassian is proud to announce the release of Bamboo 1.2.1! Bamboo 1.2.1 is mainly a bug fix release.

It is strongly recommended that you upgrade to Bamboo 1.2.1! It contains a fix to a critical security exploit in the system.

Major fixes include:

- Security exploit in Webwork 2.2.
- JDK 1.4 support
- Import & Export of build plan dependencies
- Upgrading from 1.0.x to 1.2.

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-1477</td>
<td>Webwork 2.2.2 security exploit.</td>
<td>![ ]</td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1475</td>
<td>Plan dependencies not exported or imported (not sure which).</td>
<td>![ ]</td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1471</td>
<td>Bamboo 1.2 JAVA (Unsupported major.minor version 49.0) error.</td>
<td>![ ]</td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1460</td>
<td>Bamboo POM's need to be updated for the next point release</td>
<td>![ ]</td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1455</td>
<td>Slow performance on test results page.</td>
<td>![ ]</td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1453</td>
<td>Importing from a different server from builds pre: 406 causes inconsistent build directories</td>
<td>![ ]</td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1452</td>
<td>Upgrades from 1.0.x directly to 1.2 fails</td>
<td>![ ]</td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1359</td>
<td>Integrate enhanced regex labeller plugin</td>
<td>![ ]</td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-978</td>
<td>Bamboo Plugin Dev Kit pom.xml has a snapshot dependency.</td>
<td>![ ]</td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1470</td>
<td>The Exclude email header &quot;Precedence: bulk&quot; option is not exported and imported.</td>
<td>![ ]</td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-1437</td>
<td>Line breaks are lost in user comments</td>
<td>![ ]</td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-977</td>
<td>Plugin Dev Kit has Mac OSX specific files in it.</td>
<td>![ ]</td>
<td>Resolved</td>
</tr>
</tbody>
</table>
Bamboo 1.2.1 Upgrade Guide

This page last changed on Jul 18, 2007 by edwin@atlassian.com.

Upgrading from Bamboo 1.2 to 1.2.1

Please follow the Bamboo Upgrade Guide

Upgrading from Bamboo 1.1 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.
Atlassian is proud to announce the release of Bamboo 1.2.2!

Major features include:

- **Bulk editing of plan permissions.**
- Administrators can now change users’ passwords.
- Improved caching on the dashboard, for better performance.

Major fixes include:

- Import and export when integrated with LDAP or Crowd.
- More import and export fixes.
- Users can now IM with Crowd integrated.

Updates and Issues fixed

<table>
<thead>
<tr>
<th>T</th>
<th>Key</th>
<th>Summary</th>
<th>Assignee</th>
<th>Reporter</th>
<th>Pr</th>
<th>Status</th>
<th>Res</th>
<th>Created</th>
<th>Updated</th>
<th>Due</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>BAM-1086</td>
<td>z-index of comment hover text is lower than other page elements</td>
<td>Adrian Hempel</td>
<td>Chris Beams [Atlassian]</td>
<td></td>
<td></td>
<td></td>
<td>Mar 30, 2007</td>
<td>Nov 08, 2007</td>
<td></td>
</tr>
<tr>
<td>Issue Key</td>
<td>Summary</td>
<td>Assignee</td>
<td>Resolution</td>
<td>Created</td>
<td>Updated</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----------</td>
<td>--------------------------------------------------------------------------</td>
<td>-------------------</td>
<td>------------</td>
<td>-----------</td>
<td>-----------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-1284</td>
<td>Trying to edit the build properties to include the bamboo.buildVersion parameter</td>
<td>Brydie John McCoy, Reynolds</td>
<td>FIXED</td>
<td>May 24, 2007</td>
<td>Aug 08, 2007</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-1562</td>
<td>&quot;a&quot; duplicated in the Indexing admin page</td>
<td>Brydie James McCoy, Odeen</td>
<td>FIXED</td>
<td>Aug 06, 2007</td>
<td>Aug 07, 2007</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-1566</td>
<td>Bamboo does not export users notification preference</td>
<td>Brydie, Brydie McCoy, McCoy</td>
<td>FIXED</td>
<td>Aug 07, 2007</td>
<td>Aug 07, 2007</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-1550</td>
<td>Exports (Scheduled back-ups) do not work when Crowd is integrated with Bamboo.</td>
<td>Ajay Sridhar, Bryant McCoy</td>
<td>FIXED</td>
<td>Aug 02, 2007</td>
<td>Aug 07, 2007</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-1188</td>
<td>Can't send IM when Bamboo is integrated with Crowd</td>
<td>Mark Chaimungkalanont, Ajay Sridhar</td>
<td>FIXED</td>
<td>May 06, 2007</td>
<td>Aug 06, 2007</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-1529</td>
<td>Bamboo leaves build queue's in a disabled state, if exports (scheduled backups) fail</td>
<td>Adrian Hempel, Ajay Sridhar</td>
<td>FIXED</td>
<td>Jul 31, 2007</td>
<td>Aug 01, 2007</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Issue</td>
<td>Summary</td>
<td>Reporter</td>
<td>Assignee</td>
<td>Resolution</td>
<td>Status</td>
<td>Created</td>
<td>Updated</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>-------------------------------------------------------------------------</td>
<td>-------------------</td>
<td>---------------------------</td>
<td>------------</td>
<td>---------</td>
<td>----------</td>
<td>----------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-1530</td>
<td>Poor documentation for link on the chooseBuildsToMove.action page</td>
<td>Brydie McCoy</td>
<td>Sridhar Ajay</td>
<td>FIXED</td>
<td>Resolved</td>
<td>Jul 31</td>
<td>Aug 01</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-1454</td>
<td>Vague error when exporting while building</td>
<td>Brydie McCoy</td>
<td>Sridhar Ajay</td>
<td>FIXED</td>
<td>Resolved</td>
<td>Jul 11</td>
<td>Aug 01</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-1445</td>
<td>Time Taken* duplication on successful export</td>
<td>Brydie McCoy</td>
<td>James Odeen</td>
<td>FIXED</td>
<td>Resolved</td>
<td>Jul 09</td>
<td>Jul 31</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-1390</td>
<td>JavaScript error while navigation through the tabs in Edit plan screen</td>
<td>Brydie McCoy</td>
<td>Suresh Gopalakrishnan</td>
<td>FIXED</td>
<td>Resolved</td>
<td>Jun 25</td>
<td>Jul 31</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-1043</td>
<td>Artifacts Tab on build configuration tab is misbehaving</td>
<td>Brydie McCoy</td>
<td>Brydie McCoy</td>
<td>FIXED</td>
<td>Resolved</td>
<td>Mar 19</td>
<td>Jul 31</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-1499</td>
<td>The plan level &quot;All builds&quot; rss feed is not sorted in the right chronological order.</td>
<td>Brydie McCoy</td>
<td>Sridhar Ajay</td>
<td>FIXED</td>
<td>Resolved</td>
<td>Jul 24</td>
<td>Jul 31</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-1326</td>
<td>Clover Page On Build Result Is Cutoff in IE6</td>
<td>Brydie McCoy</td>
<td>Sam Berlin</td>
<td>FIXED</td>
<td>Resolved</td>
<td>Jun 07</td>
<td>Jul 30</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-1523</td>
<td>Typo in Build Notification screen</td>
<td>Brydie McCoy</td>
<td>Christopher Owen</td>
<td>FIXED</td>
<td>Resolved</td>
<td>Jul 30</td>
<td>Jul 30</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-1436</td>
<td>It's possible to add duplicate labels for a build result you then can't</td>
<td>Adrian Hempel</td>
<td>Mark Chaimungkalanont</td>
<td>FIXED</td>
<td>Resolved</td>
<td>Jul 05</td>
<td>Jul 27</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
remove
them
Bamboo 1.2.2 Upgrade Guide

This page last changed on Aug 16, 2007 by rosie@atlassian.com.

Upgrading from Bamboo 1.2 (or 1.2.1) to 1.2.2

Please follow the Bamboo Upgrade Guide, plus:

⚠ **Bamboo on Tomcat 5**
If you are running Bamboo on Tomcat 5, please follow the instructions on this page.

Upgrading from Bamboo 1.1 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 1.2.3 Release Notes

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

⚠️ Bamboo 1.2.4 has now been released. Read the full Bamboo 1.2.4 Release Notes and Upgrade Guide.

12 September 2007

Atlassian is proud to announce the release of Bamboo 1.2.3. This point release includes more than 20 minor fixes and improvements. Most notably, for greater flexibility when configuring a build plan, variables can now be used in a number of different places.

Bamboo 1.2.3 can be downloaded here. When upgrading, please refer to the Bamboo 1.2.3 Upgrade Guide.

Updates and issues fixed

<table>
<thead>
<tr>
<th>Key</th>
<th>Summary</th>
<th>Assignee</th>
<th>Reporter</th>
<th>Pr</th>
<th>Status</th>
<th>Res</th>
<th>Created</th>
<th>Updated</th>
<th>Due</th>
</tr>
</thead>
<tbody>
<tr>
<td>BAM-1646</td>
<td>Implement rules to determine whether an artifact is to be downloaded or viewed in the browser</td>
<td>Brydie McCoy</td>
<td>Adrian Hempel [Atlassian]</td>
<td></td>
<td>FIXED</td>
<td>Resolved</td>
<td>Aug 28, 2007</td>
<td>Oct 21, 2007</td>
<td></td>
</tr>
<tr>
<td>BAM-1749</td>
<td>System Environment Variables Separator Not Indicated Upgrade Serpah dependency from 0.7.17 to 0.7.23</td>
<td>Rosie Jameson [Atlassian]</td>
<td>Geoffrey Wiseman</td>
<td></td>
<td>FIXED</td>
<td>Resolved</td>
<td>Sep 28, 2007</td>
<td>Oct 07, 2007</td>
<td></td>
</tr>
</tbody>
</table>

Atlassian JIRA (24 issues)
<table>
<thead>
<tr>
<th>Issue</th>
<th>Summary</th>
<th>Resolution</th>
<th>Assigned To</th>
<th>Created</th>
<th>Updated</th>
</tr>
</thead>
<tbody>
<tr>
<td>BAM-1442</td>
<td>User comments should have the standard link formatting applied to it</td>
<td>Fixed</td>
<td>Brydie McCoy Chaimungkalanont [Atlassian]</td>
<td>Jul 08, 2007</td>
<td>Aug 15, 2007</td>
</tr>
<tr>
<td>BAM-1660</td>
<td>Duplicate Path variable(s) added by Bamboo when executing Build on windows</td>
<td>Fixed</td>
<td>Ajay Sridhar [Atlassian]</td>
<td>Aug 31, 2007</td>
<td>Sep 11, 2007</td>
</tr>
<tr>
<td>BAM-1668</td>
<td>SSO with Crowd causes NoSuchMethodError</td>
<td>Fixed</td>
<td>Unassigned</td>
<td>Sep 07, 2007</td>
<td>Sep 10, 2007</td>
</tr>
<tr>
<td>BAM-1667</td>
<td>Bamboo can not parse test results, where</td>
<td>Fixed</td>
<td>Adrian Hempel Sridhar [Atlassian]</td>
<td>Sep 03, 2007</td>
<td>Sep 03, 2007</td>
</tr>
<tr>
<td>Issue Number</td>
<td>Summary</td>
<td>Assigned To</td>
<td>Resolution</td>
<td>Status</td>
<td>Created</td>
</tr>
<tr>
<td>--------------</td>
<td>---------</td>
<td>-------------</td>
<td>------------</td>
<td>--------</td>
<td>---------</td>
</tr>
<tr>
<td>BAM-1638</td>
<td>Bamboo fails to export when no plans are configured</td>
<td>Ajay Sridhar</td>
<td>FIXED</td>
<td>Resolved</td>
<td>Aug 27, 2007</td>
</tr>
<tr>
<td>BAM-1629</td>
<td>Bamboolog is configured but does not show &lt;user&gt; &lt;url&gt; &lt;starting memory free in bamboo.log</td>
<td>Levent Tutar</td>
<td>FIXED</td>
<td>Resolved</td>
<td>Aug 22, 2007</td>
</tr>
<tr>
<td>BAM-1563</td>
<td>Incorrect format for Build Time when configuring a plan</td>
<td>James Odeen</td>
<td>FIXED</td>
<td>Resolved</td>
<td>Aug 06, 2007</td>
</tr>
<tr>
<td>BAM-1590</td>
<td>Bamboo logs show a harmless LicenseException during Setup Wizard</td>
<td>Adrian Hempel</td>
<td>FIXED</td>
<td>Resolved</td>
<td>Aug 14, 2007</td>
</tr>
<tr>
<td>BAM-1559</td>
<td>Build results list sorted</td>
<td>Mark Chaimungkalanont</td>
<td>FIXED</td>
<td>Resolved</td>
<td>Aug 05, 2007</td>
</tr>
</tbody>
</table>
Bamboo 1.2.3 Upgrade Guide

Upgrading from Bamboo 1.2.x to 1.2.3

Please follow the Bamboo Upgrade Guide, plus:

⚠️ Bamboo on Tomcat 5
If you are running Bamboo on Tomcat 5, please follow the instructions on this page.

Upgrading from Bamboo 1.1.x or earlier

In addition to the above, please read the Bamboo 1.2 Upgrade Guide and the Upgrade Guide for every version you are skipping during the upgrade. The complete list of Upgrade Guides is available here.
Bamboo 1.2.4 Release Notes

This page last changed on Dec 03, 2007 by rosie@atlassian.com.
17 October 2007

Atlassian is proud to announce the release of Bamboo 1.2.4. This point release includes more than 20 minor fixes and improvements. Bamboo 1.2.4 can be downloaded here. When upgrading, please refer to the Bamboo 1.2.4 Upgrade Guide.

Updates and issues fixed

<table>
<thead>
<tr>
<th>Key</th>
<th>Summary</th>
<th>Assignee</th>
<th>Reporter</th>
<th>Pr</th>
<th>Status</th>
<th>Res</th>
<th>Created</th>
<th>Updated</th>
<th>Due</th>
</tr>
</thead>
<tbody>
<tr>
<td>BAM-151</td>
<td>Have a nicer Error page, when trying to see a non existing build report</td>
<td>Brydie McCoy [Atlassian]</td>
<td>Samuel Le [Atlassian]</td>
<td></td>
<td>FIXED</td>
<td>Resolved</td>
<td>Jul 25, 2007</td>
<td>Sep 17, 2007</td>
<td></td>
</tr>
<tr>
<td>Issue</td>
<td>Title</td>
<td>Details</td>
<td>Resolution Date</td>
<td>Reporter</td>
<td>Comment Date</td>
<td>Status</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td>-------</td>
<td>---------</td>
<td>-----------------</td>
<td>----------</td>
<td>--------------</td>
<td>--------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-166</td>
<td>Build expiry does not occur if global configuration is turned off</td>
<td>Brydie McCoy, Mark Chaimungkalanont [Atlassian]</td>
<td>FIXED</td>
<td>Sep 02, 2007</td>
<td>Oct 10, 2007</td>
<td>Resolved</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-179</td>
<td>Build expiry is set to expire from last built date - as opposed to current working date.</td>
<td>Brydie McCoy, Ajay Sridhar [Atlassian]</td>
<td>FIXED</td>
<td>Oct 08, 2007</td>
<td>Oct 10, 2007</td>
<td>Resolved</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-179</td>
<td>Bamboo gives the impression that Build Expiry will run at the end of the build</td>
<td>Unassigned, Brydie McCoy [Atlassian]</td>
<td>FIXED</td>
<td>Oct 07, 2007</td>
<td>Oct 10, 2007</td>
<td>Resolved</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-178</td>
<td>View commits page may</td>
<td>Mark Chaimungkalanont [Atlassian]</td>
<td>FIXED</td>
<td>Oct 05, 2007</td>
<td>Oct 05, 2007</td>
<td>Resolved</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Issue Number</td>
<td>Title</td>
<td>Assigned To</td>
<td>Type</td>
<td>Resolution</td>
<td>Created</td>
<td>Updated</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------------</td>
<td>------------------------------------------------------------------------</td>
<td>---------------------</td>
<td>--------</td>
<td>------------</td>
<td>---------------</td>
<td>--------------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-172</td>
<td>Exception on ViewBuilder due to failingSinceBuild=action.getFailingSinceTestResult</td>
<td>Brydie McCoy Sridhar</td>
<td>FIXED</td>
<td></td>
<td>Sep 24, 2007</td>
<td>Oct 02, 2007</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-169</td>
<td>Bamboo dashboard never updates Plan name if it has been changed</td>
<td>Edwin James Wong Dumay</td>
<td>FIXED</td>
<td></td>
<td>Sep 13, 2007</td>
<td>Oct 02, 2007</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-158</td>
<td>Bamboo throws a null pointer exception, due to session timeout while creating a plan.</td>
<td>Edwin Ajay Wong Sridhar</td>
<td>FIXED</td>
<td></td>
<td>Aug 13, 2007</td>
<td>Sep 25, 2007</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-170</td>
<td>Unable to see the test detail page when the test name contains a slash</td>
<td>Brydie Samuel McCoy Langlois</td>
<td>FIXED</td>
<td></td>
<td>Sep 13, 2007</td>
<td>Sep 25, 2007</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-169</td>
<td>Build artifacts won't open correctly in Firefox</td>
<td>Brydie Arjan McCoy Schaf</td>
<td>FIXED</td>
<td></td>
<td>Sep 12, 2007</td>
<td>Sep 19, 2007</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ID</td>
<td>Description</td>
<td>Assignee</td>
<td>Resolution</td>
<td>Created</td>
<td>Resolved</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-----------------</td>
<td>------------</td>
<td>----------</td>
<td>-----------</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-168</td>
<td>Users without Admin rights can't change number of shown builds in ViewbuildResults page</td>
<td>Edwin Wong, Arjan Schaaf [Atlassian]</td>
<td>FIXED</td>
<td>Sep 10, 2007</td>
<td>Sep 17, 2007</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-169</td>
<td>When editing notification rule, warning message about IM server missing appears even when one configured</td>
<td>Brydie McCoy, Edwin Wong [Atlassian]</td>
<td>FIXED</td>
<td>Sep 12, 2007</td>
<td>Sep 16, 2007</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Bamboo 1.2.4 Upgrade Guide

Upgrading from Bamboo 1.2.x to 1.2.4

Please follow the Bamboo Upgrade Guide, plus:

⚠️ Bamboo on Tomcat 5
If you are running Bamboo on Tomcat 5, please follow the instructions on [this page](#).

Upgrading from Bamboo 1.1.x or earlier

In addition to the above, please read the Bamboo 1.2 Upgrade Guide and the Upgrade Guide for every version you are skipping during the upgrade. The complete list of Upgrade Guides is available [here](#).
Bamboo 0.1 Release Notes

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

⚠️ Bamboo 1.2.4 has now been released. Read the full Bamboo 1.2.4 Release Notes and Upgrade Guide.

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 and 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 Aug 28, 2007 by rosie@atlassian.com.

⚠️ Bamboo 1.2.4 has now been released. Read the full Bamboo 1.2.4 Release Notes and Upgrade Guide.

⚠️ 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
Upgrading from Bamboo 0.1 to 0.2

Please follow the Bamboo Upgrade Guide.
Bamboo 0.3 Release Notes

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

⚠️ Bamboo 1.2.4 has now been released. Read the full Bamboo 1.2.4 Release Notes and Upgrade Guide.

⚠️ Upgrading? Please see the Bamboo 0.3 Upgrade Guide.

New Features for 0.3

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

<table>
<thead>
<tr>
<th>Key</th>
<th>Summary</th>
<th>Pr</th>
<th>Status</th>
<th>Res</th>
</tr>
</thead>
<tbody>
<tr>
<td>BAM-171</td>
<td>View Screens for Mail Server and IM Server</td>
<td></td>
<td>FIXED</td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-136</td>
<td>Can't add builder when there are no builders</td>
<td></td>
<td>FIXED</td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-97</td>
<td>IM Server details should be shareable</td>
<td></td>
<td>FIXED</td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-19</td>
<td>Artifacts should respect MIME types</td>
<td></td>
<td>FIXED</td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-22</td>
<td>Bamboo creating phantom directories</td>
<td></td>
<td>FIXED</td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-150</td>
<td>Installation fails to locate bambo-init.properties file for bamboo home directory</td>
<td></td>
<td>ANSWERED</td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-166</td>
<td>Add new screens and improve existing screen to improve overall information delivery</td>
<td></td>
<td>FIXED</td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-113</td>
<td>UnsupportedClassVersionError deploying Bamboo on BEA WLS 8.1 SP 4</td>
<td></td>
<td>FIXED</td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-96</td>
<td>New build failures count in Jabber message is different to web site</td>
<td></td>
<td>FIXED</td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-93</td>
<td></td>
<td></td>
<td>FIXED</td>
<td></td>
</tr>
<tr>
<td>#</td>
<td>Description</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----</td>
<td>-----------------------------------------------------------------------------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-130</td>
<td>Project dependencies don't work</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-131</td>
<td>Validation for groups and users nonfunctional</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-137</td>
<td>Clicking on a project name for queued project cause 500 error</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-134</td>
<td>Context Path duplicated when submitting add/edit row in builders</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-178</td>
<td>Clicking on a project name for queued project cause 500 error</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-180</td>
<td>Context Path duplicated when submitting add/edit row in builders</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-89</td>
<td>Attempts to email even when email not selected</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-140</td>
<td>Fix up footer to include the date that the build was made as well as the build number</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-70</td>
<td>Delete a Project screen always checks the first project when clicking.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-79</td>
<td>Add decent 404 and 500 pages</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-140</td>
<td>double slash (&quot;&quot;) in link</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-70</td>
<td>Phantom project folders created when creating a new project/build</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Bamboo 0.3 Upgrade Guide

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

Upgrading from Bamboo 0.2 to 0.3

Please follow the Bamboo Upgrade Guide.

Upgrading from Bamboo 0.1

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

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

<table>
<thead>
<tr>
<th>T</th>
<th>Key</th>
<th>Summary</th>
<th>Assignee</th>
<th>Reporter</th>
<th>Pr</th>
<th>Status</th>
<th>Res</th>
<th>Created</th>
<th>Updated</th>
<th>Due</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>BAM-221</td>
<td>View CVS URL is optional - mark such as such</td>
<td>Ben Kuo</td>
<td>Riaz Khanmohamed</td>
<td></td>
<td></td>
<td></td>
<td>Sep 14, 2006</td>
<td>Sep 15, 2006</td>
<td></td>
</tr>
<tr>
<td></td>
<td>BAM-220</td>
<td>Optional field appears to be mandatory on update (cvs password)</td>
<td>Ben Kuo</td>
<td>Riaz Khanmohamed</td>
<td></td>
<td>FIXED</td>
<td>Resolved</td>
<td>Sep 14, 2006</td>
<td>Oct 05, 2006</td>
<td></td>
</tr>
<tr>
<td></td>
<td>BAM-213</td>
<td>Grammer on Log page</td>
<td>Ben Kuo</td>
<td>Riaz Khanmohamed</td>
<td></td>
<td>FIXED</td>
<td>Resolved</td>
<td>Sep 14, 2006</td>
<td>Sep 14, 2006</td>
<td></td>
</tr>
<tr>
<td></td>
<td>BAM-211</td>
<td>Keep refresh option for logs</td>
<td>Ben Kuo</td>
<td>Riaz Khanmohamed</td>
<td></td>
<td>FIXED</td>
<td>Resolved</td>
<td>Sep 14, 2006</td>
<td>Sep 15, 2006</td>
<td></td>
</tr>
<tr>
<td></td>
<td>BAM-210</td>
<td>Log page refreshing</td>
<td>Ben Kuo</td>
<td>Riaz Khanmohamed</td>
<td></td>
<td>FIXED</td>
<td>Resolved</td>
<td>Sep 14, 2006</td>
<td>Sep 15, 2006</td>
<td></td>
</tr>
<tr>
<td></td>
<td>BAM-207</td>
<td>WebUrl link should link to the diff of the file</td>
<td>Mark Chaimungkalanont</td>
<td>MattyJ [Atlassian]</td>
<td></td>
<td>FIXED</td>
<td>Resolved</td>
<td>Sep 14, 2006</td>
<td>Sep 14, 2006</td>
<td></td>
</tr>
<tr>
<td></td>
<td>BAM-196</td>
<td>Adding VisualStudio as builder launches IDE when built</td>
<td>Mark Chaimungkalanont</td>
<td>MattyJ [Atlassian]</td>
<td></td>
<td>FIXED</td>
<td>Resolved</td>
<td>Sep 06, 2006</td>
<td>Sep 06, 2006</td>
<td></td>
</tr>
<tr>
<td></td>
<td>BAM-194</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>FIXED</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Issue Key</td>
<td>Description</td>
<td>Assignee</td>
<td>Reporter</td>
<td>Resolution Status</td>
<td>Resolved</td>
<td>Fixed</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----------</td>
<td>--------------------------------------------------</td>
<td>----------</td>
<td>----------</td>
<td>--------------------</td>
<td>----------</td>
<td>-------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-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>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Implement Lucene for indexing of data

Edwin Wong [Atlassian] Resolved Sep 05, 2006 Sep 19, 2006
Bamboo 0.4 Upgrade Guide

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

Upgrading from Bamboo 0.3 to 0.4

Please follow the Bamboo Upgrade Guide.

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

Upgrading from Bamboo 0.2 and earlier.

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

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

⚠️ Bamboo 1.2.4 has now been released. Read the full Bamboo 1.2.4 Release Notes and Upgrade Guide.

⚠️ Upgrading? Please see the Bamboo 0.5 Upgrade Guide.

New in Release 0.5

Enhanced UI Navigation

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

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

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

Home > Bamboo > results > 968

Relevant information

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

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

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

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

This build is fixed in BAM-1001.

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

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

### Recent Failures

- Average time to fix a failure: 9 hours, 54 minutes
- Average number of builds between fixes: 2.50 builds
- The longest time taken to fix a failure is 6 days, 3 hours, 1 minute, from failure starting in build 1.
- The greatest number of builds taken to fix a failure is 26, from failure starting in build 1.

<table>
<thead>
<tr>
<th>Failed</th>
<th>Fixed</th>
<th>Time to Fix</th>
</tr>
</thead>
<tbody>
<tr>
<td>907 (Updated by dsyoung)</td>
<td>908 (Manual build)</td>
<td>1 builds</td>
</tr>
<tr>
<td>902 (Updated by jshumacher)</td>
<td>903 (Manual build)</td>
<td>1 builds</td>
</tr>
<tr>
<td>Oct 9, 2006 2:05:41 PM</td>
<td>Oct 9, 2006 3:11:02 PM</td>
<td>85 minutes</td>
</tr>
<tr>
<td>892 (Updated by ara)</td>
<td>894 (Manual build)</td>
<td>2 builds</td>
</tr>
</tbody>
</table>

### Revised Project Editing

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

<table>
<thead>
<tr>
<th>Source Repository</th>
<th>Builder Configuration</th>
<th>Build Artifacts</th>
<th>Build Notification</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Repository:</strong></td>
<td>** CVS**</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>CVS Root:</strong></td>
<td>:pserver:<a href="mailto:mehai@cvs.atlassian.com">mehai@cvs.atlassian.com</a>:/cvsroot/atlassian:confluence</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Module:</strong></td>
<td>** confluence**</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Web Repository URL:</strong></td>
<td>** <a href="http://rroschos.sydney.atlassian.com:8181/viewre;confluence/">http://rroschos.sydney.atlassian.com:8181/viewre;confluence/</a> **</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Change Detection:</strong></td>
<td>** Triggered Build Strategy**</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

## Screenshots

---

### Confluence 842: Build Result Summary

**summary | tests | code changes | artifacts | logs**

---

#### Changes

This build was the result of the following change:

- [tdavies](#) fix tests

---

#### When:

Sep 26, 2008 10:49:24 AM - 2 weeks ago

#### Duration:

31 minutes

---

### Tests

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

---

### Fixed Tests

<table>
<thead>
<tr>
<th>Test</th>
<th>Failing Since Build</th>
</tr>
</thead>
<tbody>
<tr>
<td>TestConflicteCachingBananaPersister Retrieve</td>
<td>CONF-835 (20 hours before)</td>
</tr>
<tr>
<td>TestConflicteCachingBananaPersister Retrieve non serializable</td>
<td>CONF-841 (99 minutes before)</td>
</tr>
</tbody>
</table>

---

### Build Errors

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

- **Note:** Some input files use or override a deprecated API.
- **Note:** Reconcile with `deprecation` for details.
- **Note:** Some input files use or override a deprecated API.
- **Note:** Reconcile with `deprecation` for details.
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 Chaimungkalanont</td>
</tr>
<tr>
<td>BAM-257</td>
<td>link error to jira</td>
<td></td>
<td>Resolved</td>
<td>Riaz Khanmohamed</td>
</tr>
<tr>
<td>BAM-256</td>
<td>New instance of bamboo, error upgrading allbuild.xml</td>
<td></td>
<td>Resolved</td>
<td>Riaz Khanmohamed</td>
</tr>
<tr>
<td>BAM-246</td>
<td>Changing context path seems to cause installation to fail</td>
<td></td>
<td>Resolved</td>
<td>Mark Chaimungkalanont</td>
</tr>
<tr>
<td>BAM-245</td>
<td>Build Summary Page improvements</td>
<td></td>
<td>Resolved</td>
<td>Mark Chaimungkalanont</td>
</tr>
<tr>
<td>BAM-244</td>
<td>Bread Crumbs</td>
<td></td>
<td>Resolved</td>
<td>Riaz Khanmohamed</td>
</tr>
<tr>
<td>BAM-242</td>
<td>properties file location incorrect</td>
<td></td>
<td>Resolved</td>
<td>Riaz Khanmohamed</td>
</tr>
<tr>
<td>BAM-235</td>
<td>Ant xml files can't be read</td>
<td></td>
<td>Resolved</td>
<td>Riaz Khanmohamed</td>
</tr>
<tr>
<td>BAM-231</td>
<td>Support period date in incorrect format</td>
<td></td>
<td>Resolved</td>
<td>Riaz Khanmohamed</td>
</tr>
<tr>
<td>BAM-197</td>
<td>Next and Previous build link for Test + TestCase screen</td>
<td></td>
<td>Resolved</td>
<td>Mark Chaimungkalanont</td>
</tr>
<tr>
<td>BAM-192</td>
<td>Create Current Building Page</td>
<td></td>
<td>Resolved</td>
<td>Edwin Wong [Atlassian]</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 Chaimungkalanont</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 [Atlassian]</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 EJBs&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>Issue</td>
<td>Description</td>
<td>Assignee</td>
<td>Status</td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>------------------------------------------</td>
<td>----------------</td>
<td>----------</td>
<td></td>
</tr>
<tr>
<td>BAM-234</td>
<td>Cancelling a build part way is marked as a successful build</td>
<td>Riaz Khanmohamed</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-227</td>
<td>build timer on project screen</td>
<td>Riaz Khanmohamed</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-189</td>
<td>Mail template reporting wrong authors count</td>
<td>Mark Chaimungkalanont [Atlassian] Michael Mekaail</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-135</td>
<td>Option to disable projects</td>
<td></td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-261</td>
<td>cvs branch/head consistency</td>
<td>Riaz Khanmohamed</td>
<td>Resolved</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 Aug 28, 2007 by rosie@atlassian.com.

⚠️ Bamboo 1.2.4 has now been released. Read the full Bamboo 1.2.4 Release Notes and Upgrade Guide.

⚠️ 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></td>
<td>Edwin Wong</td>
</tr>
<tr>
<td>BAM-295</td>
<td>Not compatible with JDK 1.4</td>
<td></td>
<td></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></td>
<td>Gwyn Evans</td>
</tr>
</tbody>
</table>

Atlassian JIRA (3 issues)
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 1.2.4 has now been released. Read the full Bamboo 1.2.4 Release Notes and Upgrade Guide.

Upgrading? Please see the Bamboo 0.6 Upgrade Guide.

New in Release 0.6

JIRA issue integration

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

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

BAM-6: Builds Results

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

<table>
<thead>
<tr>
<th>Build Number</th>
<th>Reason</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>BAM-583</td>
<td>Updated by mohai</td>
<td>2 months ago</td>
</tr>
<tr>
<td>BAM-582</td>
<td>Updated by mohai</td>
<td>2 months ago</td>
</tr>
<tr>
<td>BAM-575</td>
<td>Updated by mohai</td>
<td>2 months ago</td>
</tr>
<tr>
<td>BAM-572</td>
<td>Updated by mohai</td>
<td>2 months ago</td>
</tr>
<tr>
<td>BAM-571</td>
<td>Updated by mohai</td>
<td>2 months ago</td>
</tr>
</tbody>
</table>

Authors

Bamboo has now introduced the concept of authors which allows you to see all the build results across all builds that a user has contributed to (and broken!)

<table>
<thead>
<tr>
<th>Build</th>
<th>Last 10 Builds</th>
<th>Last 10 Broken</th>
<th>Last 10 Fixed</th>
</tr>
</thead>
<tbody>
<tr>
<td>JIRA-1087</td>
<td>21 minutes ago</td>
<td>Change 'kicks' to be table again, as this fixes functional tests.</td>
<td></td>
</tr>
<tr>
<td>JIRA-1059</td>
<td>5 hours ago</td>
<td>Also javascript to load the issue operations in a div, rather than a whole test.</td>
<td></td>
</tr>
<tr>
<td>JIRA-1061</td>
<td>1 week ago</td>
<td>Initial javascript for dynamic updating of the edit issue screens in JIRA.</td>
<td></td>
</tr>
<tr>
<td>JIRA-1056</td>
<td>1 week ago</td>
<td>Put the body content in a div.</td>
<td></td>
</tr>
<tr>
<td>JIRA-1013</td>
<td>1 week ago</td>
<td>Produce some slightly near HTML.</td>
<td></td>
</tr>
<tr>
<td>JIRA-1012</td>
<td>1 week ago</td>
<td>Remove 1 link of javascript from every page by making the quicksearch.</td>
<td></td>
</tr>
<tr>
<td>JIRA-1000</td>
<td>1 week ago</td>
<td>A very basic database version of the comment RSS feed. Committing to issues.</td>
<td></td>
</tr>
<tr>
<td>JIRA-355</td>
<td>1 week ago</td>
<td>Remove the need for an issue key which was just used for logging again.</td>
<td></td>
</tr>
<tr>
<td>JIRA-362</td>
<td>1 week ago</td>
<td>Use helper method, rather than duplicating code.</td>
<td></td>
</tr>
</tbody>
</table>

Look at the trends of broken vs fixed builds for each author!
And even and compare authors to see who has broken the most builds.

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

**Improved Fisheye links**

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

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

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

**Other updates & Bug fixes**

<table>
<thead>
<tr>
<th>Key</th>
<th>Summary</th>
<th>Pr</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>BAM-346</td>
<td>Catastrophic failure trying to view test results for a particular build on Panda</td>
<td><img src="image1.png" alt="Image" /></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-486</td>
<td>viewing build config for ant-based builds</td>
<td><img src="image2.png" alt="Image" /></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-394</td>
<td>FishEye file linking</td>
<td><img src="image3.png" alt="Image" /></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-383</td>
<td>Error when clearing build dependency list</td>
<td><img src="image4.png" alt="Image" /></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-377</td>
<td>error during page refresh</td>
<td><img src="image5.png" alt="Image" /></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 menu bar 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</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-302</td>
<td>Quick-search to non-existent builds throws error</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-266</td>
<td>Spacing error on Configuration -&gt; Build Notifications listing</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-190</td>
<td>Create Author Pages</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-95</td>
<td>Build hangs if exception is thrown during build</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-38</td>
<td>Add Bash as a default custom command builder</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-395</td>
<td>Encryption exceptions after a mail send failure</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-375</td>
<td>Use NumberFormat &quot;#&quot; for build numbers in instant messages</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-300</td>
<td>Log error catching</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-292</td>
<td>Disabled build should also hide the build option</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-208</td>
<td>Build table doesn't always refresh correctly</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-314</td>
<td>Double slash (&quot;/&quot;) in link</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-290</td>
<td>Bamboo checklist page doesn't give example of bamboo_home property</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-271</td>
<td>Error page should link back to the home page</td>
<td>Resolved</td>
<td></td>
</tr>
</tbody>
</table>
Bamboo 0.6 Upgrade Guide

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

Upgrading from Bamboo 0.5.1 to 0.6

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 0.7 Release Notes**

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

⚠️ Bamboo 1.2.4 has now been released. Read the full Bamboo 1.2.4 Release Notes and Upgrade Guide.

⚠️ Upgrading? Please see the Bamboo 0.7 Upgrade Guide.

**New in Release 0.7**

**Cross Build Reports**

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

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

**Percentage of Successful Builds**

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

**Author to User Profile Linking**

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

Linking your profile to an alias will enable you to view your builds (including most recent, broken, and fixed builds) in your profile page.
In addition, Bamboo now retrieves details from your user profile wherever your repository alias is referenced.

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

Build Grouping at Project Level

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

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>![ ]</td>
</tr>
<tr>
<td>BAM-465</td>
<td>Dashboard ordered by project key rather than name</td>
<td>![ ]</td>
<td>![ ]</td>
</tr>
<tr>
<td>BAM-435</td>
<td>Write an upgrade documentation</td>
<td>![ ]</td>
<td>![ ]</td>
</tr>
<tr>
<td>BAM-386</td>
<td>Automate deploying for all Maven II Projects once build is successful</td>
<td>![ ]</td>
<td>![ ]</td>
</tr>
<tr>
<td>BAM-7</td>
<td>Use project IDs instead of names in all URLs</td>
<td>![ ]</td>
<td>![ ]</td>
</tr>
<tr>
<td>BAM-469</td>
<td>Build Reports</td>
<td>![ ]</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>![ ]</td>
<td>![ ]</td>
</tr>
<tr>
<td>BAM-454</td>
<td>Improve page titles</td>
<td>![ ]</td>
<td>![ ]</td>
</tr>
<tr>
<td>BAM-453</td>
<td>Use profile names on View Authors page</td>
<td>![ ]</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 artifacts and build results</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-324</td>
<td>Add upgrade task to move Build Definitions</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-320</td>
<td>Build Definitions into Hibernate</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-318</td>
<td>Add tests graph to Tests page of build</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-274</td>
<td>Display related jiras in a build</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-224</td>
<td>Validation not working for edit configuration of build definitions</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-191</td>
<td>Create Commit Page</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-188</td>
<td>Add validation when adding artifacts</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-158</td>
<td>Distributions have confusing directory structure</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-141</td>
<td>Introduce Project -&gt; Build Definition concept</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-92</td>
<td>Project added to build queue more than once</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>Ticket</td>
<td>Description</td>
<td>Status</td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------</td>
<td></td>
</tr>
<tr>
<td>BAM-18</td>
<td>Changing Trigger type forces a new build build key case issue</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

Bamboo 1.2.4 has now been released. Read the full Bamboo 1.2.4 Release Notes and Upgrade Guide.

Upgrading? Please see the Bamboo 0.8 Upgrade Guide.

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

Brand spanking new UI

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

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

Labels Labels Labels

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

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

View the labels: Alphabetically | By Popularity

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

Build Result BAM-MAIN-1493

Clover Reports Plugin

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

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

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

<table>
<thead>
<tr>
<th>Status</th>
<th>Project - Plan</th>
<th>Latest Completed Build</th>
<th>Last Run</th>
<th>Reason</th>
<th>Duration</th>
<th>Test Count</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>✔</td>
<td>Bamboo - HEAD</td>
<td>BAM-MAIN-1465</td>
<td>1 hour ago</td>
<td>Updated by Ben Hoo</td>
<td>5 minutes</td>
<td>1245 passed</td>
<td><img src="image" alt="View Logs" /></td>
</tr>
<tr>
<td>✔</td>
<td>Java - MainDB</td>
<td>JAVA-MAIN-1231</td>
<td>1 hour ago</td>
<td>Updated by [Name]</td>
<td>4 minutes</td>
<td>204 passed</td>
<td><img src="image" alt="View Logs" /></td>
</tr>
</tbody>
</table>

Sloooowest running tests

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

<table>
<thead>
<tr>
<th>Test</th>
<th>Average Duration (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detects updates to a non repository</td>
<td>21.25 (21 seconds)</td>
</tr>
<tr>
<td>Not deletes all source directories when getting all source code</td>
<td>1.72 (1.72 seconds)</td>
</tr>
<tr>
<td>Ensures the web repository url is a valid url</td>
<td>1.47 (1.47 seconds)</td>
</tr>
<tr>
<td>Successful build with single target</td>
<td>4.53 (4.53 seconds)</td>
</tr>
</tbody>
</table>

Pretty URLs

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

Projects, Plans, Builds

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

We’ve also made a myriad of general improvements:

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

Legendary Pluggability

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

We haven’t had time to document them all yet, but they include arbitrary build actions, post build actions, custom builders and custom reports. Stay tuned!
Bamboo Plugin for JIRA 3.7

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

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 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>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-512</td>
<td>Internal Server Error on editing build for Confluence (PostgreSQL, Tomcat) project</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-510</td>
<td>UI Improvements</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-503</td>
<td>&quot;Cannot find or execute 'null' at 'null'&quot; while trying to create a new build</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-502</td>
<td>Additional Build Strategies</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-494</td>
<td>Wrong title on 'Test Running Duration' graph</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-487</td>
<td>Server error creating build</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-482</td>
<td>Allow moving of builds between projects</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>BAM-481</td>
<td>Checking out non-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 <code>&lt;pre&gt;</code> &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>webpage 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>Issue</td>
<td>Description</td>
<td>Resolution</td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>--------------------------------------------------</td>
<td>------------</td>
<td></td>
</tr>
<tr>
<td>BAM-371</td>
<td>Poll at specific times</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-143</td>
<td>Update Quartz usage to utilise Spring</td>
<td>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 Release Notes

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

⚠️ Bamboo 1.2.4 has now been released. Read the full Bamboo 1.2.4 Release Notes and Upgrade Guide.

⚠️ Upgrading? Please see the Bamboo 0.9 Upgrade Guide.

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

### Comments

#### Comments

<table>
<thead>
<tr>
<th>Status</th>
<th>Project - Plan</th>
<th>Latest Completed Build</th>
<th>Last Run</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>✔️</td>
<td>Bamboo - HEAD</td>
<td>BAM-MAIN-1585</td>
<td>2 days ago</td>
<td>Updated by Ben Kuo</td>
</tr>
<tr>
<td>✔️</td>
<td>Confluence - Main Build</td>
<td>CONF-MAIN-1234</td>
<td>2 days ago</td>
<td>Updated by Christopher Owen</td>
</tr>
<tr>
<td>✔️</td>
<td>JIRA - Main Build</td>
<td>JIRA-MAIN-1277</td>
<td>2 days ago</td>
<td>Updated by Ben Kuo</td>
</tr>
</tbody>
</table>

Successful builds are green, failed builds are red.

For all builds or all failed builds.

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

Build Result BAM-MAIN-1585

#### Comments

- Edwin Wong (Dec 21, 2006 7:19:18 PM)
  
  I'm glad you like them, I'm the one that implemented comments =)

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

### Maintenance

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

### Other Updates and Bug Fixes

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

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-RC1 Release Notes

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

Bamboo 1.2.4 has now been released. Read the full Bamboo 1.2.4 Release Notes and Upgrade Guide.

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

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

New in Release 1.0 - RC1

New Bamboo Dashboard

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

**Atlassian Bamboo**

![Dashboard interface with tabs]

"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.

<table>
<thead>
<tr>
<th>My Favourite Builds</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Confluence</strong> : 1 out of 1 build plans failing.</td>
</tr>
<tr>
<td><strong>Test</strong></td>
</tr>
<tr>
<td><strong>Blah &gt; TST-BLAH-61</strong></td>
</tr>
<tr>
<td>Ran: 1 hour ago</td>
</tr>
<tr>
<td><strong>Coverage Test &gt; TST-COVER-184</strong></td>
</tr>
<tr>
<td>Ran: 4 hours ago</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>My Broken Builds</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Test</strong></td>
</tr>
<tr>
<td><strong>Blah &gt; TST-BLAH-61</strong></td>
</tr>
<tr>
<td>Ran: 1 hour ago</td>
</tr>
<tr>
<td><strong>Coverage Test &gt; TST-COVER-184</strong></td>
</tr>
<tr>
<td>Ran: 4 hours ago</td>
</tr>
</tbody>
</table>
Keep check of all your recent commits with "My Changes" portlet.

### My Latest Changes

<table>
<thead>
<tr>
<th>Build</th>
<th>When</th>
<th>Comments</th>
<th>Test Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONF-HEAD-22</td>
<td>4 hours ago</td>
<td>Brydie Just Committed this - Fake Jira Number (BAM-6630)</td>
<td>2 passed</td>
</tr>
<tr>
<td>DD-BB-2</td>
<td>4 hours ago</td>
<td>Brydie Just Committed this - Fake Jira Number (BAM-6630)</td>
<td>2 passed</td>
</tr>
<tr>
<td>TST-DEF-27</td>
<td>4 hours ago</td>
<td>Brydie Just Committed this - Fake Jira Number (BAM-6630)</td>
<td>2 passed</td>
</tr>
</tbody>
</table>

"All Plans"

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

"Current Activity"

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

### Recently Completed Builds

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

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

Dynamic Updates

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

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.

<table>
<thead>
<tr>
<th>Status</th>
<th>Project Plan</th>
<th>Latest Completed Build</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Bamboo Plan Summary

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

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

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

Talk to Bamboo via IM

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

Usage:

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

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

<table>
<thead>
<tr>
<th>Key</th>
<th>Summary</th>
<th>Pr</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>BAM-784</td>
<td>Improve build list UI</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>BAM-780</td>
<td>Freemaker template error on AtlassianUser project on Keg</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-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></td>
</tr>
<tr>
<td>BAM-626</td>
<td>Ajaxify Dashboard</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-618</td>
<td>AJAXify favourites</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-613</td>
<td>Display project, plan hierarchy on dashboard</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-390</td>
<td>build queue/log out of sync with dashboard status</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-20</td>
<td>Build Queue sometimes doesn't match actual project building</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-744</td>
<td>Bamboo does not work properly with Opera 9.10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-728</td>
<td>change &quot;build form&quot; to &quot;build plan&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-689</td>
<td>Rogue JNDI mail settings label</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-564</td>
<td>Live activity logs can sometimes error out</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM-789</td>
<td>Clumsy wording in dependency build cause</td>
<td></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 Dec 03, 2007 by rosie@atlassian.com.

⚠️ Bamboo 1.2.4 has now been released. Read the full Bamboo 1.2.4 Release Notes and Upgrade Guide.

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

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

New in Release 1.0 - RC2

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

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

Bamboo Installer

Bamboo Release 1.0 - RC2 has had its standalone installer revamped. It is now encapsulated inside a Java Service Wrapper to allow you to install and run the standalone Bamboo as a service.

The benefits of running Bamboo as a service include:
1. Bamboo happily runs in the background (no console window required)
2. Bamboo server does not have to be manually started every time you want to use it.

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

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

Other updates and bug fixes

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

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

Upgrading from Bamboo 1.0-RC1 to 1.0-RC2

Please follow the Bamboo Upgrade Guide

Upgrading from Bamboo 1.0-Beta and earlier

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

Before you begin
Please read the Release Notes and Upgrade Guides for the version you are upgrading to.

Step 1. Identify your Bamboo directories

Go to the ‘System Info’ page in the 'Administration' menu of your Bamboo instance and note the location of the Bamboo Home, Build Data Path and Configuration Path directories:

Step 2. Shut down Bamboo

You need to shut down Bamboo before backing up or performing the upgrade.

Step 3. Back up Bamboo

Back up the three directories you identified in Step 1 — your Bamboo Home, Build Data Path and Configuration Path directories (for more information about these directories please see Important Directories and Files).

Note that it is not necessary to back up your Working Directory (located inside your Bamboo Home directory).

Step 4. Re-install Bamboo

Please note:

• When specifying the {BAMBOO_HOME} directory, use the same Bamboo Home directory as in your old installation. That is, specify the same directory and path as the Bamboo Home directory shown in ‘Step 1’ (above).
• Make sure that your {BAMBOO_INSTALL} directory is either a new directory, or else delete your old {BAMBOO_INSTALL} directory before you begin, as legacy files may cause problems.
• The {BAMBOO_HOME} directory must be different from the {BAMBOO_INSTALL} directory. This will ensure that your data is not lost when upgrading or re-installing Bamboo.

Follow steps 1 and 2 of the installation instructions for your operating system:

• Bamboo Standalone Installation Guide — Windows
• Bamboo Standalone Installation Guide — Linux
• Bamboo Standalone Installation Guide — Mac
• Bamboo EAR-WAR Installation Guide

Step 5. Start Bamboo

Once you have installed Bamboo and set the ‘bamboo.home’ property (as described in the Installation Guides), start Bamboo. The upgrade process will be performed when Bamboo starts up. You will not see the Setup Wizard.

Monitor the atlassian-bamboo.log to ensure that the upgrade process has completed successfully.
Step 6. Re-index Bamboo (if indicated in release notes)

Bamboo maintains an index of its build results. This allows 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). Also note that you only need to do this if you have existing data in Bamboo.

To re-index, go to 'Administration', then 'Indexing', and click the 'Reindex' button.

⚠️ 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 in progress when you start the re-indexing process. If you have a large instance, it is recommended that you reindex overnight.

Step 7. Update plugins

If you are using any plugins other than the ones that ship with Bamboo, check that each one is compatible with the new version of Bamboo. Upgrade any plugins that are out-of-date, and disable any plugins that are incompatible with your new version of Bamboo.

Step 8. Re-configure external user repositories (if applicable)

- LDAP integration — If you had previously integrated Bamboo with LDAP/AD, copy your old `<Bamboo-install>/webapps/WEB-INF/atlassian-user.xml` file to the new Bamboo installation.
- Crowd integration — If you had previously integrated Bamboo with Crowd, you will need to re-enable Crowd integration. For details please see integrating Crowd with Bamboo.

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 Knowledge Base

Answers to commonly raised questions about configuring and using Bamboo:

- **What is continuous integration?**
- **Installation FAQ**
  - Adding MIME types to Bamboo Standalone
  - Bamboo 1.2.2 on Tomcat 5
  - Can I have Bamboo.home pointing to a resource on a network share?
  - Changing Bamboo's port from the default 8085
  - Changing the Root Context Path
  - Configuring Bamboo on start-up
  - Getting Bamboo Standalone to use the jetty.xml file
  - Hardware sizing considerations
  - Installation notes for Bamboo on JBoss 4.x
  - Running Bamboo over HTTPS
  - Setting up JNDI on Jetty
  - Supported databases
- **Usage FAQ**
  - Backing up Bamboo instances over 4GB
  - Binding Bamboo to one IP address
  - Can Bamboo build and test non-Java projects?
  - 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.
  - Connecting to Subversion repositories
  - Deactivating a Bamboo user
  - Fixing OutOfMemory Errors in Bamboo
  - Problems running Bamboo under Sun JDK 1.4
  - Restoring passwords to recover admin users
  - Testing LDAP or Active Directory connectivity with Paddle
  - Troubleshooting an SVN connection in Bamboo
  - Using Bamboo with Clover
  - Working with Sun JAVA libraries

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

You may also like to check out the forums:

- Bamboo Announcements
- Bamboo General Forum
- Bamboo Developers Forum
Installation FAQ

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

- Adding MIME types to Bamboo Standalone
- Bamboo 1.2.2 on Tomcat 5
- Can I have Bamboo.home pointing to a resource on a network share?
- Changing Bamboo's port from the default 8085
- Changing the Root Context Path
- Configuring Bamboo on start-up
- Getting Bamboo Standalone to use the jetty.xml file
- Hardware sizing considerations
- Installation notes for Bamboo on JBoss 4.x
- Running Bamboo over HTTPS
- Setting up JNDI on Jetty
- Supported databases
Adding MIME types to Bamboo Standalone

Bamboo Standalone ships with the Jetty application server (see Bamboo 'distributions'). To add additional MIME content types, edit the `mimetypes.xml` file in `../<Bamboo-install>/webapps/WEB-INF/mimetypes.xml` and insert your new mime-mapping.

For instance to add an additional text/plain mime type with txt2 extension, insert the following:

```xml
<mime-mapping>
  <extension>txt2</extension>
  <mime-type>text/plain</mime-type>
</mime-mapping>
```

Restart Bamboo, for the changes to take effect.
If you are upgrading to Bamboo 1.2.2 on Tomcat 5, the endorsed XML libraries (Specifically, 'xml-apis.jar' and 'xercesImpl.jar') in "./<Tomcat_install>/common/endorsed/" get loaded before Bamboo starts-up, which results in the exception below, as Bamboo uses the wrong XML libraries.

javax.xml.transform.TransformerFactoryConfigurationError: Provider
  org.apache.xalan.processor.TransformerFactoryImpl not found
  at javax.xml.transform.TransformerFactory.newInstance(Unknown Source)
  at com.opensymphony.xwork.util.DomHelper$DOMBuilder.<clinit>(DomHelper.java:168)
  at com.opensymphony.xwork.util.DomHelper.parse(DomHelper.java:116)
  at com.opensymphony.xwork.config.providers.XmlConfigurationProvider.loadConfigurationFile(XmlConfigurationProvider.java:641)
  at com.opensymphony.xwork.config.providers.XmlConfigurationProvider.init(XmlConfigurationProvider.java:91)
  at com.opensymphony.xwork.config.ConfigurationManager.getConfiguration(ConfigurationManager.java:55)
  at com.opensymphony.xwork.DefaultActionProxy.<init>(DefaultActionProxy.java:60)
  at com.opensymphony.webwork.dispatcher.DispatcherUtils.serviceAction(DispatcherUtils.java:216)
  at com.opensymphony.webwork.dispatcher.FilterDispatcher.doFilter(FilterDispatcher.java:202)
  at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:186)
  at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:157)
  at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:186)
  at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:157)
  at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:186)
  at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:157)
  at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:186)
  at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:157)
  at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:186)
  at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:157)
  at com.atlassian.acegi.proxy.filter.BambooAcegiProxyFilter.doFilter(BambooAcegiProxyFilter.java:21)
  at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:186)
  at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:157)
  at com.atlassian.acegi.proxy.filter.SecurityFilter.doFilter(SecurityFilter.java:83)
  at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:186)
  at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:157)
  at com.atlassian.spring.filter.FlushingSpringSessionInViewFilter.doFilterInternal(FlushSpringSessionInViewFilter.java:26)
  at org.springframework.web.filter.OncePerRequestFilter.doFilter(OncePerRequestFilter.java:77)
2007-08-14 12:10:30 org.apache.catalina.core.StandardHostValve@e7e8eb: Exception Processing ErrorPage[errorCode=500, location=/500.action]
javax.servlet.ServletException
at com.opensymphony.webwork.dispatcher.DispatcherUtils.serviceAction(DispatcherUtils.java:236)
at com.opensymphony.webwork.dispatcher.ServletDispatcher.service(ServletDispatcher.java:111)
at javax.servlet.http.HttpServlet.service(HttpServlet.java:802)
at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:237)
at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:157)
at org.apache.catalina.core.ApplicationDispatcher.invoke(ApplicationDispatcher.java:704)
at org.apache.catalina.core.ApplicationDispatcher.doForward(ApplicationDispatcher.java:409)
at org.apache.catalina.core.StandardHostValve.custom(StandardHostValve.java:396)
at org.apache.catalina.core.StandardHostValve.throwable(StandardHostValve.java:301)
at org.apache.catalina.core.StandardHostValve.invoke(StandardHostValve.java:244)
at org.apache.catalina.core.StandardHostValve.invoke(StandardHostValve.java:145)
at org.apache.catalina.core.StandardValveContext.invokeNext(StandardValveContext.java:104)
at org.apache.catalina.core.StandardValveContext.invoke(StandardValveContext.java:102)
at org.apache.catalina.core.StandardEngineValve.invoke(StandardEngineValve.java:109)
at org.apache.catalina.core.StandardValveContext.invokeNext(StandardValveContext.java:104)
at org.apache.catalina.core.StandardPipeline.invoke(StandardPipeline.java:520)
at org.apache.catalina.core.ContainerBase.invoke(ContainerBase.java:929)
at org.apache.coyote.tomcat5.CoyoteAdapter.service(CoyoteAdapter.java:160)

\----\ Root Cause \----\njava.lang.NullPointerException
at com.opensymphony.xwork.DefaultActionProxy.<init>(DefaultActionProxy.java:46)
at com.opensymphony.webwork.dispatcher.ServletDispatcher.service(ServletDispatcher.java:111)
at javax.servlet.http.HttpServlet.service(HttpServlet.java:802)

Document generated by Confluence on Dec 06, 2007 19:00
To fix this issue;

1. Stop your Tomcat5 application server.
2. Remove the ‘xml-apis.jar’ and ‘xercesImpl.jar’ from "./<Tomcat_install>/common/endorsed/".
3. Start Tomcat5, and upgrade to 1.2.2.

The error above seems to be persistent with Bamboo 1.2.2, and doesn't affect any other versions,

Should you have problems upgrading, please raise a support request at https://support.atlassian.com/
Can I have Bamboo.home pointing to a resource on a network share?

Yes, it is possible to point Bamboo.home to a directory on your Windows network share. However, you need to specify the absolute path to the network share (e.g. `\Network_computer\path_to_director`).

⚠️ Note:

Please ensure that the user which Bamboo is running as has sufficient privileges to access the network resource.
Changing Bamboo's port from the default 8085

To configure Bamboo to start on a port other than 8085, please see Configuring Bamboo on start-up.

Note: If you are running Bamboo on Linux, you need to start Bamboo on a privileged port (0-1024). You also need to run the Bamboo process as root.
Changing the Root Context Path

When running Bamboo behind a proxy, you might need to change the Root Context Path i.e. the host URL referenced while accessing Bamboo (e.g. http://localhost:8085/bamboo).

To change the context path from '/' to '/Your_Context_Path':

- If you are using the bamboo.sh script to start Bamboo:
  
  Change the following line in your bamboo.sh script:

  ```bash
  RUN_CMD="java \-server \-Xms256m \-Xmx512m \-XX:MaxPermSize=256m \-Djava.awt.headless=true \-
  classpath $CLASSPATH \-Dorg.mortbay.xml.XmlParser.NotValidating=true \-Djetty.port=8085
  com.atlassian.bamboo.server.Server 8085 ./webapp /
  
  to:

  RUN_CMD="java \-server \-Xms256m \-Xmx512m \-XX:MaxPermSize=256m \-Djava.awt.headless=true \-
  classpath $CLASSPATH \-Dorg.mortbay.xml.XmlParser.NotValidating=true \-Djetty.port=8085
  com.atlassian.bamboo.server.Server 8085 ./webapp /Your_Context_Path"
  
  Or, if you are using the wrapper to start Bamboo:
  
  The wrapper reads the configuration information from the wrapper.conf file in the ../<Bamboo-
  Install>/conf/ folder. Find the following line:

  wrapper.app.parameter.4=/

  Replace it with the following line:

  wrapper.app.parameter.4=/Your_Context_Path
  ```
Configuring Bamboo on start-up

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

After editing the options below, Bamboo needs to be shut down and restarted for the changes to take effect.

- Configuring Bamboo's start-up parameters under Linux
- Configuring Bamboo's start-up parameters under Windows

Configuring Bamboo's start-up parameters under Linux

Bamboo on Linux/Unix can be started by either executing the bamboo.sh script or using the wrapper. Either way, the Bamboo server can be customised at start-up.

Modifying the bamboo.sh script.

The bamboo.sh script takes four parameters: start|stop|restart|status.

To customise these parameters at startup, edit the $RUN_CMD variable:

```
RUN_CMD="java -Xms256m -Xmx512m -Djava.awt.headless=true -classpath $CLASSPATH -Dorg.mortbay.xml.XmlParser.NotValidating=true -Djetty.port=8085 com.atlassian.bamboo.server.Server 8085 ./webapp /
```

- java -Xms256m -Xmx512m specifies the minimum and maximum Java Heap size.
- -classpath $CLASSPATH sets the class path at startup.
- -Djetty.port=8085 specifies the port number for the Jetty server.
- com.atlassian.bamboo.server.Server 8085 ./webapp / is the main class that will be executed followed by the context path.

In some cases it might be useful to increase the PermGen space. To do this, add the following parameter to the RUN_CMD variable: "XX:MaxPermSize=512m". This will set the PermGen space to 512mb next time Bamboo is run.

Modifying the wrapper.

The wrapper reads the configuration from wrapper.conf found in ../<BAMBOO_INSTALL>/conf. (The properties are documented inside the file.)

Configuring Bamboo's start-up parameters under Windows

Bamboo can be started in Windows with the startup.bat file (from the command line) or as a Windows Service. Both use the wrapper to start Bamboo. As in Linux (see above), the wrapper reads the configuration from wrapper.conf. Please edit the .../wrapper/wrapper.conf file (situated in the root of your Bamboo_Installation directory) as required.
Getting Bamboo Standalone to use the jetty.xml file

By default Bamboo doesn't use the jetty.xml file to configure itself. If you need to modify the jetty.xml for advanced configuration (such as JNDI or https), you will also need to tell Bamboo to use it.

The method for doing this depends on whether you are using the bamboo.sh startup script or the Java Service Wrapper.

Step 1 - Instructing Bamboo to use jetty.xml

If you are using the bamboo.sh script to start Bamboo:

The standard Bamboo startup script can be customised to use the jetty.xml file by modifying the following section in your bamboo.sh script (this section specifies how the Bamboo server will start):

```
RUN_CMD="java -Xms256m -Xmx512m -Djava.awt.headless=true -classpath $CLASSPATH -Dorg.mortbay.xml.XmlParser.NotValidating=true -Djetty.port=8085 com.atlassian.bamboo.server.Server 8085 ./webapp /
```

Now, modify this startup script to read the jetty.xml file from webapp/WEB-INF/classes/jetty.xml by changing the RUN_CMD argument as follows:

```
RUN_CMD="java -Xms256m -Xmx512m -Djava.awt.headless=true -classpath $CLASSPATH -Dorg.mortbay.xml.XmlParser.NotValidating=true com.atlassian.bamboo.server.Server 8085 ../webapp/WEB-INF/classes/jetty.xml
```

If you are using the Java Service Wrapper to start Bamboo:

When starting up Bamboo with the Java Service Wrapper, you'll need to modify the wrapper.conf file in the conf directory.

- You will need to replace the argument which specifies your port number "wrapper.app.parameter.2=8085" with "wrapper.app.parameter.2=../webapp/WEB-INF/classes/jetty.xml".
- You will need to comment out the other arguments: "wrapper.app.parameter.3=../webapp" and "wrapper.app.parameter.4=/"

This will make Bamboo start up using your jetty.xml configuration file instead of the default three arguments (port, web app directory, context path).

Step 2 - Setting root context web application in jetty.xml

Edit the webapp root context in your jetty.xml file situated in <Bamboo-install>/webapp/WEB-INF/classes/jetty.xml. From

```
<Call name="addWebApplication">
  <Arg>/bamboo</Arg>
  <Arg>
    <SystemProperty name="bamboo.webapp" default="bamboo-web-app/src/main/webapp"/>
  </Arg>
</Call>
</Configure>
```

To

```
<Call name="addWebApplication">
  <Arg>/bamboo</Arg>
  <Arg>
    <SystemProperty name="bamboo.webapp" default="full/path/to/bamboo/install/directory/webapp"/>
  </Arg>
</Call>
```
</Call>
</Configure>
Hardware sizing considerations

For Bamboo, the minimum hardware requirements depend on the size and complexity of your plans. Considerations include:

1. Do your builds will have functional tests as part of the plans?
2. Are your plans executed simultaneously? If so, how many plans will be running at any given time?
3. What are the requirements for your running builds, e.g. do they need large amounts of memory/disk/swap space?
4. How many users will be using Bamboo at any given time? Like any web application, the system resource needed is proportional to the load experienced by the server.

Also see the [Bamboo System Requirements](#).
Installation notes for Bamboo on JBoss 4.x

This page is for people who are deploying the Bamboo EAR/WAR edition on the JBoss 4.x application server.

For full installation instructions please see the Bamboo EAR-WAR Installation Guide.

File extraction notes

To deploy Bamboo EAR-WAR onto your JBoss application server, copy the Bamboo WAR file to ../<JBoss-install>/server/default/deploy/atlassian-bamboo-1.1.2.war.

By default the WAR file will extract to atlassian-bamboo-<version>. The name of the directory in the webapps folder will form the URL required to access Bamboo, e.g. <JBoss-install>/server/default/deploy/atlassian-bamboo-1.1.2.war will become http://host:port/atlassian-bamboo-1.1.2/

How to set Java OPTs on JBoss 4.x

• Windows:
  1. Find the run.bat file.
  2. Edit JAVA_OPTS to set the desired properties variable:

FROM
  if exist "%JBoss_HOME%\bin\native" set JAVA_OPTS=%JAVA_OPTS% -Djava.library.path=%JBoss_HOME% \bin\native

TO
  if exist "%JBoss_HOME%\bin\native" set JAVA_OPTS=%JAVA_OPTS% -Djava.library.path=%JBoss_HOME% \bin\native -server -XX:MaxPermSize=256m -Dbamboo.home=/opt/bamboo/bamboohome -Xmx512m -Djava.awt.headless=true -Dbamboo.home=Your_Path_To_Bamboo $JAVA_OPTS

• Linux-based systems:
  1. Find the run.sh file
  2. Edit JAVA_OPTS to set the desired properties variable:

FROM
  # Setup JBoss specific properties
  JAVA_OPTS="-Dprogram.name=$PROGNAME $JAVA_OPTS"

TO
  # Setup JBoss specific properties
  JAVA_OPTS="-Dprogram.name=$PROGNAME -server -XX:MaxPermSize=256m -Dbamboo.home=/opt/bamboo/bamboohome -Xmx512m -Djava.awt.headless=true -Dbamboo.home=Your_Path_To_Bamboo $JAVA_OPTS"

For further reference
Please visit the JBoss Wiki page on setting JavaOpts
Running Bamboo over HTTPS

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

This document is a guide to configuring Bamboo Standalone with basic HTTPS authentication. For further reference please visit the Jetty page on configuring SSL with Jetty.

1. Generating a certificate with the JDK keytool

The simplest way to generate keys and certificates is to use the keytool application that comes with the JDK, as it generates keys and certificates directly into the keystore.

The following command will generate a key pair and certificate directly into a keystore:

```
keytool -keystore keystore -alias jetty -genkey -keyalg RSA
```

This command will prompt for information about the certificate and for passwords to protect both the keystore and the keys within it. The only mandatory response is to provide the fully qualified host name of the server at the “first and last name” prompt.

Now, we need to configure an SSL listener.

2. Configuring Jetty

Using the Sun JVM, add the SunJsseListener as a HttpListeners, In the ../<Bamboo_Application_Directory>/webapp/WEB-INF/classes/jetty.xml file add the following lines:

```
<Call name="addListener">
  <Arg>
    <New class="org.mortbay.http.SunJsseListener">
      <Set name="Port">8443</Set>
      <Set name="Keystore"><SystemProperty name="jetty.home" default="."/>/keystore</Set>
      <Set name="Password">password</Set>
      <Set name="KeyPassword">password</Set>
    </New>
  </Arg>
</Call>
```

This will make Bamboo accessible in port 8443 on https://localhost:8443/

The keystore file in this example is given relative to the Bamboo Application Directory.

Please ensure that jcert.jar, jnet.jar and jsse.jar are on your classpath.
Setting up JNDI on Jetty

The Bamboo start up script can be customised to setup JNDI resources
Find the following section in your Bamboo.sh script. (This section specifies how the Bamboo server will start)

```
#       This is how the Bamboo server will be started
#
RUN_CMD="java -Xms256m -Xmx512m -Djava.awt.headless=true -classpath $CLASSPATH -Dorg.mortbay.xml.XmlParser.NotValidating=true -Djetty.port=8085 com.atlassian.bamboo.server.Server 8085 ./webapp /
"
```

Now, modify this startup script to read the jetty.xml file from webapp/WEB-INF/classes/jetty.xml, which can be used to specify JNDI resources. Modify the RUN_CMD argument as below

```
RUN_CMD="java -Xms256m -Xmx512m -Djava.awt.headless=true -classpath $CLASSPATH -Dorg.mortbay.xml.XmlParser.NotValidating=true webapp/WEB-INF/classes/jetty.xml
```

You will also need to change the jetty.xml file under webapp/WEB-INF/classes by change the context path from /bamboo to /. Example of this is below:

```
<!-- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -->
<!-- Add root context web applications. -->
<!-- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -->
<Call name="addWebApplication">
<Arg></Arg>
<SystemProperty name="bamboo.webapp" default="webapp"/>
</Arg>
</Call>
```

To set up the JNDI mail session, you will also need to uncomment and modify the section of this jetty.xml which looks like this:

```
<!--
<Call name="addService">
<Arg>
...</Arg>
</Call>
-->
```

You will need to replace the values inside the <Arg> tags with appropriate values (username, password, host, from address).
Supported databases

Bamboo is bundled with a pre-configured HSQL database for evaluation purposes only. We recommend migrating to an external database, if Bamboo is to be used in production. For a list of supported databases, please see 2. Connecting Bamboo to an external database.
Usage FAQ

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

- Backing up Bamboo instances over 4GB
- Binding Bamboo to one IP address
- Can Bamboo build and test non-Java projects?
- Can multiple plans share a common 3rd-party directory?
- Connecting to Subversion repositories
- Deactivating a Bamboo user
- Fixing OutOfMemory Errors in Bamboo
- Problems running Bamboo under Sun JDK 1.4
- Restoring passwords to recover admin users
- Testing LDAP or Active Directory connectivity with Paddle
- Troubleshooting an SVN connection in Bamboo
- Using Bamboo with Clover
- Working with Sun JAVA libraries
Backing up Bamboo instances over 4GB

Due to limitations of the original ZIP file format, and the TrueZIP library used to generate ZIP files, it is not possible to export a Bamboo instance when the resulting ZIP file, or the original size of any of its components, is larger than 4GB. Instead, you will need to backup Bamboo manually. We strongly recommend performing regular backups.

To backup Bamboo manually:

1. Shut down Bamboo.
2. Copy the contents of your Bamboo-Home directory.
3. If you are using an external database, use the database's native backup tool to backup your database (please consult your database documentation for further instructions). Alternatively, perform an SQL dump of your database.

To restore your Bamboo instance to a previous state:

1. Edit the ..../Bamboo-Install-Directory/webapps/WEB-INF/bamboo.init.properties file to point to your backed-up Bamboo-Home directory.
2. If you are using an external database, restore your database using the database's native backup tool.
Binding Bamboo to one IP address

These instructions apply to Bamboo Standalone, which ships with the Jetty application server.

If you have installed Bamboo on a machine with multiple interfaces, and need to bind Bamboo to a single IP address, follow these instructions.

**Step 1 — Instruct Bamboo to read its configuration from the jetty.xml file**

By default Bamboo doesn't use the jetty.xml file to configure itself. You will need to **tell Bamboo to use it**.

**Step 2 — Edit the jetty.xml file**

Your jetty.xml file is located in `<Bamboo_Install_directory>/webapp/WEB-INF/classes/jetty.xml`. Find the following section:

```xml
<Call name="addListener">
  <Arg>
    <New class="org.mortbay.http.SocketListener">
      <Set name="Port">
        <SystemProperty name="jetty.port" default="8085"/>
      </Set>
      <Set name="Host">127.0.0.1</Set>
    </New>
  </Arg>
</Call>
```

Change the last line as follows:

```xml
<Call name="addListener">
  <Arg>
    <New class="org.mortbay.http.SocketListener">
      <Set name="Port">
        <SystemProperty name="jetty.port" default="8085"/>
      </Set>
      <Set name="Host">YOUR_HOST_URL</Set>
    </New>
  </Arg>
</Call>
```

**Please note:** YOUR_HOST_URL should be the same as the Bamboo base URL configured in Bamboo.

**Step 3 — Restart Bamboo**
Can Bamboo build and test non-Java projects?

Bamboo can be ported to be used on any architecture and can build projects in virtually any language/script (Java, C++, ruby, perl, VB.net, bash, make and C# to name a few of many projects currently built with Bamboo).

Bamboo can execute any script/build that has a return code after the build process is completed. Ideally, you would configure a build tool (such as Maven or Ant) to build your code. Bamboo will then call on the build tool to build your project (depending on how your build process is configured).

Regarding tests, Bamboo uses JUnit tests to integrate test results with Java and is capable of reading test results from any testing framework that outputs to a Junit XML report.
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.
Connecting to Subversion repositories

Note:
Bamboo 1.0.4 does not authenticate with Subversion. Please upgrade Bamboo to 1.0.5 or above (we recommend upgrading to the latest version of Bamboo).

Cannot validate Subversion repository during plan configuration

The most common repository error is:

This is not a valid Subversion Repository: svn: Authentication required for '<https://svn.atlassian.com:443> Atlassian Subversion Repository'

In all likelihood, the authentication failed due to invalid login credentials. Please double-check your username and password and try again.

If you are still having trouble connecting to repository,

1. Try logging in outside of Bamboo, from the command line or from any other repository client. If this step fails, then it's likely that your repository is not configured correctly.
2. Please the consult our JIRA issue tracker for known Bamboo repository issues.
3. If you still can't connect to Subversion, please [click here](http://support.atlassian.com) to raise a support request. We will get back to you ASAP.

Debugging Subversion connectivity

Bamboo uses SVNKit library for SVN connectivity. In some cases, you may find it useful to debug Subversion. To do this you need to instruct the Java VM to enable logging, by modifying the following system property while launching the Java VM:

```java
-Djava.util.logging.config.file=path/to/logging.properties.disabled
```

To modify the above system property:

1. Download the `logging.properties.disabled` file attached to this document.
2. Modify Bamboo to start with `-Djava.util.logging.config.file=path/to/logging.properties.disabled` command where "path/to/" refers to absolute path to logging.properties.disabled file from step (1).
   - If you are running Bamboo under Linux, modify the `bamboo.sh` script to pass in the logging parameter. To do this, find the following section:
     ```bash
     RUN_CMD= "java \-Xms256m \-Xmx512m \-XX:MaxPermSize=256m \-Djava.awt.headless=true \-classpath $CLASSPATH \-Dorg.mortbay.xml.XmlParser.NotValidating=true \-Djetty.port=8085 com.atlassian.bamboo.server.Server 8085 ./webapp /
     
     Change the RUN_CMD variable to:
     
     RUN_CMD="java \-Xms256m \-Xmx512m \-XX:MaxPermSize=256m \-Djava.awt.headless=true \-classpath $CLASSPATH \-Dorg.mortbay.xml.XmlParser.NotValidating=true \-Djava.util.logging.config.file=path/to/logging.properties.disabled \-Djetty.port=8085 com.atlassian.bamboo.server.Server
     
     - If you are running Bamboo under Windows, modify the `<Bamboo_install_Home>/conf/wrapper.conf` file to add the logging parameter. To do this, find the following section:
     ```
     wrapper.java.classpath.1=../lib/*.jar
     wrapper.java.vmargs=-Djava.util.logging.config.file=path/to/logging.properties.disabled
     ```
     Add the following line to the end of the section:
     ```
     wrapper.java.vmargs=-Djava.util.logging.config.file=path/to/logging.properties.disabled
     ```
3. Save the changes, and restart Bamboo for your changes to take effect.
4. You will find the SVNKit log file in \texttt{USER\_HOME/svnkit.0.log} (where \texttt{USER\_HOME} is the home directory of the user running Bamboo or logged in), e.g: \texttt{/home/aj/svnkit.0.log}.
Deactivating a Bamboo user

To deactivate a Bamboo user account (rather than deleting it), change the password so that the user cannot login. To do this,

1. Click the 'Administration' link in the top navigation bar.
2. Click the 'Users' link in the left navigation column.
3. The 'Manage Users' screen will be displayed. Locate the relevant user in the list, and click the corresponding 'Edit' link in the 'Operations' column.
4. In the 'Email' field, enter an invalid email address.
5. Click the 'Save' button.
6. Log out of Bamboo. The Bamboo 'Login' screen will be displayed.
7. Click the 'Forgotten your password' link.
8. The 'Forgotten your password' screen will be displayed. In the 'Username' field, enter the Bamboo username of the user account you wish to deactivate.
9. Click the 'Send me a new password' button.
Fixing OutOfMemory Errors in Bamboo

I am getting Out of Memory errors, how can I allocate more memory to Bamboo?

Since the default memory setting usually is 256MB in Bamboo, you might have to adjust the settings to run a bigger Bamboo instance with sufficient memory.

On Linux

- In the unpacked bamboo standalone directory, edit the file `bamboo.sh`
- Edit the line beginning with `RUN_CMD=` substituting new values for `-Xms` (starting memory) and `-Xmx` (maximum memory)
- Leave the rest of the options in that line unchanged

An example of a minimal setting for a large system - max heap size is set to 768 megabytes:

```
RUN_CMD="java -server -Xms512m -Xmx768m -XX:MaxPermSize=256m -Djava.awt.headless=true
-classpath "$CLASSPATH" -Dorg.mortbay.xml.XmlParser.NotValidating=true -Djetty.port=8085
com.atlassian.bamboo.server.Server 8085 ./webapp /"
```

- Leave the rest of the options in that line unchanged

On Windows

Bamboo uses wrapper to start, in Bamboo either as a service or in console

- The wrapper reads the configuration from wrapper.conf found in BAMBOO_INSTALL/conf. The parameters in file are well documented and is out of the scope of this article.
- Edit the `-Xms` value to increase your Bamboo memory

Other Notes

Allocating too much memory to your JVM Heap can also cause OutOfMemory Errors.

java.lang.OutOfMemoryError: unable to create new native thread

This error occurs when the operating system is unable to create new threads. This is due to the JVM Heap taking up the available RAM. Big heaps take away from the space that can be allocated for the stack of a new thread.

For Linux the maximum heap size of the JVM cannot be greater than 2GB. If you only have 2GB RAM in your server, it is not recommended to set the Max size of the JVM that high.

The size of the stack per thread can also contribute to this problem. The stack size can reduce the number of threads that can be created.

To fix this problem, you should reduce the size of your JVM Heap and also the size of the stack per thread.

The stack size can be changed with the following (example) parameter:

```
-Xss512k
```

Please refer to this guide as a reference for JVM tuning.

Permanent Generation Size

If you get the error message: `java.lang.OutOfMemoryError: PermGen space` this means that you have exceeded Java's fixed 64Mb block for loading class files. You will need to add the argument `-XX:MaxPermSize` and increase the memory.

JDK 1.4 does not provide information why the OutOfMemory error occurred.
<table>
<thead>
<tr>
<th></th>
<th>JDK 1.5 and above are recommended as it provides description of the error as in the above example.</th>
</tr>
</thead>
</table>

Problems running Bamboo under Sun JDK 1.4

Backing up Bamboo

With Bamboo versions 1.2.1 and 1.2.2, exporting Bamboo when running under Sun JDK version 1.4 will fail due to incompatible with JAVA versions.

⚠️ Exporting Bamboo

- [This issue](#) was fixed in Bamboo 1.2.3.

Bamboo version 1.2 will not start under Sun JDK 1.4

It is recommended that users upgrade to Bamboo 1.2.1 or above as a fix for [this issue](#).
Restoring passwords to recover admin users

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

Use this document if you are unable to login as administrator or have forgotten your password and do not have Mail Server configured, to manually replace administrator passwords.

Follow the instructions for either the Embedded Database or External Database. If you have not configured a database, use the Embedded instructions.

Embedded Database Instructions

Stage One - Identify Administrator

This guide assumes that the first user added was an administrator. If this is not the case, search for the admin username and find their user id number, then modify their password hash instead.

1. Shutdown Bamboo
2. In your Bamboo home directory, open `database\defaultdb.script` file in a text editor
3. Search for the text:

   `INSERT INTO USERS VALUES(1
   To find the administrator login entry:

   `INSERT INTO USERS VALUES(1,'USERNAME','PASSWORD_HASH')

   Where the 1 is the user id number, and USERNAME and PASSWORD_HASH are actual values.

   As an example, my table entry for user admin with password admin looks like this;

   `INSERT INTO USERS
   VALUES(1,'admin','x61Ey612Kl2gpFL56FT9weDnpSo4AV8j8+qx2AuThdRyY036xxzTTrw10Wq3+4qQyB+xURFPw3ONxp3Y3pB37A==','admin@admin.com','2007-08-14 11:26:18.504000000','admin')

1. This step makes admin the administrator's password. Bamboo does not store passwords in plain text in the database, but uses hashes computed from the original password. The hash for the characters admin is below:

   `x61Ey612Kl2gpFL56FT9weDnpSo4AV8j8+qx2AuThdRyY036xxzTTrw10Wq3+4qQyB+xURFPw3ONxp3Y3pB37A==`

   Paste the admin password hash between the '' characters of their existing PASSWORD_HASH. The new administrator login entry should look like:

   `INSERT INTO USERS
   VALUES(1,'USERNAME','x61Ey612Kl2gpFL56FT9weDnpSo4AV8j8+qx2AuThdRyY036xxzTTrw10Wq3+4qQyB+xURFPw3ONxp3Y3pB37A==','EMAIL','DATE_TIME','FULL_NAME')`

   Where USERNAME is the administrator username.

2. Save the file
3. Start up Bamboo
4. Login with the administrator username and password admin

External Database Instructions

Stage One - Identify User

The first user added is always an admin. To restore your password you simply need to update the password hash in the USERS table with the admin hash.

Connect to your database using a database admin tool such as DBVisualiser. Please download a database admin tool now if you do not have one installed already. Once installed, connect to your database and retrieve the list of administrator usernames with:
Stage Two - Replace Administrator Password

Bamboo does not store passwords in plain text in the database, but uses hashes computed from the original password. You instead cut and paste a hash, rather than the plain password, over the existing password. Below is the hash for the password admin:

$x61Ey612Kl2gpFL56FT9weDnpSo4AV8j8+qx2AuTHdRyY036xxzTTkw10Wq3+4qQyb+XURFWx1ONxp3Y3pB37A==$

To change the password to admin for a given username:

1. Shutdown Bamboo
2. Connect to your database. Run this SQL on your database:
   
   ```
   select * from USERS where ID=1
   ```
   
   This command should list all users who belong to Bamboo-Admin user group.

   ```
   update USERS set PASSWORD = 'x61Ey612Kl2gpFL56FT9weDnpSo4AV8j8+qx2AuTHdRyY036xxzTTkw10Wq3+4qQyb+XURFWx1ONxp3Y3pB37A=='
   where NAME = 'USER_NAME_FROM_STAGE_ONE'
   ```

3. Start Bamboo
4. Login with your username and your password is now admin
Testing LDAP or Active Directory connectivity with Paddle

Introduction

Paddle is a tool that will test the LDAP or Active Directory settings in your `atlassian-user.xml`.

Instructions for use

You do not need to have Bamboo running to run this tool. The steps are:

1. Download into a directory where you have permissions to create files.
2. Copy your `atlassian-user.xml` into that directory - this is found in your `<Bamboo-Install>/webapp/WEB-INF/classes/` directory.
3. Run `java -jar paddle-x.x.jar` (where x.x is the version of Paddle you downloaded).

Parameters

Paddle currently supports the following parameters:

<table>
<thead>
<tr>
<th>Name</th>
<th>Example</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>debug</td>
<td><code>java -jar paddle-x.x.jar debug</code></td>
<td>Prints DEBUG messages to the console as well as paddle.log.</td>
</tr>
<tr>
<td>limit</td>
<td><code>java -jar paddle-x.x.jar limit=100</code></td>
<td>Sets the limit on the number of results returned by user and group queries. Defaults to 10.</td>
</tr>
</tbody>
</table>

Sample output

This is an example of a successful run:

```
###########################################################################################################################
LDAP Support Tool version 1.1
###########################################################################################################################
Connection to LDAP/Active Directory Server at ldap://192.168.0.86:389 SUCCESSFUL.

TEST 1: Search and list 10 users

User: CN=Administrator
Member of:
(1) CN=Schema Admins
(2) CN=Enterprise Admins
(3) CN=Domain Admins
(4) CN=Group Policy Creator Owners

User: CN=Guest
Does not belong to any LDAP groups.

User: CN=SUPPORT_388945a0
Member of:
(1) CN=HelpServicesGroup

User: CN=IUSR_MALTSHOVEL
Does not belong to any LDAP groups.
```
User: CN=IWAM_MALTSHOVEL
Member of:
(1) CN=IIS_WPG

User: CN=ASPNET
Does not belong to any LDAP groups.

User: CN=krbtgt
Does not belong to any LDAP groups.

User: CN=John\, Smith
Member of:
(1) CN=Domain Users
(2) CN=Sales and Marketing

User: CN=Matt Ryall
Member of:
(1) CN=Enterprise Admins
(2) CN=Domain Admins

User: CN=Justin Koke
Member of:
(1) CN=Domain Controllers
(2) CN=Enterprise Admins

Found more than 10 results.

-----------------------------------------------------------------
TEST 2: Search and list 10 groups
-----------------------------------------------------------------

Group: CN=HelpServicesGroup
Members:
(1) CN=SUPPORT_388945a0,CN=Users,DC=ad,DC=atlassian,DC=com

Group: CN=TelnetClients
No members in this group.

Group: CN=IIS_WPG
Members:
(1) CN=S-1-5-20,CN=ForeignSecurityPrincipals,DC=ad,DC=atlassian,DC=com
(2) CN=S-1-5-6,CN=ForeignSecurityPrincipals,DC=ad,DC=atlassian,DC=com
(3) CN=S-1-5-18,CN=ForeignSecurityPrincipals,DC=ad,DC=atlassian,DC=com
(4) CN=IWAM_MALTSHOVEL,CN=Users,DC=ad,DC=atlassian,DC=com

Group: CN=SQLServer2005SQLBrowserUser$MALTSHOVEL
Members:
(1) CN=S-1-5-18,CN=ForeignSecurityPrincipals,DC=ad,DC=atlassian,DC=com

Group: CN=SQLServer2005MSSQLServerADHelperUser$MALTSHOVEL
Members:
(1) CN=S-1-5-20,CN=ForeignSecurityPrincipals,DC=ad,DC=atlassian,DC=com

Group: CN=SQLServer2005SQLAgentUser$MALTSHOVEL$MSSQLSERVER
Members:
(1) CN=S-1-5-18,CN=ForeignSecurityPrincipals,DC=ad,DC=atlassian,DC=com

Group: CN=SQLServer2005MSSQLUser$MALTSHOVEL$MSSQLSERVER
Members:
(1) CN=S-1-5-18,CN=ForeignSecurityPrincipals,DC=ad,DC=atlassian,DC=com

Group: CN=SQLServer2005MSFTEUser$MALTSHOVEL$MSSQLSERVER
Members:
(1) CN=S-1-5-18,CN=ForeignSecurityPrincipals,DC=ad,DC=atlassian,DC=com
Group: CN=SQLServer2005MSOLAPUser$MALTSHOVEL$MSSQLSERVER
Members:
(1) CN=S-1-5-18,CN=ForeignSecurityPrincipals,DC=ad,DC=atlassian,DC=com

Group: CN=SQLServer2005NotificationServicesUser$MALTSHOVEL
No members in this group.

Found more than 10 results.
Troubleshooting an SVN connection in Bamboo

To troubleshoot your SVN connection, it is useful to enable logging. Modify the system property while launching the Java VM with:

$java -Djava.util.logging.config.file=path/to/logging.properties.disabled

To do this:
1) Download the logging.properties.disabled file attached to this document.

The logging.properties.disabled file contains a system property svnkit.level used to control the log level:

- FINE (default) — the level for non-detailed logging
- FINER — the level for more detailed logging
- FINEST --- the level for full logging

2) Configure Bamboo to start with the -Djava.util.logging.config.file=*path/to/*logging.properties.disabled command, where "path/to/" refers to the absolute path to the logging.properties.disabled file from step (1).

The method of doing this depends on which operating system you are using:

- If you are running Bamboo under Linux:
  Modify the bamboo .sh script to pass in the logging parameter. To do this, find the following section:

RUN_CMD= "java \-Xms256m \-Xmx512m \-XX:MaxPermSize=256m \-Djava.awt.headless=true \-classpath $CLASSPATH \-Dorg.mortbay.xml.XmlParser.NotValidating=true\nbasp; \-Djetty.port=8085 com.atlassian.bamboo.server.Server 8085 ./webapp /

Change the RUN_CMD variable to:

RUN_CMD= "java \-Xms256m \-Xmx512m \-XX:MaxPermSize=256m \-Djava.awt.headless=true \-classpath $CLASSPATH \-Dorg.mortbay.xml.XmlParser.NotValidating=true \-Djava.util.logging.config.file=*path/to/*logging.properties.disabled \-Djetty.port=8085 com.atlassian.bamboo.server.Server\nbsb; 8085 ./webapp /

- If you are running Bamboo under Windows:
  Modify the <Bamboo_install_Home>/conf/wrapper.conf file to add the logging parameter. To do this, find the following section:

```
...........................
wrapper.java.classpath.1=../lib/*.jar
wrapper.java.classpath.2=..\lib
wrapper.java.classpath.3=../webapp/WEB-INF/classes
wrapper.java.classpath.4=../webapp/WEB-INF/lib/*.jar
wrapper.java.library.path.1=../lib
wrapper.java.additional.1=Dorg.mortbay.xml.XmlParser.NotValidating=true
wrapper.java.additional.2=\-XX:MaxPermSize=256m
wrapper.java.additional.3=Djava.awt.headless=true
...........................
```

Add the following line to the end of the section:

```
wrapper.java.additional.4=\-Djava.util.logging.config.file=*path/to/*logging.properties.disabled
```

3) Save your changes, and restart Bamboo for your changes to take effect.

4) You will find the SVNKit log file in USER_HOME/svnkit.0.log (where USER_HOME is the home directory of the user running Bamboo or logged in e.g: /home/user/svnkit.0.log).
Using Bamboo with Clover

Getting started

To use Clover with Bamboo, you need to:

1. Either:
   • call the Clover goal in your plan configuration (see 2.1 Specifying a Plan's Build Resources);
   or:
   • add the maven-clover-plugin report to the reports section in your POM.
2. Ensure that there are tests present in your build plan that generate test results in JUnit test report format.
3. Ensure that your build creates a Clover report (that is, a clover.xml file). Bamboo will use this Clover report as source.
4. Set up Bamboo to read the Clover report (clover.xml file) generated by Clover. To do this:
   a. Ensure the 'Clover output will be produced' check-box is ticked in your plan's build configuration page.
   b. Instruct Bamboo on the location of your 'Clover XML Directory' — where Bamboo will look for the XML report output file from Clover. Please specify file path relative to your plan's root directory (e.g. /home/bamboouser/bamboo-home/xml-data/build-dir/MY_PLAN/), i.e. please do not specify an absolute path.

For further details, please see 2.1 Specifying a Plan's Build Resources.

Q&A:

Q: I have managed to get Clover statistics displayed in numerical form for each build, but the graphs do not show a history of these statistics?
A: The history of Clover is displayed over time periods (e.g. a day, a week, a month), and the minimum data point is per day. The Clover coverage will not display data that is less than a day old.

Q: Will the Bamboo/Clover integration run on failed builds?
A: Before Bamboo version 1.2.1, Bamboo would only report Clover coverage for successful builds. As of Bamboo 1.2.1, Bamboo will report Clover coverage regardless of the build outcome.
Due to licensing restrictions, we are not allowed to re-distribute native SUN libraries through our maven2 public repositories.

If you are developing plugins for Bamboo or building Bamboo from source, you might need `javax.mail` and `javax.transaction:jta:jar` for Bamboo to build successfully. The relevant POMs for this look something like this:

```
......
<dependency>
  <groupId>javax.mail</groupId>
  <artifactId>mail</artifactId>
  <version>1.3.2</version>
  <scope>compile</scope>
</dependency>
<dependency>
  <groupId>jta</groupId>
  <artifactId>jta</artifactId>
  <version>1.0.1</version>
  <scope>compile</scope>
</dependency>
......
```

Before building, please install the Sun JAR's into your local Maven2 repositories by following the instructions below.

To install the `javax.mail` JAR into your local Maven2 repository:

1. Download the `javax.mail` Jar from Sun's website.
2. Install it on your local machine by entering the following command in a terminal:

   ```
   mvn install:install-file -DgroupId=javax.mail -DartifactId=mail -Dversion=1.3.3 -Dpackaging=jar -Dfile=YOUR/PATH/TO/FILE
   ```

To install `javax.transaction:jta:jar` JAR into your local Maven2 repository:

1. Download the `javax.transaction:jta:jar` Jar from Sun’s website.
2. Install it on your local machine by entering the following command in a terminal:

   ```
   mvn install:install-file -DgroupId=javax.transaction -DartifactId=jta -Dversion=1.0.1B -Dpackaging=jar -Dfile=YOUR/PATH/TO/FILE
   ```
Bamboo User's Guide

This page last changed on Aug 09, 2007 by rosie@atlassian.com.

### About

Bamboo is a **continuous integration** (CI) server. Bamboo assists software development teams by providing:

- automated building and testing of software source-code status.
- updates on successful/failed builds.
- reporting tools for statistical analysis.

The Bamboo User's Guide provides information about using Bamboo. If you need information about installing Bamboo or configuring builds, please visit [Bamboo Documentation Home](#).

If you have a question about using Bamboo that hasn't been answered here, please [let us know](#).

### Download

You can download the Bamboo documentation in PDF, HTML or XML formats.

### Search the User's Guide

### Getting Help

[Support](#) | [Feature Request](#) | [Forums](#) | [Knowledge Base](#)

---

**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](#)
  - [1.4.1 Adding a Plan to your Favourites](#)
  - [1.4.2 Removing a Plan from your Favourites](#)
- [1.5 Displaying a Build Monitor](#)

### 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 the Code Changes that triggered a Build](#)
- [3.4 Viewing a Build's Artifacts](#)
- [3.5 Viewing a Build Log](#)
- [3.6 Viewing the Metadata for a Build Result](#)
- [3.7 Viewing the Clover Code-Coverage for a Build Result](#)
- [3.8 Viewing the JIRA Issues for a Build Result](#)

### 04. Working with Tests

- [4.1 Viewing Test Results for a Build](#)
• 4.2 Viewing a Test's History
• 4.3 Viewing Test Statistics for a Plan

05. Reporting on Plan Trends

• 5.1 Viewing Build Statistics for a Plan
• 5.2 Generating Reports across multiple Plans
  ° 'Build Activity per Plan' Report
  ° 'Build Duration per Plan' Report
  ° 'Clover Code Coverage per Plan' Report
  ° 'Clover Lines of Code per Plan' Report
  ° 'Number of Build Failures per Plan' Report
  ° 'Number of Tests per Plan' Report
  ° 'Percentage of Successful Builds per Plan' Report
  ° 'Time to Fix per Plan' Report

06. Reporting on Author Trends

• 6.1 Viewing Build Statistics for all Authors
• 6.2 Viewing Build Results for an Author
• 6.3 Generating Reports on selected Authors
  ° 'Build Activity per Author' Report
  ° 'Number of Build Failures per Author' Report
  ° 'Number of Builds Broken per Author' Report
  ° 'Number of Builds Fixed per Author' Report
  ° 'Percentage of Successful Builds per Author' Report

07. Working with Comments

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

08. Working with Labels

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

09. Subscribing to RSS Feeds

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

10. Working with Instant Messenger (IM) Notifications

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

11. Editing your User Profile

• 11.1 Changing your Password
• 11.2 Changing your Notification Preferences
• 11.3 Associating your Author Name with your User Profile
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
- 1.5 Displaying a Build Monitor
1.1 Using the Bamboo Dashboard

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

The Dashboard is your Bamboo 'home' page. The Dashboard contains three tabs:

- 'All Plans' — a list of build plans and each plan's latest build result.
- 'Current Activity' — Bamboo’s build queues, showing which plans Bamboo is currently building and which plans are waiting to be built.
- 'My Bamboo'¹ — a convenient summary of information that is relevant to you:
  - plans which you have nominated as your favourites.
  - your latest build results (i.e. builds that were triggered by your latest code changes).
  - a summary of your build statistics².

Screenshot: Bamboo Dashboard-'All Plans' tab

Each blue name (e.g. 'Atlassian Bucket') represents a project. The name(s) to the right of each project name are the plan(s) (e.g. 'Main Build') belonging to that project, and the build number (e.g. 'BUCKET-MAIN-74') represents the latest build result for the plan, while the icon indicates the plan's current status:

- 
- This plan's latest build was successful.
- ! This plan's latest build failed.
- ↔ Bamboo is currently checking-out the source-code for this plan, in preparation for starting a build.
- o Bamboo is currently executing a build for this plan.
- ✗ This plan has been disabled.

You can:

- click the plan name (e.g. 'Main Build') to view the plan details.
- click the build number (e.g. 'BUCKET-MAIN-74') to view the build result.
- click the author's name to view the author's details (the author is the person who triggered the build by checking-in code).

Handy Hint
You can return to the Dashboard from anywhere in Bamboo by clicking the 'Home' link in the top navigation bar.

¹ only if you have logged in to Bamboo.
² only if your Bamboo User Profile has been associated with your Author Name.

RELATED TOPICS

- 1.1 Using the Bamboo Dashboard
- 1.2 Viewing Bamboo's Current Activity
• 1.3 Viewing your Latest Build Results
• 1.4 Working with Favourites
  ° 1.4.1 Adding a Plan to your Favourites
  ° 1.4.2 Removing a Plan from your Favourites
• 1.5 Displaying a Build Monitor

Bamboo Glossary
Return to Bamboo Documentation Home
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.

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

To view Bamboo’s current activity,

1. Click the ‘Home’ link in the top navigation bar. This will display the Dashboard.
2. Click the ‘Current Activity’ tab. This will display Bamboo’s Build Queues, as well as a list of Recently Completed Builds.

The above screenshot shows a Bamboo system that has two Build Queues: ‘Fast Builds’ and ‘Queue 2’. The ‘Fast Builds’ queue is currently building a plan called ‘Crowd - Main Build’. You can:

- click a plan name (e.g. ‘Crowd - Main Build’) to view the plan details.
- click the icon to view the plan’s continuous scrolling activity log.

Additionally, in the ‘Recently Completed Builds’ section, you can:

- click a build number (e.g. ‘CWD-MAIN-189’) to view the build result.
- click a ‘Reason’ (e.g. ‘Updated by...’) to view the code changes that triggered the build.

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
- 1.5 Displaying a Build Monitor

Bamboo Glossary

Return to Bamboo 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'

<table>
<thead>
<tr>
<th>My Latest Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Build</td>
</tr>
<tr>
<td>BAM-MAIN-1846</td>
</tr>
<tr>
<td>BAM-MAIN-1844</td>
</tr>
<tr>
<td>BAM-MAIN-1842</td>
</tr>
<tr>
<td>BAM-MAIN-1842</td>
</tr>
<tr>
<td>BAM-MAIN-1840</td>
</tr>
<tr>
<td>BAM-MAIN-1839</td>
</tr>
<tr>
<td>BAM-MAIN-1838</td>
</tr>
<tr>
<td>BAM-MAIN-1836</td>
</tr>
<tr>
<td>BAM-MAIN-1834</td>
</tr>
<tr>
<td>BAM-MAIN-1833</td>
</tr>
</tbody>
</table>

If your Bamboo User Profile has not yet been associated with your Author Name, there will be no 'My Latest Changes' section.

RELATED TOPICS

• 1.1 Using the Bamboo Dashboard
• 1.2 Viewing Bamboo's Current Activity
• 1.3 Viewing your Latest Build Results
• 1.4 Working with Favourites
  • 1.4.1 Adding a Plan to your Favourites
  • 1.4.2 Removing a Plan from your Favourites
• 1.5 Displaying a Build Monitor

Bamboo Glossary
Return to Bamboo Documentation Home
1.4 Working with Favourites

Whereas the 'All Plans' tab on the Bamboo Dashboard lists every plan that exists in your Bamboo system, the 'My Bamboo' tab lists just your chosen favourites — that is, the plans you work with the most. You can easily add and remove plans from your favourites.

When you add a plan to your favourites, you become a 'watcher' of the plan. This means that you may receive notifications about the build results for some or all of your favourite plans, depending on how your administrator has configured each plan's notifications. You can choose whether you would like to receive your notifications by email and/or Instant Messaging (IM).

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. A yellow star is shown next to each.

Screenshot: 'My Bamboo--My Favourite Plans'

**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
- 1.5 Displaying a Build Monitor

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

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

To add a plan to your favourites,

1. Click the 'Home' link in the top navigation bar. This will display the Dashboard.
2. Click the 'All Plans' tab.
3. This will display a list of all plans in your Bamboo system. (Note: Plans that have already been added to your favourites are indicated by a yellow star icon. Plans that have not been added to your favourites are indicated by a grey star icon.)
4. Locate the plan and click the grey star icon:
5. Click the 'My Bamboo' tab.
6. Verify that the plan is now listed in the 'My Favourite Plans' section.

Handy Hint

You are now a 'watcher' of the plan. This means that you may receive notifications about the build results for this plan, depending on how your administrator has configured the plan's notifications. You can choose whether you would like to receive your notifications by email and/or Instant Messaging (IM).

Handy Hint

If your administrator has enabled 'Auto-Favourites', each plan will be automatically added to your favourites the first time you check-in code for that plan.

RELATED TOPICS

- 1.1 Using the Bamboo Dashboard
- 1.2 Viewing Bamboo's Current Activity
- 1.3 Viewing your Latest Build Results
- 1.4 Working with Favourites
  - 1.4.1 Adding a Plan to your Favourites
  - 1.4.2 Removing a Plan from your Favourites
- 1.5 Displaying a Build Monitor
1.4.2 Removing a Plan from your Favourites

To remove a plan from your favourites,

1. Click the 'Home' link in the top navigation bar. This will display the Dashboard.
2. Click the 'All Plans' tab.
3. This will display a list of all plans in your Bamboo system. (Note: Plans that have been added to your favourites are indicated by a yellow star icon. Plans that have not been added to your favourites are indicated by a grey star icon.)
4. Locate the plan and click the yellow star icon:
5. Click the 'My Bamboo' tab.
6. Verify that the plan is not listed in the 'My Favourite Plans' section.

RELATED TOPICS

- 1.1 Using the Bamboo Dashboard
- 1.2 Viewing Bamboo's Current Activity
- 1.3 Viewing your Latest Build Results
- 1.4 Working with Favourites
  - 1.4.1 Adding a Plan to your Favourites
  - 1.4.2 Removing a Plan from your Favourites
- 1.5 Displaying a Build Monitor

Bamboo Glossary
Return to Bamboo Documentation Home
1.5 Displaying a Build Monitor

Sometimes a development team can benefit from setting up a monitor to display Bamboo's latest build results, e.g.:

![Build Monitor Image]

You can choose to display Bamboo's latest results for your favourite plans only, or for all plans that you have permission to see.

To display Bamboo's latest build results, for all plans,

1. Log into Bamboo. (Note: if your Bamboo administrator has allowed anonymous access, this step is optional.)
2. Type the following URL into your browser, but replace 'bambooserver' with the real name of your Bamboo server:
   
   http://bambooserver:8080/bamboo/telemetry.action

To display Bamboo's latest build results, for your favourite plans only,

1. Log into Bamboo. (Note: only logged-in users can have favourites.)
2. Type the following URL into your browser, but replace 'bambooserver' with the real name of your Bamboo server:
   

Hint
If you are going to display the build monitor permanently, you may want to ask your Bamboo administrator to create a user who has only a limited set of permissions.

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
- 1.5 Displaying a Build Monitor

Bamboo Glossary
Return to Bamboo Documentation Home
02. Working with Projects and Plans

2. Working with Projects and Plans

- 2.1 About Projects and Plans
- 2.2 Viewing a Plan's Details
- 2.3 Viewing a Plan's Activity Log
2.1 About Projects and Plans

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

Every plan belongs to a project.

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

Every Bamboo plan is listed on the Dashboard, from where you can:

- Click on a Plan Name to view the plan details
- Click on a Build Number to view the plan's latest build result

Projects and plans can only be configured by Bamboo administrators. Please see the Bamboo Administrator's Guide for details.

RELATED TOPICS

- 2.1 About Projects and Plans
- 2.2 Viewing a Plan's Details
- 2.3 Viewing a Plan's Activity Log

Bamboo Glossary

Return to Bamboo Documentation Home
2.2 Viewing a Plan's Details

To view a plan's details,

- From the Dashboard, locate and click a Plan Name from the list;
- OR:
- From within a build result, click the Plan Name at the top left of the screen.

The Plan Summary will be displayed as follows:

**Screenshot: 'Plan Summary'

In the above screenshot:

The green box indicates that this plan's latest build was successful. Note that a red box in this position would indicate that the plan's latest build failed, while a blue box would indicate that a build is currently in progress.
• Click the build number (i.e. 'BUCKET-MAIN-71') to view the build result.
• Click the 'Updated by' link to view the code changes that triggered the latest build result.
• (Build duration) is the total time taken to execute a build plan. That is, the time taken to compile the code and run all of the plan's tests.

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.

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

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

Every plan has an activity log. An activity log is a temporary display of the latest output from the plan’s most recent build log.

To view a plan’s activity log,

1. From the Dashboard, locate and click a plan name from the list; OR:
   From within a build result, click the plan name at the top left of the screen.
2. Click the ‘Activity Log’ tab.

The plan’s current activity will be displayed. For example, the following screenshot shows a plan for which a build is currently in progress:

Screenshot: Plan Activity

You can also monitor a plan’s build activity over time by using the ‘Build Activity per Plan’ report.

Build activity is the number of builds that occur in a given period of time.

RELATED TOPICS

- 2.1 About Projects and Plans
- 2.2 Viewing a Plan’s Details
- 2.3 Viewing a Plan’s Activity Log

Bamboo Glossary

Return to Bamboo Documentation Home
03. Working with Build Results

3. Working with Build Results

- 3.1 About Builds and Build Results
- 3.2 Viewing a Build Result
- 3.3 Viewing the Code Changes that triggered a Build
- 3.4 Viewing a Build's Artifacts
- 3.5 Viewing a Build Log
- 3.6 Viewing the Metadata for a Build Result
- 3.7 Viewing the Clover Code-Coverage for a Build Result
- 3.8 Viewing the 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](https://example.com). Bamboo can also send [notifications](https://example.com) and generate [RSS feeds](https://example.com) about build results.

**RELATED TOPICS**

- 3.1 About Builds and Build Results
- 3.2 Viewing a Build Result
- 3.3 Viewing the Code Changes that triggered a Build
- 3.4 Viewing a Build's Artifacts
- 3.5 Viewing a Build Log
- 3.6 Viewing the Metadata for a Build Result
- 3.7 Viewing the Clover Code-Coverage for a Build Result
- 3.8 Viewing the JIRA Issues for a Build Result

[Bamboo Glossary](https://example.com)

Return to [Bamboo Documentation Home](https://example.com)
3.2 Viewing a Build Result

To view a plan's most recent build result:

1. Go to the Dashboard.
2. Locate the plan in the list, then click the Build Number.

To view all build results for a plan:

1. Go to the relevant plan.
2. Click the 'Completed Builds' tab to see a summary list of build results.
   To view the details for a particular build result, click the Build Number in the list.

A build result looks like this:

**Screenshot : 'Build Result Summary'**

In the above screenshot:
- You can click the plan name ('Main Build' in this example) to see the plan details for this build:

  ![Plan Details](image)

- You can use the build results navigator to scroll through other build results for this plan:

  ![Build Results Navigator](image)

- The 'Summary' tab shows a snapshot of the build result. To see more detail:
  - Click the 'Tests' tab to view the build's test results.
  - Click the 'Changes' tab to view the code changes that triggered this build (if applicable).
  - Click the 'Artifacts' tab to view any artifacts relating to this build.
  - Click the 'Logs' tab to view a complete build log.
  - Click the 'Comments' tab to view a trail of comments regarding this build result. You can also click the following icon to add a comment:

    ![Add Comment](image)

  - Click the 'Metadata' tab to view any metadata that relates to this build result.

- Depending on how your Bamboo administrator has configured the system, the following additional tabs may be available:
  - Click the 'Clover' tab to view the Clover code-coverage that relate to this build result (if applicable).
Click the 'JIRA' tab to view any JIRA issues that relate to this build result (if applicable).

RELATED TOPICS

- 3.1 About Builds and Build Results
- 3.2 Viewing a Build Result
- 3.3 Viewing the Code Changes that triggered a Build
- 3.4 Viewing a Build’s Artifacts
- 3.5 Viewing a Build Log
- 3.6 Viewing the Metadata for a Build Result
- 3.7 Viewing the Clover Code-Coverage for a Build Result
- 3.8 Viewing the JIRA Issues for a Build Result

Bamboo Glossary
Return to Bamboo Documentation Home

Return to Bamboo Documentation Home
3.3 Viewing the Code Changes that triggered a Build

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

If a build was triggered by a code change, the code changes will be shown in the build result.

To view the code changes that triggered a particular build result,

1. Go to the build result.
2. Click the 'Changes' tab.
3. A list of updated files will be shown. Click the filename to view the changes; or, click the version number to view the entire file; or, click the 'difs' links to view the differences between the current and previous version of each file.

Screenshot: 'Code Changes'

(Note that links to individual source-code files will only be available if your Bamboo administrator has specified a 'Web Repository URL' in the build's plan. For details please see the Bamboo Administrator's Guide.)

A note about build triggering

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

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

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

RELATED TOPICS

- 3.1 About Builds and Build Results
- 3.2 Viewing a Build Result
- 3.3 Viewing the Code Changes that triggered a Build
- 3.4 Viewing a Build's Artifacts
- 3.5 Viewing a Build Log
- 3.6 Viewing the Metadata for a Build Result
- 3.7 Viewing the Clover Code-Coverage for a Build Result
- 3.8 Viewing the JIRA Issues for a Build Result

Bamboo Glossary

Return to Bamboo Documentation Home
3.4 Viewing a Build's Artifacts

An artifact is something created by a build. There are two types of artifacts:

- User-defined artifacts (e.g. JAR files) are specified in the build’s plan by a Bamboo administrator.
- Auto-generated artifacts are created automatically by Bamboo.

To view a build's artifacts,

1. Go to the build result.
2. Click the 'Artifacts' tab.

Screenshot: 'Build Artifacts'

Build Result CONF-DIST-29

User-Defined Artifacts
This build has the following artifacts:
- standalone
- war
- source

Remove all artifacts

Auto-Generated Artifacts
The following artifacts have been generated by Bamboo:
- Build Log

RELATED TOPICS

- 3.1 About Builds and Build Results
- 3.2 Viewing a Build Result
- 3.3 Viewing the Code Changes that triggered a Build
- 3.4 Viewing a Build's Artifacts
- 3.5 Viewing a Build Log
- 3.6 Viewing the Metadata for a Build Result
- 3.7 Viewing the Clover Code-Coverage for a Build Result
- 3.8 Viewing the JIRA Issues for a Build Result

Bamboo Glossary
Return to Bamboo Documentation Home
3.5 Viewing a Build Log

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

Every build has a build log. A build log is a permanent record of all the output generated by compiling the plan’s source-code and executing the tests.

To view a build log,

1. Go to the build result.
2. Click the 'Logs' tab.

Screenshot: 'Build Log'

```
<table>
<thead>
<tr>
<th>Build</th>
<th>BUCKET-MAIN-74</th>
<th>Status</th>
<th>Logs</th>
<th>Artifacts</th>
<th>Changes</th>
<th>Connects</th>
<th>Project</th>
<th>Plan</th>
<th>Source Build</th>
<th>Build</th>
<th>Plan</th>
<th>Source Build</th>
</tr>
</thead>
</table>
```

RELATED TOPICS

- 3.1 About Builds and Build Results
- 3.2 Viewing a Build Result
- 3.3 Viewing the Code Changes that triggered a Build
- 3.4 Viewing a Build’s Artifacts
- 3.5 Viewing a Build Log
- 3.6 Viewing the Metadata for a Build Result
- 3.7 Viewing the Clover Code-Coverage for a Build Result
- 3.8 Viewing the JIRA Issues for a Build Result

Bamboo Glossary

Return to Bamboo Documentation Home
3.6 Viewing the Metadata for a Build Result

If your source-code repository provides metadata for your build results, Bamboo will display it.

To view the metadata for a build result:

1. Go to the plan.
2. Click the 'Completed Builds' tab, then click the Build Number in the list.
3. This will display the Build Result Summary. Click the 'Metadata' tab.

Screenshot: Metadata for a Build Result

This build has the following metadata. These are property-value pairs describing the build. You can specify your own metadata in the build process via plugins.

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>buildKey</td>
<td>CORE-MAIN</td>
</tr>
<tr>
<td>buildNumber</td>
<td>55</td>
</tr>
<tr>
<td>custom.exeversion</td>
<td>2.01.75</td>
</tr>
</tbody>
</table>

RELATED TOPICS

- 3.1 About Builds and Build Results
- 3.2 Viewing a Build Result
- 3.3 Viewing the Code Changes that triggered a Build
- 3.4 Viewing a Build's Artifacts
- 3.5 Viewing a Build Log
- 3.6 Viewing the Metadata for a Build Result
- 3.7 Viewing the Clover Code-Coverage for a Build Result
- 3.8 Viewing the JIRA Issues for a Build Result

Bamboo Glossary
Return to Bamboo Documentation Home
3.7 Viewing the Clover Code-Coverage for a Build Result

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

If your organisation uses the Atlassian Clover code-coverage tool, Bamboo can record code-coverage details (i.e. the percentage of code covered by tests) for each build result.

This is only available if the build's plan specifies a Clover directory (for details please refer to the Bamboo Administrator's Guide). Also note that the Clover analysis will only be recorded for successful builds.

To view Clover code-coverage for a build result:

1. Go to the plan.
2. Click the 'Completed Builds' tab, then click the Build Number in the list.
3. This will display the Build Result Summary. Click the 'Clover' tab.

Bamboo also provides data on code-coverage trends for a plan over a period of time. For details see:

- 'Clover Code Coverage per Plan' Report.

RELATED TOPICS

- 3.1 About Builds and Build Results
- 3.2 Viewing a Build Result
- 3.3 Viewing the Code Changes that triggered a Build
- 3.4 Viewing a Build's Artifacts
- 3.5 Viewing a Build Log
- 3.6 Viewing the Metadata for a Build Result
- 3.7 Viewing the Clover Code-Coverage for a Build Result
- 3.8 Viewing the JIRA Issues for a Build Result

Bamboo Glossary
Return to Bamboo Documentation Home
3.8 Viewing the JIRA Issues for a Build Result

If your organisation uses the JIRA issue-tracker, Bamboo can automatically create links to any issues mentioned in code check-in comments, as well as displaying a summary in the build result (see below). This provides an easy way to jump to relevant issue(s) to see details about what the code is intended to achieve.

Links to JIRA issues are indicated by a small green arrow to the right of the issue key.

⚠️ This will only be available if Bamboo-JIRA communication has been enabled by your Bamboo administrator.

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.

Note that, when typing your check-in comment, the JIRA issue key(s) must be in upper case (e.g. "ABC-123", not "abc-123").

Screenshot: JIRA Issues for a Build Result

To view all build results related to a JIRA issue:

1. Click the 'Show related builds' link in the above screenshot.

RELATED TOPICS

- 3.1 About Builds and Build Results
- 3.2 Viewing a Build Result
- 3.3 Viewing the Code Changes that triggered a Build
- 3.4 Viewing a Build's Artifacts
- 3.5 Viewing a Build Log
- 3.6 Viewing the Metadata for a Build Result
- 3.7 Viewing the Clover Code-Coverage for a Build Result
- 3.8 Viewing the JIRA Issues for a Build Result

Bamboo Glossary

Return to Bamboo Documentation Home
04. Working with Tests

4. Working with Tests

- 4.1 Viewing Test Results for a Build
- 4.2 Viewing a Test's History
- 4.3 Viewing Test Statistics for a Plan
4.1 Viewing Test Results for a Build

Bamboo provides a convenient summary of all the tests that were run when a particular build was executed — as well as full details of any errors. This is useful when you are investigating what caused a build to fail.

To view the tests for a particular build:

1. Go to the build result.
2. Click the ‘Tests’ tab.
3. Any failed tests will be listed first. Scroll down to see a list of successful tests.

For more meaningful display of test names within Bamboo, the word ‘test’ is stripped out of test case names if it occurs at the beginning, and capitals and underscores are treated as word separators.

Screenshot 1: Failed Tests for a Build

<table>
<thead>
<tr>
<th>Test Name</th>
<th>Failing Tests</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>com.atlassian.jira.service.handler.TestVcsLogHandler</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
To see a particular test’s results for other builds, click the test name.

### RELATED TOPICS

- [4.1 Viewing Test Results for a Build](#)
- [4.2 Viewing a Test’s History](#)
- [4.3 Viewing Test Statistics for a Plan](#)

**Bamboo Glossary**

**Return to Bamboo Documentation Home**
4.2 Viewing a Test's History

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

A test's history shows you:

- The occasions when the test has failed. This can be useful when investigating what code changes were related to a failed test (see below).
- The test's average duration (running time), and whether the duration is increasing or decreasing across builds.

To view a test's history,

1. Go to a plan or a build result.
2. Click the 'Tests' tab.
3. Click the name of the test in which you are interested.
4. (Skip this step if you are looking at a plan.) The test's latest result will be displayed. Click the link 'View test case across builds'.

5. The 'Test History' will be displayed as shown below.

To view the code changes that relate to a failed test,
1. Under 'Recent Failures', click the relevant build result ('47' in the above screenshot).
2. This will display the build result. Click the 'Changes' tab to display the code changes.

RELATED TOPICS

- 4.1 Viewing Test Results for a Build
- 4.2 Viewing a Test's History
- 4.3 Viewing Test Statistics for a Plan

Bamboo Glossary
Return to Bamboo Documentation Home
4.3 Viewing Test Statistics for a Plan

Bamboo provides a summary of test results across all of a plan’s builds. This helps you to:

- Troubleshoot by identifying which tests fail most frequently, and which tests take longest to fix.
- Manage your build duration by identifying the plan's slowest running tests.
- Ensure quality by monitoring the number of tests over time: are your test cases growing with your code base?

To view the test statistics for all of a plan’s builds:

1. Go to the plan.
2. Click the 'Tests' tab.
3. The plan’s 'Top 10 Most Failing Tests' sub-tab will be displayed. Click the other three sub-tabs to view the plan’s 'Top 10 Longest to Fix Tests', 'Top 10 Longest Running Tests', 'Number of Tests' (see screenshots below).

**Screenshot 1: Top 10 Most Failing Tests**

<table>
<thead>
<tr>
<th>Test</th>
<th>Times Failed</th>
<th>Most Recent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Failing</td>
<td>2</td>
<td>37, 26</td>
</tr>
<tr>
<td>Image naked case on mapped</td>
<td>1</td>
<td>40</td>
</tr>
<tr>
<td>Link: link.js</td>
<td>1</td>
<td>40</td>
</tr>
<tr>
<td>Example: example.js</td>
<td>1</td>
<td>40</td>
</tr>
<tr>
<td>Kit member: cancel-case off</td>
<td>1</td>
<td>40</td>
</tr>
<tr>
<td>Kit member cancel case off</td>
<td>1</td>
<td>40</td>
</tr>
<tr>
<td>Link: link.js</td>
<td>1</td>
<td>40</td>
</tr>
<tr>
<td>Kit member cancel case off</td>
<td>1</td>
<td>40</td>
</tr>
<tr>
<td>Testcase triple_f1</td>
<td>1</td>
<td>40</td>
</tr>
<tr>
<td>Link: link.js</td>
<td>1</td>
<td>40</td>
</tr>
</tbody>
</table>

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

**Screenshot 2: Top 10 Longest to Fix Tests**

<table>
<thead>
<tr>
<th>Test</th>
<th>Average Time to Fix</th>
</tr>
</thead>
<tbody>
<tr>
<td>Failing</td>
<td>1 hour 2 minutes</td>
</tr>
<tr>
<td>Image naked case on mapped</td>
<td>19 minutes</td>
</tr>
<tr>
<td>Link: link.js</td>
<td>19 minutes</td>
</tr>
<tr>
<td>Example: example.js</td>
<td>19 minutes</td>
</tr>
<tr>
<td>Kit member: cancel-case off</td>
<td>19 minutes</td>
</tr>
<tr>
<td>Kit member cancel case off</td>
<td>19 minutes</td>
</tr>
<tr>
<td>Link: link.js</td>
<td>19 minutes</td>
</tr>
<tr>
<td>Kit member cancel case off</td>
<td>19 minutes</td>
</tr>
<tr>
<td>Testcase triple_f1</td>
<td>19 minutes</td>
</tr>
<tr>
<td>Link: link.js</td>
<td>19 minutes</td>
</tr>
</tbody>
</table>

**Screenshot 3: Top 10 Longest Running Tests**
### Related Topics

- **4.1 Viewing Test Results for a Build**
- **4.2 Viewing a Test's History**
- **4.3 Viewing Test Statistics for a Plan**

**Bamboo Glossary**

[Return to Bamboo Documentation Home](#)
5. Reporting on Trends

• **5.1 Viewing Build Statistics for a Plan**
• **5.2 Generating Reports across multiple Plans**
  ° 'Build Activity per Plan' Report
  ° 'Build Duration per Plan' Report
  ° 'Clover Code Coverage per Plan' Report
  ° 'Clover Lines of Code per Plan' Report
  ° 'Number of Build Failures per Plan' Report
  ° 'Number of Tests per Plan' Report
  ° 'Percentage of Successful Builds per Plan' Report
  ° 'Time to Fix per Plan' Report
5.1 Viewing Build Statistics for a Plan

To view a plan's build statistics,

- From the Dashboard, locate and click a Plan Name from the list;
- OR:
  - From within a build result, click the Plan Name at the top left of the screen.

The Plan Summary will be displayed as follows:

**Screenshot: 'Plan Summary'

![Plan Summary Screenshot]

- **Latest build** BUCKET-MAIN-74 was successful
- **96%** SuccessfulRuns: 24/26, Average Duration: 58 seconds

**Build Timing & Number of Failures per Build**

**% Successful Build & Avg Duration per Time Period**

- **Note:** Lines are drawn from the results of the last 25 builds. Use the drop down in the right column to change your timeframe.

**Recent Failures**

- Average time to fail a build: 42 minutes
- Average number of builds between failed: 1 build
- The longest time taken to fail a failure is 42 minutes, from failure starting in build 73.
- The greatest number of builds taken of failures is 1, from failure starting in build 73.

<table>
<thead>
<tr>
<th>Failed Build</th>
<th>Fixed In</th>
<th>Time Taken To Fix</th>
</tr>
</thead>
<tbody>
<tr>
<td>73 (Updated by Marissa)</td>
<td>24 (Updated by Marissa)</td>
<td>93 hours</td>
</tr>
</tbody>
</table>

**In the above screenshot:**

The green box indicates that this plan's latest **build** was successful. Note that a red box in this position would indicate that the plan's latest build failed, while a blue box would indicate that a build is currently in progress.

- Click the build number (i.e. 'BUCKET-MAIN-71') to **view the build result**.
• Click the 'Updated by' link to view the code changes that triggered the latest build result.

• Build duration is the total time taken to execute a build plan — that is, the time taken to compile the code and run all of the plan’s tests.

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 Documentation Home
5.2 Generating Reports across multiple Plans

Bamboo provides a report generator that enables you to compare build statistics across one or more plans, using a variety of different metrics.

To report on build statistics per plan,

1. Click the 'Reports' link in the top navigation bar. This will display the 'Report Parameters' screen as shown below.
2. 'Report' — choose from the available reports. Available reports include:
   - 'Build Activity per Plan' Report
   - 'Build Duration per Plan' Report
   - 'Clover Code Coverage per Plan' Report
   - 'Clover Lines of Code per Plan' Report
   - 'Number of Build Failures per Plan' Report
   - 'Number of Tests per Plan' Report
   - 'Percentage of Successful Builds per Plan' Report
   - 'Time to Fix per Plan' Report
   Additionally, your Bamboo administrator may configure custom reports by using plugins. For details please see the Bamboo Administrator's Guide.
3. 'Build plans' — choose the plan(s) on which you want to report. You can use the <Ctrl> key to select multiple plans.
   - Project names are shown in italics, e.g. 'Geronimo SVN'. Plan names are shown in non-italics, e.g. 'Main Build'.
4. 'Group By' — choose whether your report's horizontal axis should show days, months or weeks. You can also specify 'Auto', which varies by report, but will generally default to 'week'.
5. 'Date Filter' — choose from:
   - 'All builds'
   - 'Last 7 days'
   - 'Last 30 days'
   - 'Last 90 days'
   - 'Select Range' — choosing this option will display two boxes in which you will need to specify the 'from' and 'to' dates (dd/MM/yyyy).
6. Click the 'Submit' button to generate your report.

Screenshot: 'Report Parameters--Build Plans'

RELATED TOPICS

- 5.1 Viewing Build Statistics for a Plan
- 5.2 Generating Reports across multiple Plans
  - 'Build Activity per Plan' Report
° 'Build Duration per Plan' Report
° 'Clover Code Coverage per Plan' Report
° 'Clover Lines of Code per Plan' Report
° 'Number of Build Failures per Plan' Report
° 'Number of Tests per Plan' Report
° 'Percentage of Successful Builds per Plan' Report
° 'Time to Fix per Plan' Report

Bamboo Glossary
Return to Bamboo Documentation Home
'Build Activity per Plan' Report

Build activity is the number of builds that occur in a given period of time. You can choose the plan(s) and time period on which you want to report.

Sample Report: 'Build Activity per Plan'

![Build Activity Graph](image)

**RELATED TOPICS**

- [5.1 Viewing Build Statistics for a Plan](#)
- [5.2 Generating Reports across multiple Plans](#)
  - 'Build Activity per Plan' Report
  - 'Build Duration per Plan' Report
  - 'Clover Code Coverage per Plan' Report
  - 'Clover Lines of Code per Plan' Report
  - 'Number of Build Failures per Plan' Report
  - 'Number of Tests per Plan' Report
  - 'Percentage of Successful Builds per Plan' Report
  - 'Time to Fix per Plan' Report

Bamboo Glossary

Return to [Bamboo Documentation Home](#)
'Build Duration per Plan' Report

Build duration is the total time taken to execute a build plan — that is, the time taken to compile the code and run all of the plan's tests.

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

Sample Report: 'Build Duration per Plan'

 RELATED TOPICS

• 5.1 Viewing Build Statistics for a Plan
• 5.2 Generating Reports across multiple Plans
  ◦ 'Build Activity per Plan' Report
  ◦ 'Build Duration per Plan' Report
  ◦ 'Clover Code Coverage per Plan' Report
  ◦ 'Clover Lines of Code per Plan' Report
  ◦ 'Number of Build Failures per Plan' Report
  ◦ 'Number of Tests per Plan' Report
  ◦ 'Percentage of Successful Builds per Plan' Report
  ◦ 'Time to Fix per Plan' Report

Bamboo Glossary
Return to Bamboo Documentation Home
'Clover Code Coverage per Plan' Report

This report will only be available if your administrator has specified 'Clover output will be produced' in the plan's configuration. For details please see the Bamboo Administrator’s Guide.

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

Sample Report: 'Clover Code Coverage per Plan'

Clove Code Coverage

Comparing code coverage gives you an idea of how well the code base is tested. 100% coverage means that all code elements have been covered by your tests.

RELATED TOPICS

- 5.1 Viewing Build Statistics for a Plan
- 5.2 Generating Reports across multiple Plans
  - 'Build Activity per Plan' Report
  - 'Build Duration per Plan' Report
  - 'Clover Code Coverage per Plan' Report
  - 'Clover Lines of Code per Plan' Report
  - 'Number of Build Failures per Plan' Report
  - 'Number of Tests per Plan' Report
  - 'Percentage of Successful Builds per Plan' Report
  - 'Time to Fix per Plan' Report

Bamboo Glossary

Return to Bamboo Documentation Home
'Clover Lines of Code per Plan' Report

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

This report will only be available if your administrator has specified 'Clover output will be produced' in the plan's configuration. For details please see the Bamboo Administrator's Guide.

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

Sample Report: 'Clover Lines of Code per Plan'

'Clover Lines of Code'

Provides an indication of the size of the code base for the build.

RELATED TOPICS

- 5.1 Viewing Build Statistics for a Plan
- 5.2 Generating Reports across multiple Plans
  - 'Build Activity per Plan' Report
  - 'Build Duration per Plan' Report
  - 'Clover Code Coverage per Plan' Report
  - 'Clover Lines of Code per Plan' Report
  - 'Number of Build Failures per Plan' Report
  - 'Number of Tests per Plan' Report
  - 'Percentage of Successful Builds per Plan' Report
  - 'Time to Fix per Plan' Report

Bamboo Glossary

Return to Bamboo Documentation Home
'Number of Build Failures per Plan' Report

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

Sample Report: 'Number of Build Failures per Plan'

**Number of Build Failures**

How many builds are being broken? A high value indicates a reliably rebuildable build that needs to be broken often.

**RELATED TOPICS**

- **5.1 Viewing Build Statistics for a Plan**
- **5.2 Generating Reports across multiple Plans**
  - 'Build Activity per Plan' Report
  - 'Build Duration per Plan' Report
  - 'Clover Code Coverage per Plan' Report
  - 'Clover Lines of Code per Plan' Report
  - 'Number of Build Failures per Plan' Report
  - 'Number of Tests per Plan' Report
  - 'Percentage of Successful Builds per Plan' Report
  - 'Time to Fix per Plan' Report

**Bamboo Glossary**

Return to Bamboo Documentation Home
'Number of Tests per Plan' Report

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

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

Sample Report: 'Number of Tests per Plan'

![Number of Tests](image)

### RELATED TOPICS

- [5.1 Viewing Build Statistics for a Plan](#)
- [5.2 Generating Reports across multiple Plans](#)
  - 'Build Activity per Plan' Report
  - 'Build Duration per Plan' Report
  - 'Clover Code Coverage per Plan' Report
  - 'Clover Lines of Code per Plan' Report
  - 'Number of Build Failures per Plan' Report
  - 'Number of Tests per Plan' Report
  - 'Percentage of Successful Builds per Plan' Report
  - 'Time to Fix per Plan' Report

[Bamboo Glossary](#)

Return to [Bamboo Documentation Home](#)
'Percentage of Successful Builds per Plan' Report

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

Sample Report: 'Percentage of Successful Builds per Plan'

<table>
<thead>
<tr>
<th>Percentage of Successful Builds</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computing success percentages gives you an idea of how stable a build is compared to one another. 100% means your build is always successful, 0% means something is seriously wrong.</td>
</tr>
</tbody>
</table>

Chart showing the percentage of successful builds over time.

RELATED TOPICS

- 5.1 Viewing Build Statistics for a Plan
- 5.2 Generating Reports across multiple Plans
  - 'Build Activity per Plan' Report
  - 'Build Duration per Plan' Report
  - 'Clover Code Coverage per Plan' Report
  - 'Clover Lines of Code per Plan' Report
  - 'Number of Build Failures per Plan' Report
  - 'Number of Tests per Plan' Report
  - 'Percentage of Successful Builds per Plan' Report
  - 'Time to Fix per Plan' Report

Bamboo Glossary

Return to Bamboo Documentation Home
'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'

![Graph showing time to fix per plan]

RELATED TOPICS

- 5.1 Viewing Build Statistics for a Plan
- 5.2 Generating Reports across multiple Plans
  - 'Build Activity per Plan' Report
  - 'Build Duration per Plan' Report
  - 'Clover Code Coverage per Plan' Report
  - 'Clover Lines of Code per Plan' Report
  - 'Number of Build Failures per Plan' Report
  - 'Number of Tests per Plan' Report
  - 'Percentage of Successful Builds per Plan' Report
  - 'Time to Fix per Plan' Report

Bamboo Glossary

Return to Bamboo Documentation Home
06. Reporting on Author Trends

This page last changed on May 14, 2007 by rosie@atlassian.com.

6. Reporting on Author Trends

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

- 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 contributes to a build by checking-in code to a repository that is associated with a Bamboo plan. An author need not be a Bamboo user. To view a summary of all authors' statistics,

1. Click the 'Authors' link in the top navigation bar.
2. This will display the following screen, where you can click any column-header to sort in ascending order (or click twice to sort in descending order).

Screenshot: 'Authors Summary-sorted by 'Fixed' (descending)'

<table>
<thead>
<tr>
<th>Name</th>
<th>Triggered</th>
<th>Failed</th>
<th>% Failed</th>
<th>Broken</th>
<th>Fixed</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>New Builder</td>
<td>962</td>
<td>275</td>
<td>20%</td>
<td>124</td>
<td>128</td>
<td>5</td>
</tr>
<tr>
<td>Max Smith</td>
<td>963</td>
<td>99</td>
<td>11%</td>
<td>47</td>
<td>94</td>
<td>3</td>
</tr>
<tr>
<td>Tom Brown</td>
<td>411</td>
<td>89</td>
<td>22%</td>
<td>50</td>
<td>40</td>
<td>10</td>
</tr>
<tr>
<td>SallyLee</td>
<td>452</td>
<td>102</td>
<td>24%</td>
<td>44</td>
<td>22</td>
<td>6</td>
</tr>
</tbody>
</table>

Handy Hint
You can click any author's name to see their recent build results.

RELATED TOPICS

- 6.1 Viewing Build Statistics for all Authors
- 6.2 Viewing Build Results for an Author
- 6.3 Generating Reports on selected Authors
  - 'Build Activity per Author' Report
  - 'Number of Build Failures per Author' Report
  - 'Number of Builds Broken per Author' Report
  - 'Number of Builds Fixed per Author' Report
  - 'Percentage of Successful Builds per Author' Report

Bamboo Glossary
Return to Bamboo Documentation Home
6.2 Viewing Build Results for an Author

This author is any person who contributes to a build by checking-in code to a repository that is associated with a Bamboo plan. An author need not be a Bamboo user.

To view an author's build results,

1. Click the 'Authors' link in the top navigation bar.
2. This will display the 'Authors Summary' screen. Click the relevant author's name.
3. This will display the author's 'User Details' (email address, etc) — see screenshot below. Click through the following tabs to view recent build results:
   • 'Builds Summary' — a statistical summary of all the author's builds.
   • 'Last 10 Builds' — a list of the last 10 builds that were triggered by this author.
   • 'Last 10 Broken' — a list of the last 10 builds that were triggered by this author, where the build failed and the previous build for the same plan was successful.
   • 'Last 10 Fixed' — a list of the last 10 builds that were triggered by this author, where the build was successful and the previous build for the same plan failed.

If your Bamboo User Profile has not yet been associated with your Author Name, there will be no 'User Details' tab.

Screenshot 1: 'User Details' tab

<table>
<thead>
<tr>
<th>Personal Details</th>
<th>Build Summary</th>
<th>Last 10 Builds</th>
<th>Last 10 Broken</th>
<th>Last 10 Fixed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Username:</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Full Name:</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Email:</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Joiner Address:</td>
<td><a href="mailto:email@company.com">email@company.com</a></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Group:</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Source Repository Alias:</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Email:</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Screenshot 2: 'Builds Summary' tab

<table>
<thead>
<tr>
<th>Builds Triggered by Author</th>
<th>Build Summary</th>
<th>Last 10 Builds</th>
<th>Last 10 Broken</th>
<th>Last 10 Fixed</th>
</tr>
</thead>
<tbody>
<tr>
<td>All builds triggered</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Failed Builds</td>
<td>275 (93%)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Successful Builds</td>
<td>7 (2%)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Screenshot 3: 'Last 10 Builds' tab

<table>
<thead>
<tr>
<th>Developers and Fixes</th>
<th>Broken by Author</th>
<th>Fixed By Author</th>
<th>Difference of Fixed and Broken</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>123 (96% of all builds triggered)</td>
<td>127 (96% of all builds triggered)</td>
<td>4 (96% of all builds triggered)</td>
</tr>
</tbody>
</table>
### Screenshot 4: 'Last 10 Broken' tab

<table>
<thead>
<tr>
<th>Build Status</th>
<th>Build Time</th>
<th>Build Details</th>
<th>Build Summary</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Broken</td>
<td>1 week ago</td>
<td>Added some behavior to the test data</td>
<td>[ details ]</td>
<td>Test Results: 7/10 passed</td>
</tr>
<tr>
<td>Broken</td>
<td>1 week ago</td>
<td>CVS library update</td>
<td>[ details ]</td>
<td>Test Results: 7/10 passed</td>
</tr>
<tr>
<td>Broken</td>
<td>2 weeks ago</td>
<td>Made sure the test data was correct when we checked [ details ]</td>
<td>[ details ]</td>
<td>Test Results: 7/10 passed</td>
</tr>
<tr>
<td>Broken</td>
<td>2 weeks ago</td>
<td>Made sure that the test data worked on windows [ details ]</td>
<td>[ details ]</td>
<td>Test Results: 7/10 passed</td>
</tr>
<tr>
<td>Broken</td>
<td>2 weeks ago</td>
<td>Added debugging to the build process</td>
<td>[ details ]</td>
<td>Test Results: 7/10 passed</td>
</tr>
<tr>
<td>Broken</td>
<td>2 weeks ago</td>
<td>Made sure that the build process worked on Windows [ details ]</td>
<td>[ details ]</td>
<td>Test Results: 7/10 passed</td>
</tr>
<tr>
<td>Broken</td>
<td>1 month ago</td>
<td>Added some functionality to the build process</td>
<td>[ details ]</td>
<td>Test Results: 2/10 failed</td>
</tr>
<tr>
<td>Broken</td>
<td>1 month ago</td>
<td>Simplified the build process quite a bit</td>
<td>[ details ]</td>
<td>Test Results: 2/10 failed</td>
</tr>
<tr>
<td>Broken</td>
<td>1 month ago</td>
<td>Added some behavior to the test data</td>
<td>[ details ]</td>
<td>Test Results: 2/10 failed</td>
</tr>
<tr>
<td>Broken</td>
<td>1 month ago</td>
<td>Fixed some issues in the build process</td>
<td>[ details ]</td>
<td>Test Results: 2/10 failed</td>
</tr>
<tr>
<td>Broken</td>
<td>1 month ago</td>
<td>Fixed some issues in the build process</td>
<td>[ details ]</td>
<td>Test Results: 2/10 failed</td>
</tr>
<tr>
<td>Broken</td>
<td>1 month ago</td>
<td>Fixed some issues in the build process</td>
<td>[ details ]</td>
<td>Test Results: 2/10 failed</td>
</tr>
<tr>
<td>Broken</td>
<td>1 month ago</td>
<td>Fixed some issues in the build process</td>
<td>[ details ]</td>
<td>Test Results: 2/10 failed</td>
</tr>
<tr>
<td>Broken</td>
<td>1 month ago</td>
<td>Fixed some issues in the build process</td>
<td>[ details ]</td>
<td>Test Results: 2/10 failed</td>
</tr>
</tbody>
</table>

### Screenshot 5: 'Last 10 Fixed' tab

<table>
<thead>
<tr>
<th>Build Status</th>
<th>Build Time</th>
<th>Build Details</th>
<th>Build Summary</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fixed</td>
<td>2 weeks ago</td>
<td>Fixed some issues in the test data</td>
<td>[ details ]</td>
<td>Test Results: 10/11 passed</td>
</tr>
<tr>
<td>Fixed</td>
<td>2 weeks ago</td>
<td>Fixed some issues in the test data</td>
<td>[ details ]</td>
<td>Test Results: 10/11 passed</td>
</tr>
<tr>
<td>Fixed</td>
<td>2 weeks ago</td>
<td>Fixed some issues in the test data</td>
<td>[ details ]</td>
<td>Test Results: 10/11 passed</td>
</tr>
<tr>
<td>Fixed</td>
<td>1 month ago</td>
<td>Fixed some issues in the test data</td>
<td>[ details ]</td>
<td>Test Results: 10/11 passed</td>
</tr>
<tr>
<td>Fixed</td>
<td>1 month ago</td>
<td>Fixed some issues in the test data</td>
<td>[ details ]</td>
<td>Test Results: 10/11 passed</td>
</tr>
<tr>
<td>Fixed</td>
<td>1 month ago</td>
<td>Fixed some issues in the test data</td>
<td>[ details ]</td>
<td>Test Results: 10/11 passed</td>
</tr>
<tr>
<td>Fixed</td>
<td>1 month ago</td>
<td>Fixed some issues in the test data</td>
<td>[ details ]</td>
<td>Test Results: 10/11 passed</td>
</tr>
<tr>
<td>Fixed</td>
<td>1 month ago</td>
<td>Fixed some issues in the test data</td>
<td>[ details ]</td>
<td>Test Results: 10/11 passed</td>
</tr>
<tr>
<td>Fixed</td>
<td>1 month ago</td>
<td>Fixed some issues in the test data</td>
<td>[ details ]</td>
<td>Test Results: 10/11 passed</td>
</tr>
<tr>
<td>Fixed</td>
<td>1 month ago</td>
<td>Fixed some issues in the test data</td>
<td>[ details ]</td>
<td>Test Results: 10/11 passed</td>
</tr>
</tbody>
</table>

### RELATED TOPICS

- 6.1 Viewing Build Statistics for all Authors
- 6.2 Viewing Build Results for an Author
- 6.3 Generating Reports on selected Authors
  - "Build Activity per Author" Report
  - "Number of Build Failures per Author" Report
° 'Number of Builds Broken per Author' Report
° 'Number of Builds Fixed per Author' Report
° 'Percentage of Successful Builds per Author' Report

Bamboo Glossary
Return to Bamboo Documentation Home
6.3 Generating Reports on selected Authors

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

To generate a report on selected authors,

1. Click the 'Authors' link in the top navigation bar.
2. Click the 'Statistics' tab. This will display the 'Report Parameters' screen as shown below.
3. 'Report' — choose from the available reports. Available reports include:
   - 'Build Activity per Author' Report
   - 'Number of Build Failures per Author' Report
   - 'Number of Builds Broken per Author' Report
   - 'Number of Builds Fixed per Author' Report
   - 'Percentage of Successful Builds per Author' Report

   Additionally, your Bamboo administrator may configure custom reports by using plugins. For details please see the Bamboo Administrator's Guide.
4. 'Authors' — choose the author(s) on whom you want to report. You can use the <Ctrl> key to select multiple author.
5. 'Group By' — choose whether your report's horizontal axis should show days, months or weeks. You can also specify 'Auto', which varies by report, but will generally default to 'month'.
6. Click the 'Submit' button to generate your report.

Screenshot: 'Report Parameters--Authors'

RELATED TOPICS

- 6.1 Viewing Build Statistics for all Authors
- 6.2 Viewing Build Results for an Author
- 6.3 Generating Reports on selected Authors
  - 'Build Activity per Author' Report
  - 'Number of Build Failures per Author' Report
  - 'Number of Builds Broken per Author' Report
  - 'Number of Builds Fixed per Author' Report
  - 'Percentage of Successful Builds per Author' Report
'Build Activity per Author' Report

Build activity is the number of builds that occur in a given period of time. You can select the author(s) on whom you want to report.

Sample Report: 'Build Activity per Author'

RELATED TOPICS

• 6.1 Viewing Build Statistics for all Authors
• 6.2 Viewing Build Results for an Author
• 6.3 Generating Reports on selected Authors
  • 'Build Activity per Author' Report
  • 'Number of Build Failures per Author' Report
  • 'Number of Builds Broken per Author' Report
  • 'Number of Builds Fixed per Author' Report
  • 'Percentage of Successful Builds per Author' Report

Bamboo Glossary
Return to Bamboo Documentation Home
'Number of Build Failures per Author' Report

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

Sample Report: 'Number of Build Failures per Author'

![Graph of Build Failures over Time]

RELATED TOPICS

- 6.1 Viewing Build Statistics for all Authors
- 6.2 Viewing Build Results for an Author
- 6.3 Generating Reports on selected Authors
  - 'Build Activity per Author' Report
  - 'Number of Build Failures per Author' Report
  - 'Number of Builds Broken per Author' Report
  - 'Number of Builds Fixed per Author' Report
  - 'Percentage of Successful Builds per Author' Report

Bamboo Glossary
Return to Bamboo Documentation Home
'Number of Builds Broken per Author' Report

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

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

Sample Report: 'Number of Builds Broken per Author'

![Graph showing number of builds broken per author]

RELATED TOPICS

- 6.1 Viewing Build Statistics for all Authors
- 6.2 Viewing Build Results for an Author
- 6.3 Generating Reports on selected Authors
  - 'Build Activity per Author' Report
  - 'Number of Build Failures per Author' Report
  - 'Number of Builds Broken per Author' Report
  - 'Number of Builds Fixed per Author' Report
  - 'Percentage of Successful Builds per Author' Report

Bamboo Glossary
Return to Bamboo Documentation Home
'Number of Builds Fixed per Author' Report

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

Sample Report: 'Number of Builds Fixed per Author'

 RELATED TOPICS

- 6.1 Viewing Build Statistics for all Authors
- 6.2 Viewing Build Results for an Author
- 6.3 Generating Reports on selected Authors
  - 'Build Activity per Author' Report
  - 'Number of Build Failures per Author' Report
  - 'Number of Builds Broken per Author' Report
  - 'Number of Builds Fixed per Author' Report
  - 'Percentage of Successful Builds per Author' Report

Bamboo Glossary
Return to Bamboo Documentation Home
'Percentage of Successful Builds per Author' Report

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

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

Sample Report: 'Percentage of Successful Builds per Author'

![Percentage of Successful Builds](image)

RELATED TOPICS

- 6.1 Viewing Build Statistics for all Authors
- 6.2 Viewing Build Results for an Author
- 6.3 Generating Reports on selected Authors
  - 'Build Activity per Author' Report
  - 'Number of Build Failures per Author' Report
  - 'Number of Builds Broken per Author' Report
  - 'Number of Builds Fixed per Author' Report
  - 'Percentage of Successful Builds per Author' Report

Bamboo Glossary

Return to Bamboo Documentation Home
07. Working with Comments

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

7. Working with Comments

- 7.1 About Comments
- 7.2 Commenting about a Build Result
- 7.3 Viewing Comments about a Build Result
- 7.4 Viewing Code Check-in Comments
7.1 About Comments

Comments are a useful way to record and share information about builds. There are two types of comments in Bamboo:

- Comments you make when you commit code — these comments are automatically copied into Bamboo from your source-code repository. See Viewing Code Check-in Comments.
- Comments you make about a build result — these are comments that you make ad-hoc about a particular build result. See Commenting about a Build Result and Viewing Comments about a Build Result.

RELATED TOPICS

- 7.1 About Comments
- 7.2 Commenting about a Build Result
- 7.3 Viewing Comments about a Build Result
- 7.4 Viewing Code Check-in Comments

Bamboo Glossary
Return to Bamboo Documentation Home
7.2 Commenting about a Build Result

Bamboo allows you to record comments about a build result. This is a convenient way to record relevant information for future reference, and to collaborate with colleagues.

To comment on a build result,

1. From within the 'Build Result' screen, click the 'Comments' tab. A list of existing comments about this build result will be displayed.
2. Type your comment into the 'Add Comment' box, then click the 'Save' button.

Screenshot: Build Result - Comments

You must login to Bamboo before you can comment on a build result.

RELATED TOPICS

- 7.1 About Comments
- 7.2 Commenting about a Build Result
- 7.3 Viewing Comments about a Build Result
- 7.4 Viewing Code Check-in Comments

Bamboo Glossary
Return to Bamboo Documentation Home
7.3 Viewing Comments about a Build Result

To view comments about a particular build result,

1. From within the 'Build Result' screen, click the 'Comments' tab. A list of all comments about this build result will be displayed, including author and timestamp:

   ![Example Comments](image)

To view comments about all build results for a particular plan,

1. From the Dashboard, click the plan you are interested in.
2. Click the plan’s 'Completed Builds' tab.
3. This will display a list of the plan’s build results. The icon indicates that there are one or more comments about a particular build result. Hold your mouse over the icon to see the comment(s), e.g.:

   ![Completed Builds](image)

RELATED TOPICS

- 7.1 About Comments
- 7.2 Commenting about a Build Result
- 7.3 Viewing Comments about a Build Result
- 7.4 Viewing Code Check-in Comments

Bamboo Glossary
Return to Bamboo Documentation Home
7.4 Viewing Code Check-in Comments

If a build was triggered by a code change, the commit comment (or check-in comment) will be shown in the build result.

To view the code check-in comments for a particular build result,

1. Go to the build result.
2. The build's commit comment will be shown to the right of the screen, under the heading 'Code Changes'.

A note about build triggering
There are a variety of ways in which a build can be triggered for a plan:

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

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

RELATED TOPICS

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

Bamboo Glossary
Return to Bamboo Documentation Home
8. Working with Labels

- 8.1 About Labels
- 8.2 Labelling a Build Result
- 8.3 Removing a Label from a Build Result
- 8.4 Viewing Labelled Build Results
- 8.5 Viewing Popular Labels
8.1 About Labels

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

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

RELATED TOPICS

- 8.1 About Labels
- 8.2 Labelling a Build Result
- 8.3 Removing a Label from a Build Result
- 8.4 Viewing Labelled Build Results
- 8.5 Viewing Popular Labels

Bamboo Glossary
Return to Bamboo Documentation Home
8.2 Labelling a Build Result

With Bamboo, you can label your build results in whatever way works best for your team. Labels are not restricted to a particular plan, so you can apply the same label to build results from different plans.

For example, it might not be practical for your QA team to review every build, and you need to know which builds they have reviewed. By using labels such as "qa_passed" and "qa_failed", Bamboo allows them to simply indicate which builds have passed and failed QA.

To label a build result,

1. Go to the build result.
2. Locate the 'Labels' link at the top of the screen:
   ![Labels](image)
   (Hint: To view a list of existing labels, click the 'Labels' link.)
3. Click the 'Add' link. This will display the following:
   ![Add_labels](image)
   The 'Add' link will only be visible if you have logged in to Bamboo.
4. Type the relevant label (or multiple labels, separated by commas). Note that the label will be saved in lowercase characters.
5. Click the 'Done' button.

You can also label a build result via Instant Messaging (IM).

RELATED TOPICS

- 8.1 About Labels
- 8.2 Labelling a Build Result
- 8.3 Removing a Label from a Build Result
- 8.4 Viewing Labelled Build Results
- 8.5 Viewing Popular Labels

Bamboo Glossary
Return to Bamboo Documentation Home
8.3 Removing a Label from a Build Result

To remove a label from a build result,

1. Go to the build result.
2. Locate the 'Labels' link at the top of the screen:
3. Click the 'Edit' link. This will display the following:
4. Click the small red 'x' at the right of the label you want to remove.
5. Click the 'Done' button.

You must login to Bamboo before you can remove a label from a build result.

RELATED TOPICS

- 8.1 About Labels
- 8.2 Labelling a Build Result
- 8.3 Removing a Label from a Build Result
- 8.4 Viewing Labelled Build Results
- 8.5 Viewing Popular Labels

Bamboo Glossary
Return to Bamboo Documentation Home
8.4 Viewing Labelled Build Results

To view all build results which have a particular label,

1. Go to any build result.
2. Click the 'Labels' link at the top of the screen (above the 'Summary' tab).
3. Click the link 'See also labels in all projects'.
4. This will display a list of all labels that are used in Bamboo. Click the label of interest.
5. This will display a list of all build results which have that label.

RELATED TOPICS

- 8.1 About Labels
- 8.2 Labelling a Build Result
- 8.3 Removing a Label from a Build Result
- 8.4 Viewing Labelled Build Results
- 8.5 Viewing Popular Labels

Bamboo Glossary
Return to Bamboo Documentation Home
8.5 Viewing Popular Labels

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

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'

Handy Hint
You can click any label to see a list of all build results which have that label.

RELATED TOPICS

- 8.1 About Labels
- 8.2 Labelling a Build Result
- 8.3 Removing a Label from a Build Result
- 8.4 Viewing Labelled Build Results
- 8.5 Viewing Popular Labels

Bamboo Glossary
Return to Bamboo Documentation Home
09. Subscribing to RSS Feeds

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

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:
3. Right-click the 'all builds' link and copy its URL.
4. Paste the URL into your RSS reader.

RELATED TOPICS

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

Bamboo Glossary
Return to Bamboo Documentation Home
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:
3. Right-click the 'failed builds' link and copy its URL.
4. Paste the URL into your RSS reader.
# 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:
   ![RSS Icon](Feed_for_all_build_results.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](#)

[Return to Bamboo Glossary]

[Return to Bamboo Documentation Home]
9.4 Subscribing to an RSS Feed for Failed Builds for a Particular Plan

To subscribe to an RSS feed for failed builds for a particular plan,

1. Go to the plan.
2. Locate the RSS icon at the bottom of the screen:
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 Failed Builds 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
9.5 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:
7. Right-click the 'Feed for builds labelled' link and copy its URL.
8. Paste the URL into your RSS reader.

What is a label?

RELATED TOPICS

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

Bamboo Glossary
Return to Bamboo Documentation Home
10. Working with Instant Messenger (IM) Notifications

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

10. Working with Instant Messenger (IM) notifications

- 10.1 About Instant Messenger (IM) Notifications
- 10.2 Labelling a Build Result via IM
- 10.3 Commenting about a Build Result via IM
**10.1 About Instant Messenger (IM) Notifications**

Bamboo can send you notifications about **build results** for a particular plan(s). Each plan's recipients are specified by a Bamboo administrator, but you can choose whether you would like to receive your Bamboo notifications via email and/or instant messenger (IM). See 11.2 Changing your Notification Preferences.

As well as receiving IM notifications, you can interact with Bamboo via IM. By responding to an IM notification, you can:

- Label a build result via IM
- Comment about a build result via IM

**RELATED TOPICS**

- 10.1 About Instant Messenger (IM) Notifications
- 10.2 Labelling a Build Result via IM
- 10.3 Commenting about a Build Result via IM

Bamboo Glossary

Return to [Bamboo Documentation Home](#)
10.2 Labelling a Build Result via IM

You can respond to a Bamboo IM notification message with commands to label or comment on a build result.

To label a build result via Instant Messaging (IM),

In your Instant Messenger, type your comment in the following format:

```
label [build key] <labels>
```

Screenshot: Interacting with Bamboo via IM

What is a label?

RELATED TOPICS

- 10.1 About Instant Messenger (IM) Notifications
- 10.2 Labelling a Build Result via IM
- 10.3 Commenting about a Build Result via IM

Bamboo Glossary
Return to Bamboo Documentation Home
10.3 Commenting about a Build Result via IM

You can respond to a Bamboo IM notification message with commands to label or comment on a build result.

To comment on a build result via Instant Messaging (IM),

In your Instant Messenger, type your comment in the following format:

comment [build key] <comment message>

Screenshot: Interacting with Bamboo via IM

RELATED TOPICS

- 10.1 About Instant Messenger (IM) Notifications
- 10.2 Labelling a Build Result via IM
- 10.3 Commenting about a Build Result via IM

Bamboo Glossary
Return to Bamboo Documentation Home
11. Editing your User Profile

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

To change your Bamboo password,

1. Click the 'Profile' link in the top right corner of the screen. This will display the 'User Profile' screen.
2. Click the 'Change Password' link.
3. Type your old and new passwords.
4. Click the 'Change Password' button.

RELATED TOPICS

- 11.1 Changing your Password
- 11.2 Changing your Notification Preferences
- 11.3 Associating your Author Name with your User Profile

Bamboo Glossary
Return to Bamboo Documentation Home
11.2 Changing your Notification Preferences

Bamboo can send you notifications about build results. You can choose whether to receive your notifications via email or Instant Messaging (IM), or both. You can also choose not to receive notifications at all.

To change your notification preferences,

1. Click the 'Profile' link in the top right corner of the screen. This will display the 'User Profile' screen.
2. Click the 'Edit Profile' link. The 'Edit User Profile' screen will be displayed, as shown below.
3. Under 'Set Your Notification Preference', choose how you would like to receive your notifications about build results:
   - 'Do not send notifications'
   - 'Send instant message'*
   - 'Send email'
   - 'Send email and instant message'*
4. Click the 'Save' button.

* If you select this option, you need to specify your IM address in the 'Jabber Address' field.

Screenshot: User Profile

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

RELATED TOPICS

- 11.1 Changing your Password
- 11.2 Changing your Notification Preferences
- 11.3 Associating your Author Name with your User Profile

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

An author is any person who contributes to a build by checking-in code to a repository that is associated with a Bamboo plan. An author need not be a Bamboo user. Your Author Name is your login name for the source-code repository.

If your Bamboo User Profile has not yet been associated with your Author Name, then:

- your 'My Bamboo' screen will not contain any data about your recent builds.
- your 'Author' information will not include a 'User Details' tab.

To associate your Author Name with your User Profile,

1. Click the 'Profile' link in the top right corner of the screen. This will display the 'User Profile' screen.
2. Click the 'Edit Profile' link.
3. In the 'Source Repository Alias' field, select your Author Name from the list. If your Author Name does not appear in the list, select 'Add Alias' (the second item in the list) then type your Author Name in the 'New Alias' field. Note that your Author Name (Alias) need not be identical to your User Name.
4. Click the 'Save' button.

Screenshot: Bamboo User Profile

Bamboo Profile: Joe Bloggs

<table>
<thead>
<tr>
<th>Username:</th>
<th>Bloggs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full Name:</td>
<td>Joe Bloggs</td>
</tr>
<tr>
<td>Email:</td>
<td><a href="mailto:bloggs@mycompany.com">bloggs@mycompany.com</a></td>
</tr>
<tr>
<td>JobTitle:</td>
<td></td>
</tr>
<tr>
<td>Source Repository Alias:</td>
<td>Add Alias</td>
</tr>
<tr>
<td>New Alias:</td>
<td>bloggs</td>
</tr>
</tbody>
</table>

RELATED TOPICS

- 11.1 Changing your Password
- 11.2 Changing your Notification Preferences
- 11.3 Associating your Author Name with your User Profile

Bamboo Glossary

Return to Bamboo Documentation Home
Glossary

activity log
artifact
author
build
build activity
build duration
build log
build plan
build queue
build result
build telemetry
builder
committer
favourites
global permission
label
permission
plan
plan permission
project
triggering
watcher
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.
An artifact is something created by a build. There are two types of artifacts:

- User-defined artifacts (e.g. JAR files) are specified in the build's plan by a Bamboo administrator.
- Auto-generated artifacts are created automatically by Bamboo.
An author is any person who contributes to a build by checking-in code to a repository that is associated with a Bamboo plan. An author need not be a Bamboo user.

See 06. Reporting on Author Trends.
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.
build log

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

Every build has a build log. A build log is a permanent record of all the output generated by compiling the plan's source-code and executing the tests.
build plan

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

See plan.
build queue

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

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

Every completed build has a build result:

- 'Successful' — the code compiled, with or without errors, and all tests completed successfully.
- 'Failed' — either the code did not compile, or at least one test failed.

Additionally,

- if the build result is 'Failed', and the previous build result was 'Successful', the build is said to be 'Broken'.
- if the build result is 'Successful', and the previous build result was 'Failed', the build is said to be 'Fixed'.
build telemetry

Build telemetry is the insight provided by Bamboo's dynamic reports, charts and collation of build metrics. Build telemetry helps identify trends across build plans and across authors — not just focusing on the results of a single build.
A builder is a software compiler program external to Bamboo. Bamboo supports multiple builders. Once a builder is defined in the Bamboo system, it can then be specified in build plans by a Bamboo administrator.
A committer is the Bamboo user(s) who committed code to a particular build (i.e. someone who committed code after the previous build was checked out by Bamboo). Administrators can configure a plan’s notifications to be sent to the build's committer(s).
favourites

Each Bamboo user can nominate their favourite plans — that is, the plans they work with the most.

Each user's favourites are displayed on the 'My' page of the Dashboard. Bamboo administrators can also configure each plan to send build result notifications to users who have nominated the plan as one of their favourites (these users are known as the plan's watchers).
global permission

A global permission is the ability to perform a particular operation in relation to Bamboo as a whole. See also plan permission.
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 and to control build expiry.

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.
permission

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

See plan permission and global permission.
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.
plan permission

A plan permission is the ability to perform a particular operation in relation to a build plan. For each plan, different permissions can be granted to particular groups and/or users. See also global permission.
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.
A plan's watchers are the Bamboo users who have marked this plan as one of their favourites. Administrators can configure a plan's notifications to be sent to the plan's watchers.