

# PureCM Data Sheet

PureCM 2009-2 Standard  
Software Configuration Management

PureCM  
16/11/2009



## Contents

Contents	2
PureCM Standard Software Configuration Management	3
What's new in PureCM 2009-2?	4
Managing parallel development	5
PureCM changesets	5
Streams and Workspaces	6
Additional configuration management features	7
Build automation and deployment	8
Administration and security	8
System Requirements	9

## PureCM Standard

# Software Configuration Management

Improving your development lifecycle requires the capability to define, manage and retrieve any software configuration at any time with an absolute minimum of overhead. This is the essence of what you get with PureCM.

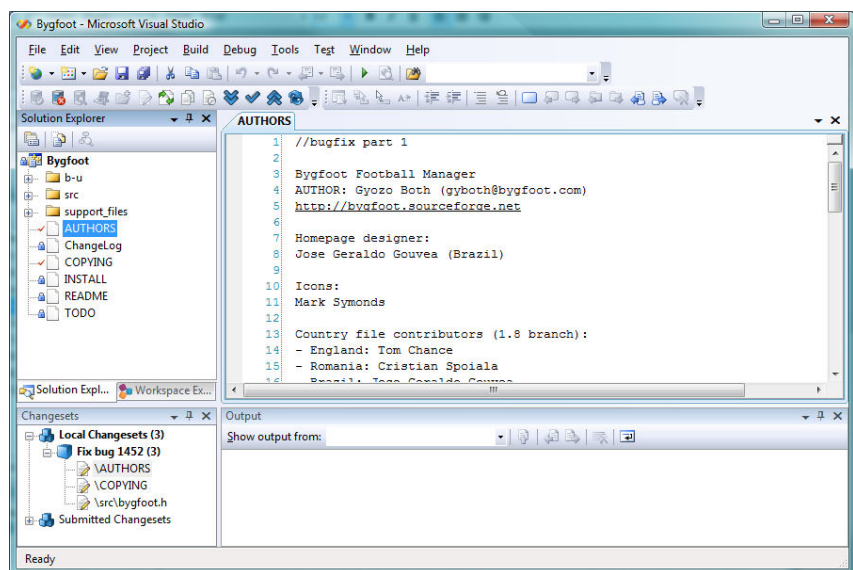
**PureCM Standard is the reliable and easy to use solution you always wanted.** Designed to provide fast, stable and intuitive source control and configuration management, development managers get a tool that helps them to boost quality. Task-based development, lightweight and transparent branching and merging, as well as full support for build automation make sure there's plenty of room for improvement.

PureCM Standard is flexible enough to support any development type whether it's agile or waterfall, desktop applications or web. And thanks to its cross-platform GUI and IDE integrations, developers can work where they want to, with only minimal training requirements.

## What's new in PureCM 2009-2?

PureCM 2009-2 brings more features than ever to the developers' environment, their IDEs. PureCM now includes a **custom plugin for both Visual Studio and Eclipse**. This means more features and better information without leaving your environment.

Custom panes give full insight into changesets submitted against a project, proving team members with full project visibility. Visual Studio users can even **organise checkouts in multiple changesets, use shelving or integrate branches**.

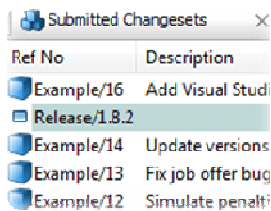


Performance has also improved dramatically, achieving up to **400% faster data transfers** when submitting data to the server. This is especially relevant when working with large binary files as it is common in web or game development.

Development managers now get **more control on the server side**, where they can directly rollback a change or preview and merge them across branches without the need to create a workspace.

## Managing parallel development

PureCM uses two core concepts to make parallel development safer, quicker and easier to understand: **task-based development and streams**.



Ref No	Description
Example/16	Add Visual Studi
Release/1.B.2	
Example/14	Update versions
Example/13	Fix job offer bug
Example/12	Simulate penalti

*PureCM **changesets** are fully dynamic, so a double click allows you to browse through the changes or its merge history.*

### PureCM changesets

Traditional version control tools regarded each file change as separate. Not PureCM. Using PureCM, you can change, add, or delete as many files as you want to fulfil a task. All these changes are grouped into a so called “**changeset**” that can be tracked throughout PureCM.

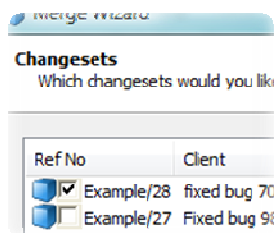
This makes it easy to add a meaningful comment or to associate the changeset with an issue in your issue tracking tool. Even months or years after changing a file, you will **understand why a change was made** and which files were affected.

To maximise database stability, all **changesets are applied atomically**, thus all or none. This makes sure that your database is not left in a corrupted state should the transmission fail in the middle of the transaction.

PureCM’s **Eclipse and Visual Studio integration** reduce training needs when migrating to PureCM. The IDE plugins enable task-based development without forcing developers to quit their favourite environment.

## Streams and Workspaces

Streams in PureCM represent both branches and labels. They can be seen as a fully dynamic snapshot of the code line at the time they are created.



*PureCM automatically detects changesets that that aren't present in the target stream. Simply **cherry pick the changes** you want to merge.*

PureCM streams **inherit code** from their parents and form a hierarchy. This design allows for instant and lightweight branching, making it easy to manage maintenance and development in parallel. Because of PureCM's lightweight streams, you can apply any branching pattern you want, taking hundreds of snapshots along the way.

To **merge changes between streams** or to integrate a branch, you can either merge them into a workspace or directly on the server. Either way, PureCM Standard launches an easy to use visual dialogue that allows you to cherry pick, preview and confirm the change(s) you want to merge.

As streams remember their hierarchy, **merge tracking** is fully automated. Each changeset remembers where it was merged to or from to give you full transparency of your development.

To work on their project, developers simply create a **local workspace** from the stream they want to work on, which lets them isolate their work until finished. They decide themselves when it's time to integrate changes from colleagues and submit their own changes, minimising disruption and enhancing their productivity.

**Read/write access** to all objects in PureCM can be defined setting policies against users or user groups, providing a powerful yet simple way to protect your key assets. Whether you want to protect a whole release or just a single file, PureCM Standard puts you in control.

## Additional configuration management features

- **Private workspace.** Each developer works in their private workspace, which is a local copy of the code. This allows you to make changes in isolation from your colleagues, create revisions, test them and integrate your colleagues' changes in a controlled manner.
- **Shelving and local revisions.** Want to checkpoint your ongoing work or pass it to your colleague? Simply shelve your change on the server. PureCM Standard even supports creating local revisions in your workspace, e.g. when working without server connection.
- **File history.** The PureCM file history shows a summary of each file revision. From there, you can dynamically access every changeset to see what happened in any revision.
- **Annotated file history.** Visualises a file's revisions by using colour codes. Highlights when and by whom a change was made for each line of code – especially useful for bug fixing.
- **Stream comparer.** Visual tool to highlight file and folder differences between any two configurations: two releases, two branches or even your local workspace vs. the server version.
- **Multiple checkout and file lock.** PureCM Standard allows you to define multiple checkout or file locking for each file type, stream or user.
- **Live checkout tracking.** Indicates for each file by whom it is currently checked out. Very useful to know who to contact when using file locks.
- **Shared components.** Define component folders that are shared among different PureCM streams to keep them automatically in sync. Manage component updates centrally in the PureCM Component Manager.
- **Cross-platform server and GUI.** PureCM comes with a Windows, Linux, Mac and Solaris GUI. Mix and match any client server setup according to your needs.
- **Eclipse and Visual Studio integration.** Use the functionality of PureCM within your favourite IDE.

## Build automation and deployment

Use PureCM Standard to improve the productivity of your development team. Automation allows you to speed up repetitive tasks, to avoid human errors, and to provide instant feedback to developers.

- **FinalBuilder and CruiseControl.NET integration.** Use the out-of-the-box integration with these favourite build tools to quickly set up an automated build solution.
- **Triggers and command line interface.** Define custom actions based on triggers in PureCM. The existing .NET and Java SDK greatly simplify the development of custom scripts.

## Administration and security



*Use PureCM's simple but powerful security policies to manage access to your configuration items.*

PureCM is designed to minimise administrative overhead based on policy-based administration.

- **Windows domain authentication** and automatic user creation. Alternatively, use password authentication or security certificates to control database access.
- **ACLs and user groups.** Permissions are policy-based to control access to all configuration items. Policies can be set against individual users and user groups.
- **SSL transmission.** Ensure secure remote access by optionally enforcing SSL encryption or VPN use.
- **Backups and database integrity checks.** PureCM allows for full automation of backup creation and database integrity testing, using its command line client and administrative tool “tdbutil”.

Platform Support	System Requirements for Server
<ul style="list-style-type: none"><li>▪ Windows client: 2000/XP/Vista/Windows 7</li><li>▪ Windows server: NT, 2003, 2008</li><li>▪ Linux Fedora 5+, Red Hat Enterprise 5+, SuSE SLES 10 SP1, Ubuntu 8.04+</li><li>▪ Mac OS X+</li><li>▪ Sun Solaris 8+</li></ul>	<ul style="list-style-type: none"><li>▪ 133 MHz or higher Pentium compatible CPU</li><li>▪ Recommended RAM of your OS plus 10 MB per user</li><li>▪ Free disk space for double the size of the source code</li></ul>