Configuration Management Procedures

Version 1.41

Edition 1

Document ID:	CM
Author:	John Newbigin
Reviewed by:	Danny Goulas Jaycee Phua
Date:	2 September 1999

Revision History

Date	Modifications	Reason	Version
1999.03.15	Document created (JN)	Explain Configuration Management Procedures	1.0
1999.03.31	Contents checked and formatted (DG)	Adherence to Style specification	1.4
1999.09.02	Content and formatting updated (JP)	Reflect changes to Document Style and Team re-structure	1.41

1. Introduction

All repository access must be in accordance with these procedures:

Module	Access
`test1`	All **Team Synergy** members
`docs`	All **Team Synergy** members
`cohesion`	*Application Engineers*, *Quality Assurance*
`cgi-src`	*Publishing Group*
`CVSROOT`	Configuration Manager

2. Code

Code can be checked out of the repository by developers and quality assurance.

Code can only be checked in by developers, after a peer review.

All code must fit into the following directory hierarchy

\<module name>

\src

\Makefile \make.bat \<package name> \.java files \build \lib \doc

Only *.java* files and the make files are permitted to be checked in to the repository. Before any files can be added to the repository, they must be approved by the *Application Architect* .

Before any files are checked in to the repository, they must be approved with peer review. The Log message must contain the name of the person who performed the peer review. The peer review does not have to be too formal; all that is required is getting a fellow developer to look over the code, making sure of no obvious flaws. If you are working remotely, you can e-mail the person the files and a description of what you have changed. Wait for a reply before checking in the code.

All code which is checked in to the repository must compile. At any time, a copy of the CVS code should be able to be checked out and built without any errors. It does not have to be functional, just compile.

You should consult the *CVS Instructions* documentation for details on how to use CVS.

Tagged versions will only be created by the *Configuration Manager*, under the direction of the *Application Architect* .

All code checked in to the repository must contain the comment

```
// $Header: /teamproj/cvsroot/teamb/docs/configuration_management.html,v 1.4
1999/03/31 01:18:36 tpXXXXXX Exp $
```

at the top of the file (where the text highlighted in red is your appropriate Student ID). This enables easy tracking of the files when they are out of the repository.

3. Documentation

All official documentation should be kept in the CVS `docs` module. This allows documentation to be added by the author, edited by the Publishing Group, certified by *Quality Assurance* (QA), and then published by the *Publishing Group* .

At each stage, the file should be checked in and then a message sent to the appropriate person who needs to see the document next.

For example, a document is written and added to the `docs` module. A message is then sent to the *Publishing Group*, or at least the Head of the *Publishing Group*. Once the Publishing Group complete validation, correspondence is sent to either all, or the Head of QA for verification. Once complete, a return-message is sent by *QA* back to the *Publishing Group* allowing publication. Obviously if at any stage there are problems with the document, it is sent back to the author for updates.

All documentation checked in to the repository must be in *HTML* format. *GIF* and *JPG* images can be checked in but remember that they must be set as binary files.

The directory structure for the docs module will be set by the *Publishing Group*. Before any document is added, it must be verified by a member of the *Publishing Group* .

All documents must contain the comment:

```
<!-- $Header: /teamproj/cvsroot/teamb/docs/configuration_management.html,v 1.4
1999/03/31 01:18:36 tpXXXXXX Exp $ -->
```

to allow easy tracking of files when they are out of the repository, where red signifies your Student ID. At the discretion of the *Publishing Group*, there may be requirements for the inclusion of the date as follows:

```
$Date: 1999/03/31 01:18:36 $
```

and others.