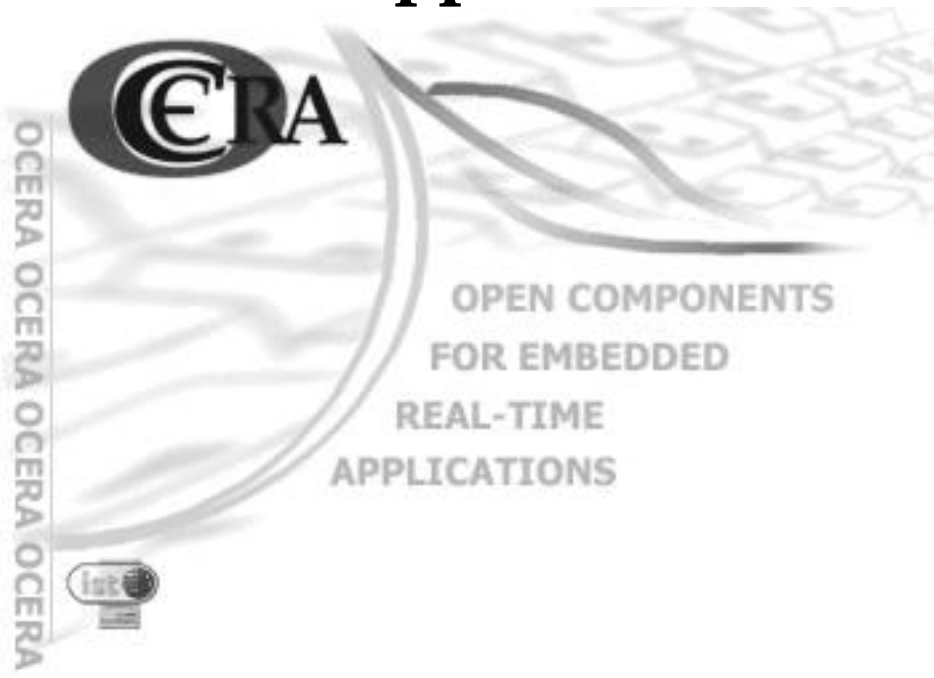
WP10 - Training and Technical Support



Deliverable D10.2 - CVS server specification

WP10 - Training and Technical Support : Deliverable D10.2 - CVS server specification
by Pierre Morel and Adrian Matellanes

Published January 2003
Copyright © 2003 by Ocera

Table of Contents

Document presentation ..... **i**

1. Document Overview **1**

2. OCERA CVS (Concurent Version System) **2**

2.1. Hosting the CVS server 2
2.2. Location 2
2.3. SourceForge CVS presentation 2
2.3.1. About CVS 2
2.3.2. Anonymous CVS Access 2
2.3.3. Developer CVS Access via SSH 3
2.3.4. Browse the CVS Tree 3

3. OCERA CVS Content **4**

3.1. Overview 4
3.2. Documentation 4
3.2.1. Images directory 4
3.2.2. Docbook exemple directory 5
3.2.3. Docbook DSSSL directory 5
3.2.4. WP directories 5
3.3. Applications development tree 5
3.4. Components development tree 5
3.5. patch tree 6
3.6. kernel tree 6
3.7. contrib tree 6
3.8. CVS Tree Structure 6

4. How to use the OCERA CVS **10**

4.1. Browsing the source tree 10
4.2. Developpers updating the source tree 10
4.3. Using the Docbook DSSL utilities 10
4.4. Solutions for different problems 11

5. Administration of the OCERA CVS **12**

5.1. Overview 12
5.2. Naming conventions 12
5.2.1. Components Directories 12
5.2.2. Components files 12
5.2.3. Documentation files 12
5.3. Making Backup 12
5.4. What to do in case of problem 12
5.4.1. I want to commit my changes but I can't 13
5.4.2. SourceForge Site temporarly not available 13
5.4.3. SourceForge Site definitivly not available 13
5.4.4. I want to retrieve a file from the snapshot archive 13

List of Tables

1. Project Co-ordinator i
2. Participant List.....i

Document presentation

Table 1. Project Co-ordinator

Organisation:	UPVLC
Responsible person:	Alfons Crespo
Address:	Camino Vera, 14. CP: 46022, Valencia, Spain
Phone:	+34 9877576
Fax:	+34 9877579
E-mail:	alfons@disca.upv.es

Table 2. Participant List

Role	Id.	Name	Acronym	Country
CO	1	Universidad Politécnica de Valencia	UPVLC	E
CR	2	Scuola Superiore S. Anna	SSSA	I
CR	3	Czech Technical University in Prague	CTU	CZ
CR	4	CEA	CEA	FR
CR	5	UNICONTROLS	UC	CZ
CR	6	MNIS	MNIS	FR
CR	7	VISUAL TOOLS S.A.	VT	E

Chapter 1. Document Overview

This document present the OCERA CVS server structure and the way to use it for developpers and for guests.

This document is subject to change allong the project, to reflect the up-to-date project tree and rules.

This document will cover the following aspect of the *Concurent Version System*:

- The *Introduction* will present the CVS server.
- The *Content* will present the documents hold on the server, and the organization of the source and documentation tree
- The *Usage and policy* chapter will present how developpers and guests can access the CVS repository to retrieve informations or how the developpers can use the CVS to store the source and documentation. We will also write some rules for the OCERA-CVS policy.
- The chapter on *Administration* will present different aspect of the CVS administration like backup, unavalability and so on.

Chapter 2. OCERA CVS (Concurent Version System)

2.1. Hosting the CVS server

We used SourceForge to host the OCERA CVS server, because it is the most popular OpenSource hosting provider used by the Linux community, and is known for:

- **popularity**

  SourceForge it is the most popular OpenSource hosting provider used by the Linux community.

- **functionalities**

  SourceForge proposes a lot of integrated functionalities dedicated to developpers that we can use at the side of the cvs server, like bug tracking system, mailing lists, news

- **easy administration**

  The administration of the SourceForge CVS server is easy and can be accessed from all over the Internet.

2.2. Location

The CVS server is located at SourceForge, at cvs://cvs.ocera.sourceforge.net/ and can be browse with a web interface at URL: http://cvs.sourceforge.net/cgi-bin/viewcvs.cgi/ocera

2.3. SourceForge CVS presentation

Here follows the presentation you can find on the SourceForge web site. We have changed the examples to fit our project.

2.3.1. About CVS

CVS (Concurrent Versions System) is a tool used by many software developers to manage changes within their source code tree. CVS provides the means to store not only the current version of a piece of source code, but a record of all changes (and who made those changes) that have occurred to that source code. Use of CVS is particularly common on projects with multiple developers, since CVS ensures changes made by one developer are not accidentally removed when another developer posts their changes to the source tree.

In order to access a CVS repository, you must install a special piece of software called a CVS client; CVS clients are available for most any operating system (we include information about many popular CVS clients in our site documentation collection).

2.3.2. Anonymous CVS Access

This project's SourceForge.net CVS repository can be checked out through anonymous (pserver) CVS with the following instruction set. The module you wish to check out must be specified as the modulename.

When prompted for a password for anonymous, simply press the Enter key. To determine the names of the modules created by this project, you may examine their CVS repository via the provided web-based CVS repository viewer.

```
cvs -d:pserver:anonymous@cvs.sourceforge.net:/cvsroot/ocera login
cvs -z3 -d:pserver:anonymous@cvs.sourceforge.net:/cvsroot/ocera co modulename
```

Updates from within the module's directory do not need the -d parameter.

2.3.3. Developer CVS Access via SSH

Only project developers can access the CVS tree via this method. A SSH client must be installed on your client machine. Substitute modulename and developername with the proper values. Enter your site password when prompted.

A significant amount of information about project CVS services may be found in our "Introduction to Project CVS Services".

```
export CVS_RSH=ssh
cvs -z3 -d:ext:developername@cvs.sourceforge.net:/cvsroot/ocera co modulename
```

2.3.4. Browse the CVS Tree

Though CVS repositories are most commonly accessed using a special piece of software called a CVS client, we also provide a web-based interface to view CVS repositories. Browsing the CVS tree gives you a great view into the current status of this project's code. You may also view the complete histories of any file in the repository.

Chapter 3. OCERA CVS Content

3.1. Overview

The Ocera CVS server holds different files in the following trees:

- **The documentation**

  The OCERA documentation contains all deliverables of the OCERA project.

  The documentation related to the components or to the applications will be hold inside of the component, i.e. application's tree.

- **Components development tree**

  The components development tree contains the source build files and documentations for the different components, the tree is divided into schedulers, qos (Quality Of Service), fault-tolerance, and communications.

- **Applications development tree**

  The applications development tree contains a subtree per application and then is divided into vision control, robotic, communication

- **kernel**

  The `kernel` directory is intended to hold all generic kernel patches for the OCERA project like the basic patches for linux or RTLinux that are not included in the OCERA componant.

  Typically, the RTLINUX-3.1_patch itself.

- **contrib**

  The `contrib` directory is intended to hold all patches (possibly from third parties) for the OCERA project that provide specific features not included in the OCERA componant or applications.

  There could be hold generic patches like some rtlinux or linux patch that are not incompatible with OCERA components but not part of it.

- **Patch**

  The patch development tree contains patchs and source code and associated documentation that are correcting bugs from a previous version, or implement a *deviant* version that we still do not want to have in the main tree.

3.2. Documentation

This subtree hold every thing abour documentation, including building files (Makefiles, Rules.make ...), images and dockbook libraries and stylesheets.

3.2.1. Images directory

You find there the images used by the project, like logo, background images for powerpoint and so on.

3.2.2. Docbook exemple directory

You can find there an exemple on to generate a docbook document with several widely used tags as well as its corresponding Makefile.

3.2.3. Docbook DSSSL directory

This part of the cvs tree contains the docbook DSSSL style sheet that is use to build the documentation.

3.2.4. WP directories

This is where you find the directories for the workpakages in the form: WPnumber where number is the IST number for the workpackage.

Under each WorkPackage directory you find one directory per deliverable, with associated xml files, images and makefile.

3.3. Applications development tree

This is the reserved place for the applications using the OCERA components during the first stage of development, three applications will be developped and three subtree will hold these applications source and build files.

Each application reflect a real world need and will be used by the Industry developping the application:

- **Robotic application (CEA)**

  a robotic application applied to a motorised arm for handicaped people.

- **Multimedia application (VISUAL TOOLS)**

  a Multimedia application applied to a video-surveillance center.

- **Process Control application (UNICONTROL)**

  a process control application applied to the control of a high presseure compressor station of a gas transit pipeline.

3.4. Components development tree

This is the reserved place for the components building of OCERA during the first stage of development, four components categories will be developped and then, four subtree will hold these components source and build files.

Each component category may hold several components, schedulers for exemple may hold CBS and EDF components.

- **communications**

  The forseen component are realtime ethernet and CAN support

- **fault-tolerance**

  fault-tolerance may interact with other components, so the components there must contain explicit links to related components like schedulers and qos.

- **Quality Of Service**

  Quality of service may interact with other components, so the components there must contain explicit links to related components like schedulers.

- **Schedulers**

  A lot of sub directories here will contain the different schedulers, related components and the associated files.

- **Misc**

  Here will go all components that cannot be classified in the groups above, like "High resolution timer patch" "Posix port" "Ada run time"...

3.5. patch tree

The `patch` directory is intended to comprise all patches (possibly from third parties) for the OCERA project that provide specific features not included in the OCERA development specifications. There could be, for example, a patch for the SPARC architecture, etc...

3.6. kernel tree

The `kernel` directory is intended to hold all generic kernel patches for the OCERA project like the basic patches for linux or RTLinux that are not included in the OCERA componant.

Typically, the RTLINUX-3.1_patch itself.

We will divide this tree in Linux and RTLinux subtree.

3.7. contrib tree

The `contrib` directory is intended to hold all patches , programms or libraries possibly from third parties, for the OCERA project that provide specific features not included in the OCERA componant or applications.

For exemple we hope to find here, tests programms, reports on tests, new applications.

There could also be store generic patches like some rtlinux or linux patch that are not incompatible with OCERA components but not part of it.

3.8. CVS Tree Structure

Lets present the current CVS directory tree.

```
|-- ocera
|   |-- app
|   |-- components
|   |   |-- communications
|   |   |   |-- canvca
|   |   |   |   |-- doc
|   |   |   |   |-- examples
|   |   |   |   `-- src
|   |   |   |-- candev
|   |   |   |   |-- doc
|   |   |   |   |-- examples
|   |   |   |   `-- src
|   |   |   |-- canmon
|   |   |   |   |-- doc
|   |   |   |   |-- examples
|   |   |   |   `-- src
|   |   |   |-- ethdev
|   |   |   |   |-- doc
|   |   |   |   |-- examples
|   |   |   |   `-- src
|   |   |   |-- ethanal
|   |   |   |   |-- doc
|   |   |   |   |-- examples
|   |   |   |   `-- src
|   |   |   |-- verbus
|   |   |   |   |-- doc
|   |   |   |   |-- examples
|   |   |   |   `-- src
|   |   |   `-- versched
|   |   |       |-- doc
|   |   |       |-- examples
|   |   |       `-- src
|   |   |-- fault-tolerance
|   |   |   |-- ftappmonitor
|   |   |   |   |-- doc
|   |   |   |   |-- examples
|   |   |   |   `-- src
|   |   |   |-- ftbuilder
|   |   |   |   |-- doc
|   |   |   |   |-- examples
|   |   |   |   `-- src
|   |   |   |-- ftcontroller
|   |   |   |   |-- doc
|   |   |   |   |-- examples
|   |   |   |   `-- src
|   |   |   |-- ftredundancymngr
|   |   |   |   |-- doc
|   |   |   |   |-- examples
|   |   |   |   `-- src
|   |   |   `-- ftreplicamngr
|   |   |       |-- doc
```

```
|   |   |       |-- examples
|   |   |       `-- src
|   |   |-- qos
|   |   |   |-- gensched
|   |   |   |   |-- doc
|   |   |   |   |-- examples
|   |   |   |   `-- src
|   |   |   |-- qlib
|   |   |   |   |-- doc
|   |   |   |   |-- examples
|   |   |   |   `-- src
|   |   |   |-- qmngr
|   |   |   |   |-- doc
|   |   |   |   |-- examples
|   |   |   |   `-- src
|   |   |   `-- qres
|   |   |       |-- doc
|   |   |       |-- examples
|   |   |       `-- src
|   |   `-- sched
|   |       |-- appsched
|   |       |   |-- doc
|   |       |   |-- examples
|   |       |   `-- src
|   |       |-- appschedcbs
|   |       |   |-- doc
|   |       |   |-- examples
|   |       |   `-- src
|   |       |-- appschededf
|   |       |   |-- doc
|   |       |   |-- examples
|   |       |   `-- src
|   |       |-- dynmem
|   |       |   |-- doc
|   |       |   |-- examples
|   |       |   `-- src
|   |       |-- pbarriers
|   |       |   |-- doc
|   |       |   |-- examples
|   |       |   `-- src
|   |       |-- pmqueue
|   |       |   |-- doc
|   |       |   |-- examples
|   |       |   `-- src
|   |       |-- psignals
|   |       |   |-- doc
|   |       |   |-- examples
|   |       |   `-- src
|   |       |-- ptimers
|   |       |   |-- doc
|   |       |   |-- examples
|   |       |   `-- src
|   |       |-- ptrace
|   |       |   |-- doc
|   |       |   |-- examples
```

```
|   |       |   `-- src
|   |       |-- rtlcbs
|   |       |   |-- doc
|   |       |   |-- examples
|   |       |   `-- src
|   |       |-- rtlgnat
|   |       |   |-- doc
|   |       |   |-- examples
|   |       |   `-- src
|   |       `-- rtlhrt
|   |           |-- doc
|   |           |-- examples
|   |           `-- src
|   |-- doc
|   |   |-- README
|   |   |-- Rules.make
|   |   |-- WP1
|   |   |-- WP10
|   |   |-- WP11
|   |   |-- WP12
|   |   |-- WP13
|   |   |-- WP2
|   |   |-- WP3
|   |   |-- WP4
|   |   |-- WP5
|   |   |-- WP6
|   |   |-- WP7
|   |   |-- WP8
|   |   |-- WP9
|   |   |-- cover.xml
|   |   |-- doall
|   |   |-- dsl
|   |   |-- example
|   |   |-- gen_tar
|   |   |-- img
|   |   `-- license.xml
|   |-- crontrib
|   |-- kernel
|   |   |-- linux
|   |   `-- rtlinux
|   `-- patch
```

Chapter 4. How to use the OCERA CVS

4.1. Browsing the source tree

The simpliest way to browse the source tree is to use the CVS-to-web interface at sourceforge at http://www.sourceforge.net

If you want to retrieve a complete subtree you may find easier to use a direct cvs command like:

```
cvs -d:pserver:anonymous@cvs.sourceforge.net:/cvsroot/ocera login
cvs -z3 -d:pserver:anonymous@cvs.ocera.sourceforge.net:/cvsroot/ocera co ocera/doc/WP2
```

4.2. Developpers updating the source tree

There are several reasons why it is important to **commit** changes **as soon as possible**:

- Let the others use and test your software

  It is important to test the software in different machines, by different people, in different situations and environments; this way we will increase software quality and robustness and we will save a lot of testing time...the others will do it for us!

- Avoid integration problems

  If we **commit** our changes soon, we will make it possible to begin integration with other modules soon. If we do not do that we will, for sure, suffer large periods of integration, API incompatibilities, etc...

- Avoid data losses

  Having your sources in CVS as soon as possible will guarantee that you have stable versions in a safe place. Catastrophic problems with your local machine will no longer mean a loss of source code, etc..

- Avoid wasting time in known errors

  Commit your changes, specially if they solve a problem, there might be people using your software and hving that problem, trying to solve it, etc...

Anyhow this arises the question: *What does it mean "as soon as possible"?* We use to think that our software is never good enough to **commit** it into CVS. A good starting point is just to define some small feature and to **commit** the sources when that feature is implemented.

A necessary condition to **commit** source changes is that the building process (usually a **make** command) is still working.

4.3. Using the Docbook DSSL utilities

A part of the documentation (/doc) tree is containing dsssl style sheet and Rules.make file. You may want to download these files to build the documentation.

We will use as exemple the WP2 documentation.

Use the following command to retrieve for the WP2 documentation:

```
cvs -d:pserver:anonymous@cvs.sourceforge.net:/cvsroot/ocera login
cvs -z3 -d:pserver:anonymous@cvs.ocera.sourceforge.net:/cvsroot/ocera co
ocera/doc/WP2
cvs -z3 -d:pserver:anonymous@cvs.ocera.sourceforge.net:/cvsroot/ocera co ocera/doc/WP2
cvs -z3 -d:pserver:anonymous@cvs.ocera.sourceforge.net:/cvsroot/ocera co ocera/doc/Rules.make
cvs -z3 -d:pserver:anonymous@cvs.ocera.sourceforge.net:/cvsroot/ocera co ocera/doc/dsl
```

Use the following command to build the WP2 documentation:

```
cd WP2
make
```

Note that the first command, login, is only mandatory the first time you log in the cvs repository.

4.4. Solutions for different problems

Most of the firewall allow the users to access the cvspserver port. However, if you have problem with this, the cvspserver port at sourceForge is the standard port, that is 2401, you may have to signal this to your system administrator.

Chapter 5. Administration of the OCERA CVS

5.1. Overview

The administration on a foreign repository forces us to be very carefull in the choices we made when creating the trees because we wont be able to remove directories.

For this, we will restrict the administrative access to the CVS to two people. The administrators are named on the SourceForge site.

5.2. Naming conventions

5.2.1. Components Directories

Each module corresponding to a Component will have a `src` directory (including header files, possibly under an `include` subdirectory), a `doc` directory for specific documentation and an `examples` directory for basic examples of use. There will also be a `README`, `Makefile` and `ChangeLog` files.

- Scheduling components will go under `ocera/components/sched`,
- Resource Reservation components under `ocera/components/qos`,
- Fault-Tolerance components under `ocera/components/fault-tolerance`
- and Communications under `ocera/components/communications`.

Components will be placed in each `ocera/components/component_kind/component_name`, regardless of the nature and scope of the component, i.e., whether it is a kernel module, a library, a tool or a patch

5.2.2. Components files

The files will be written with lowercases, with underscore to separate componant_subcomponant, dash to separate version number and sub_version number.

For exemple: patch_linux-2.4.18-cbs or posix_barrier-rtlinux-3.1

5.2.3. Documentation files

The documentation files for development must reflect the componant they apply to for example: posix_barrier-rtlinux-3.1.txt or readme.posix_barrier-rtlinux-3.1 .

The documentation files for deliverable are free, it is the main xml file reflect the deliverable number, for example: D10-2.xml

5.3. Making Backup

We cannot make a complete backup of the CVS because we do not have access to all control files.

We are checking out locally every week, the complete CVS source tree, and making a backup of this snap shot. We must then be able to recover the files we need in case of disaster.

5.4. What to do in case of problem

5.4.1. I want to commit my changes but I can't

Verify that you have the right to do such commit: ask your WorkPackage responsible. Then email to one of the administrators.

5.4.2. SourceForge Site temporarly not available

The best thing is to wait until it is available. You may also encounter problems with a fiewall, the consult your local system administrator.

5.4.3. SourceForge Site definitivly not available

If the SourceForge CVS site for ocera is becoming unavailable for some yet unknown reason, we will establish a new CVS server using the last snapshot we have.

Doing this, we will need informations from all people having post some file on the CVS server to know if there is interressant version before the actual last version. Like for exemple stable versions.

You will then be able to change your CVSROOT environment variable or use the new cvs site in your scripts.

5.4.4. I want to retrieve a file from the snapshot archive

Ask an administrator if you think one file may be on a snapshot and is not available on the CVS server.