CVS : Concurrent Versions System

Wim Delvaux

Agenda

- The Problem
- A bit of history
- Features
- Setting up
- GUI
- WEB
- Reference(s)

The problem : Software Configuration Mgmt

- Managing different releases of software source code
- Facilitating development with multiple developers
- Being able to retrieve sources of previous releases
- Tracking software changes over releases

A bit of history : SCCS

- Source Code Control System
  - Oldest system
  - Works on 1 file
  - Not networked
  - Stores differences between releases
  - Oldest release is complete
  - Part of UNIX and not really free

A bit of history : RCS

- Revision Control System
  - A bit newer
  - Still works on 1 file
  - Still not networked
  - Also stores differences between releases
  - Newest release is complete
  - Free

A bit of history : CVS

- Concurrent Versions System
  - Newest platform
  - Works op project level
  - Supports loosely networked environments
  - Support for multiple developers each in their own work environment
  - Stores differences between releases
  - Based on RCS
  - Free (has paying support)

A bit of history : Non-free

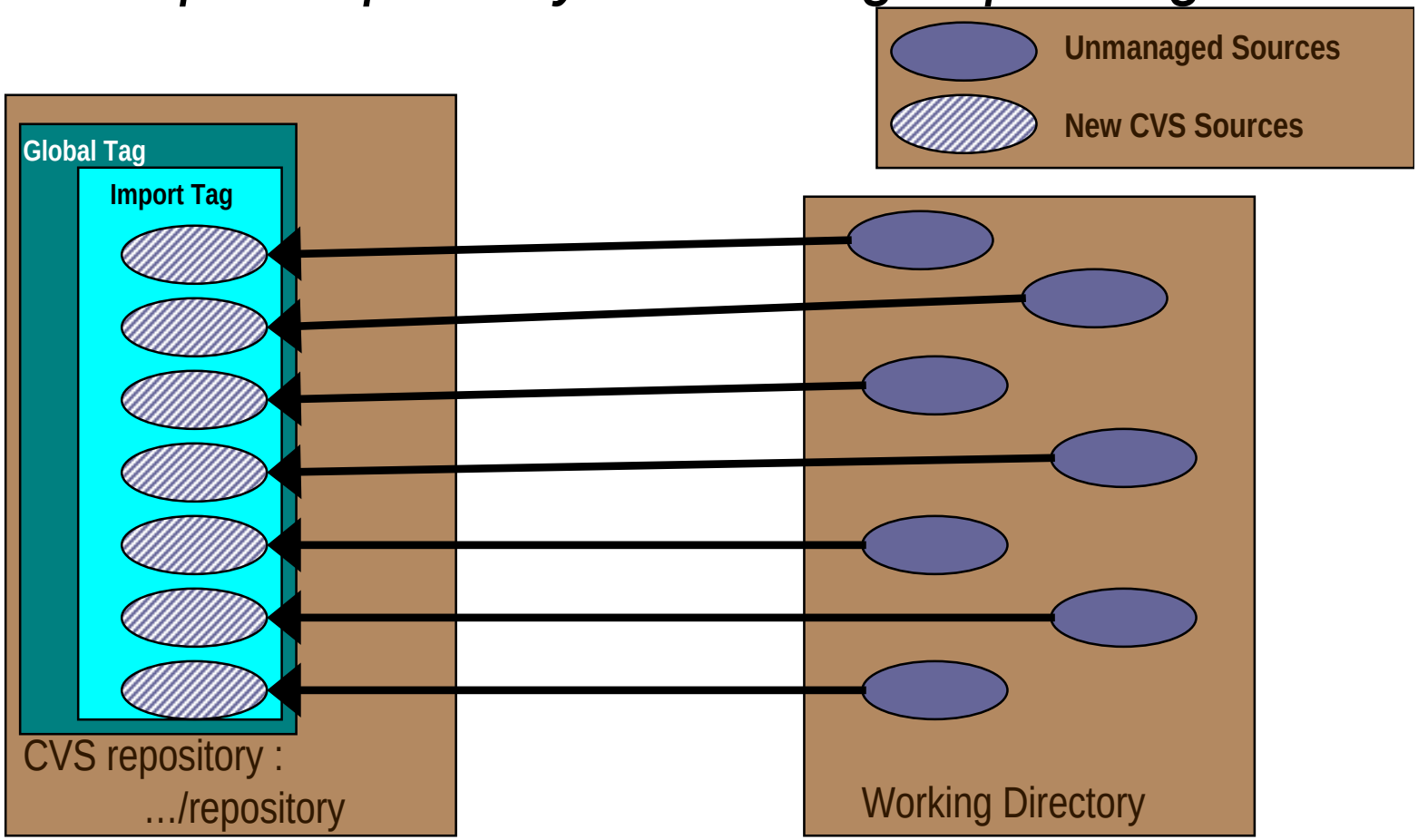
- PVCS, ClearCase
  - Most powerful platforms
  - Work on project level
  - Tightly coupled networks
  - Powerful storage mechanisms

CVS terminology

- **object** Any ‘file’ under control of CVS
- **Module** ID of a collection of objects
- **Version** Revision number of one object
- **Tag** Common ID of all ‘current’ objects in a module
- **Branch** Split in the revision path that enables parallel versions

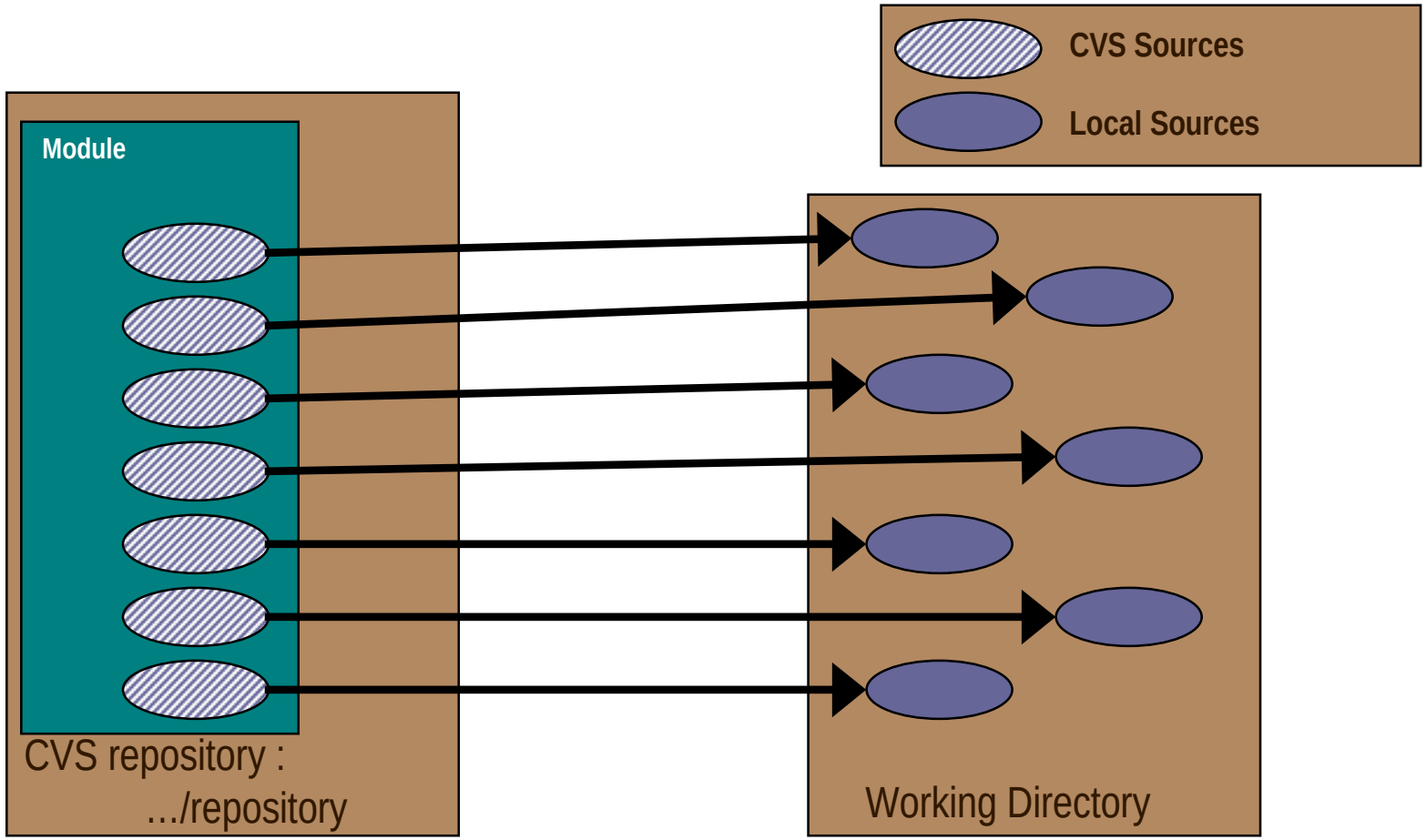
CVS Features : Storing new objects

- *cvs import repository GlobalTag ImportTag*



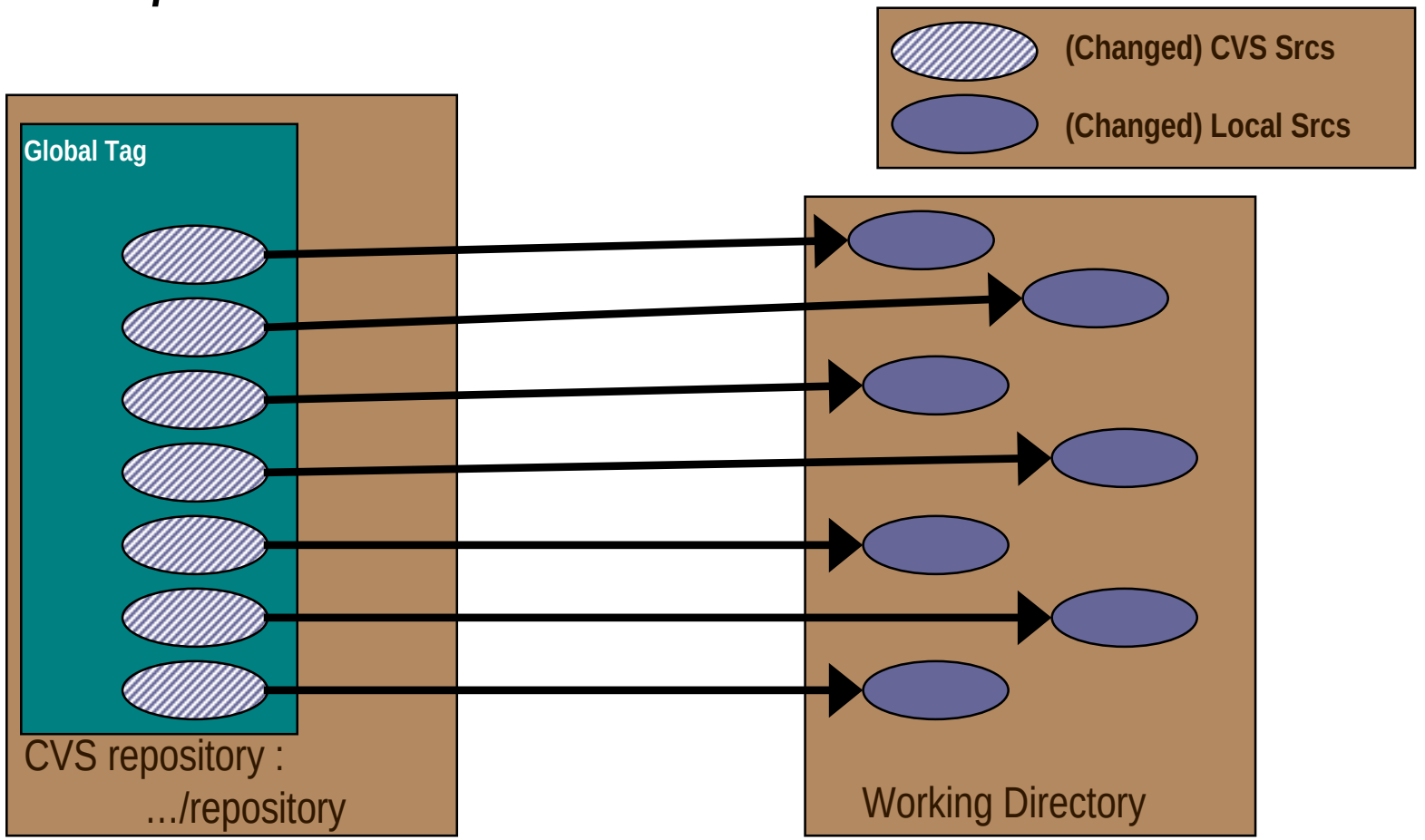
CVS Features : Retrieving objects

- *cvs checkout ModuleName*



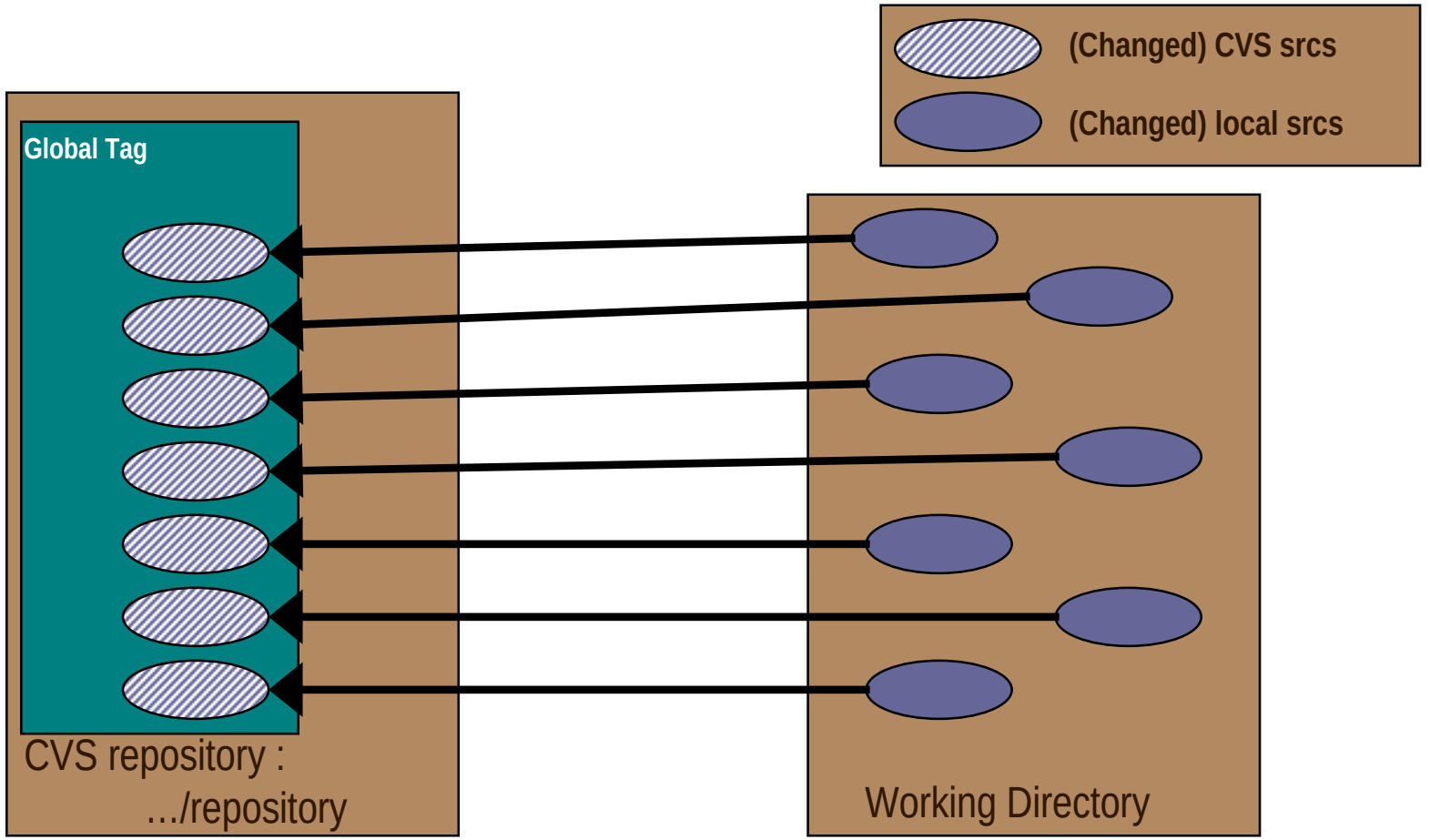
CVS Features : Resynchronizing objects

- *cvs update*



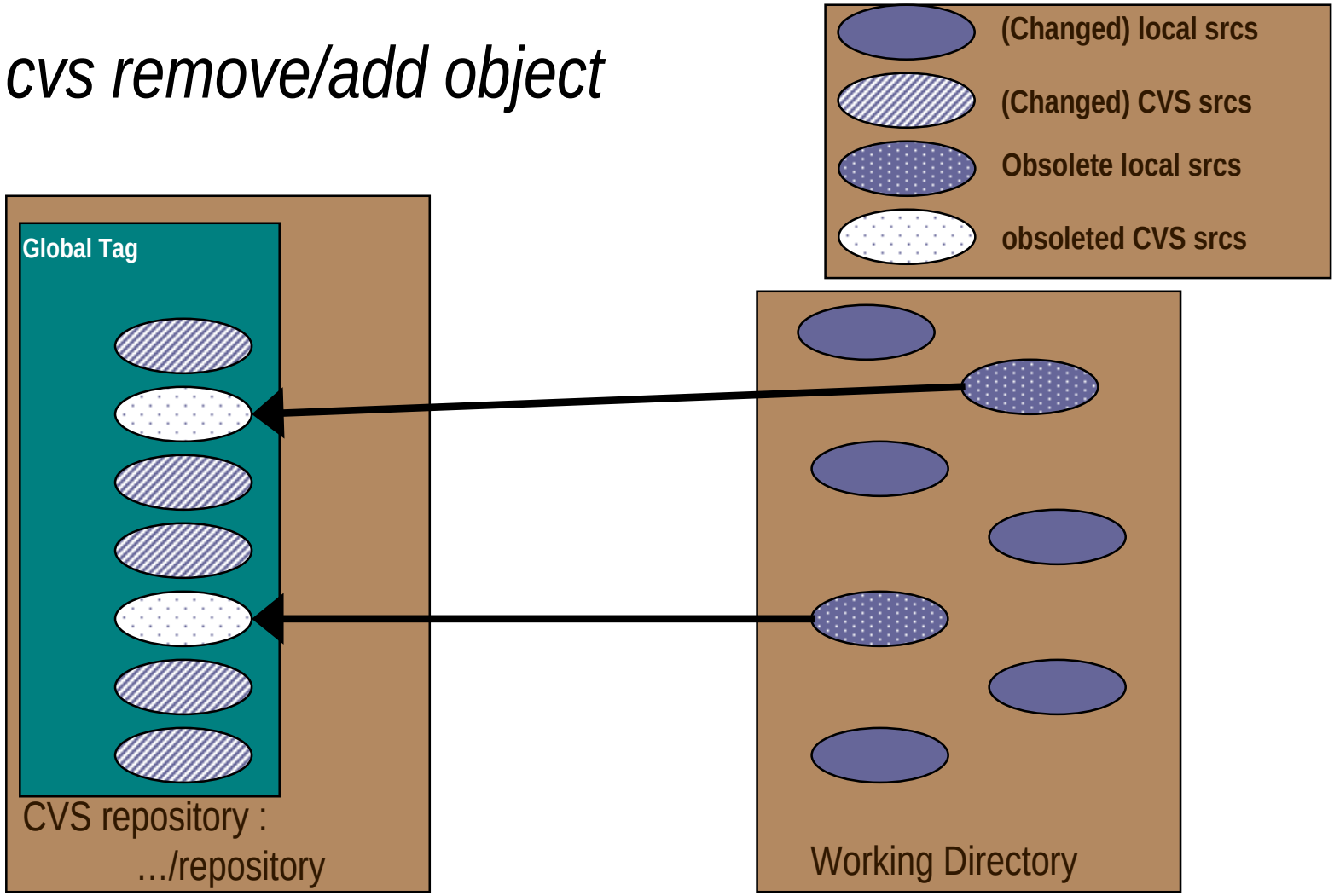
CVS Features : Committing objects

- *cvs commit*



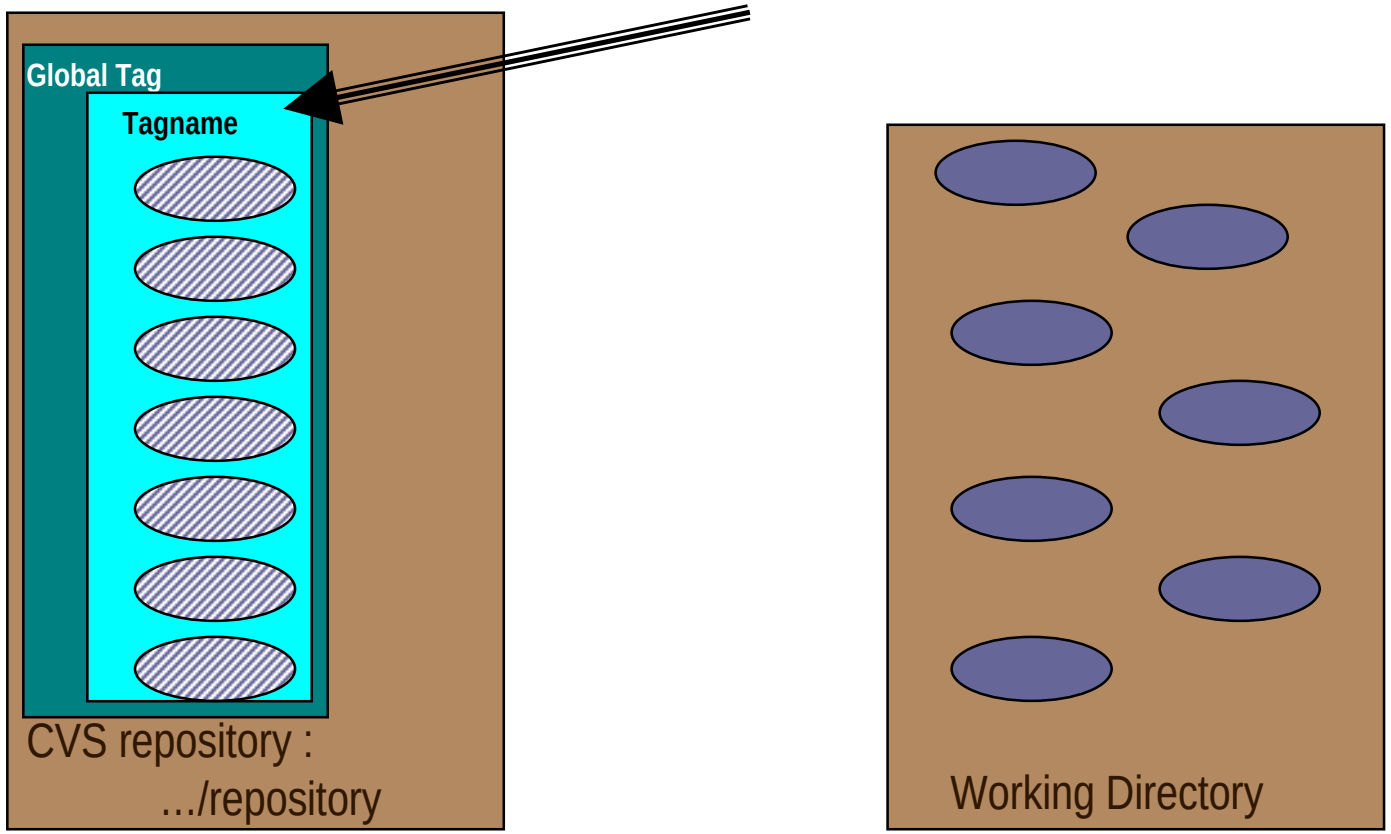
CVS Features : Adding/Removing objects

- *cvs remove/add object*



CVS Features : Tagging objects

- *cvs tag tagname*



CVS Features : Miscellaneous

- *cvs watch on/off*
- *cvs watch add/remove*
- *cvs edit*
- *cvs watchers*
- *cvs editors*

Setting up

- Download form any distribution
  - source from : http://download.cyclic.com/pub/
- Client access to server defined by CVSROOT environment variable
- Always need to define cvsuser user

Setting up : Local

- Create with : cvs -d /usr/local/xyz init
- CVSROOT=/usr/local/xyz

Setting up : Rsh

- setup .rhosts file for all user
- CVSROOT=“:server:Username@hostname:/usr/local/xyz”

Setting up : pserver

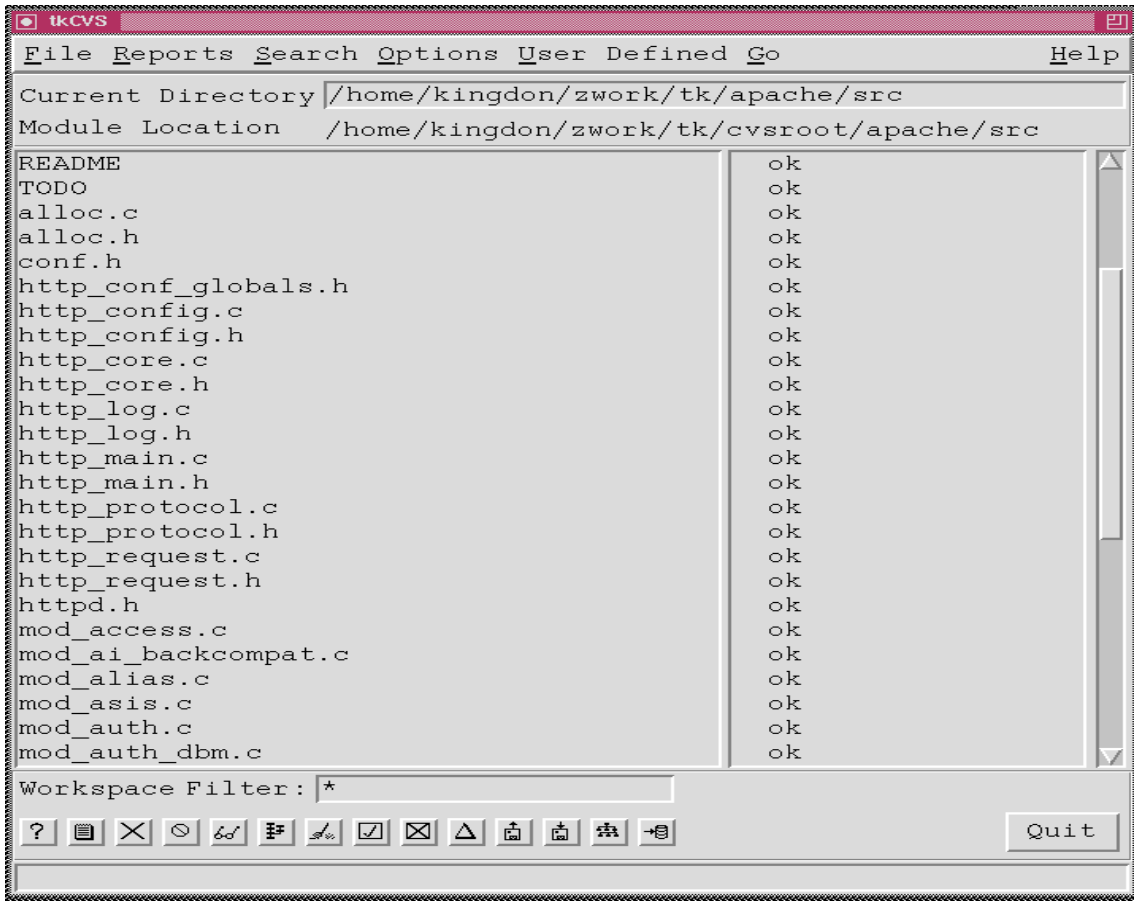
- add cvs to inetd.conf
  - cvspserver stream tcp nowait cvsuser /usr/bin/cvs -b /usr/bin--allow-root=/usr/local/xyz pserver
- CVSROOT=“:pserver:username@hostname:/usr/local/xyz”
- Log on : cvs logon
  - Password file : unix or special
  - Security/ Auditing : unix based

Related software : GUI’s

- tkCVS
- jCVS
- WinCVS/MacCVS
- gCVS
- CVSmanager
- Pharmacy

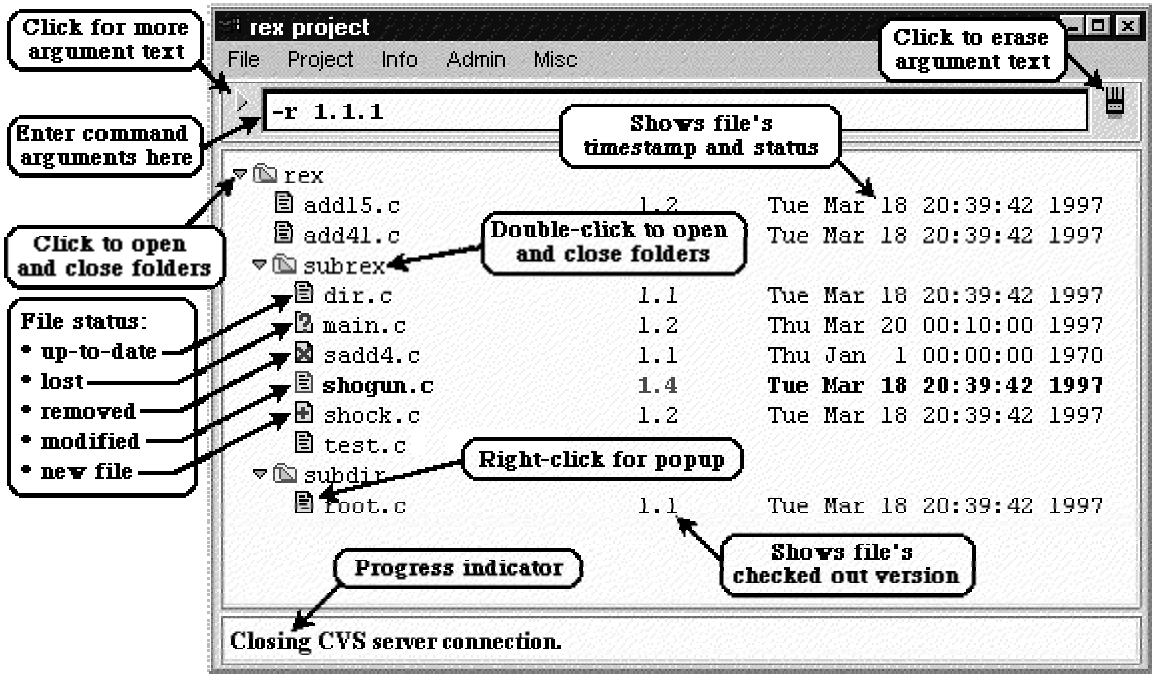
tkCVS

- Unix based tcl/tk based platform



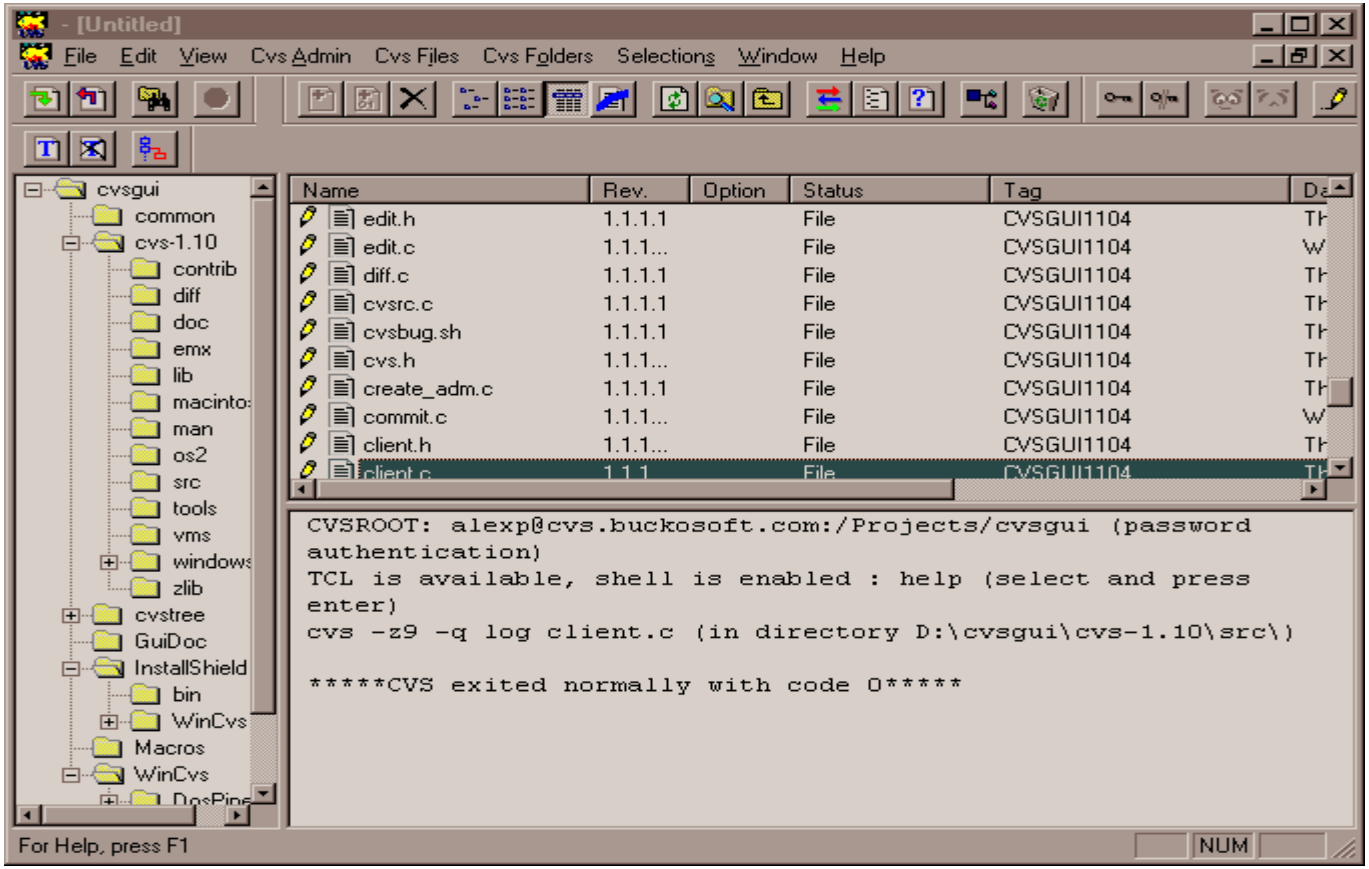
jCVS

- Any java capable platform



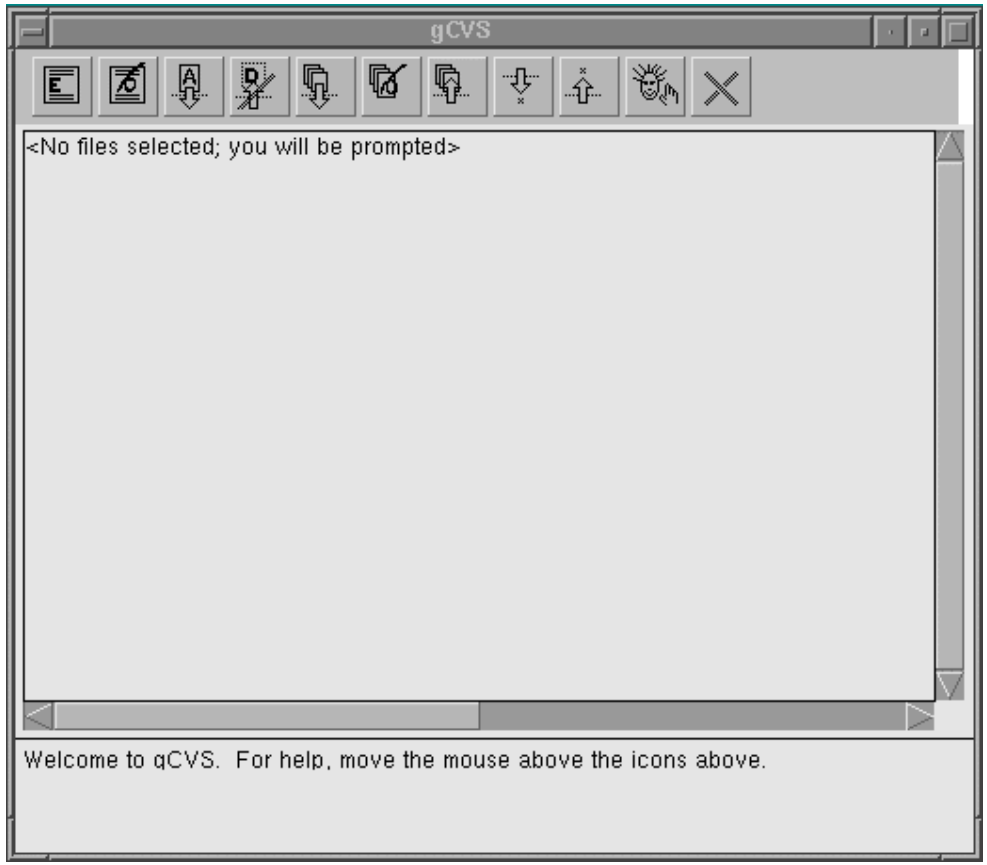
WinCVS

- Windows or Mac



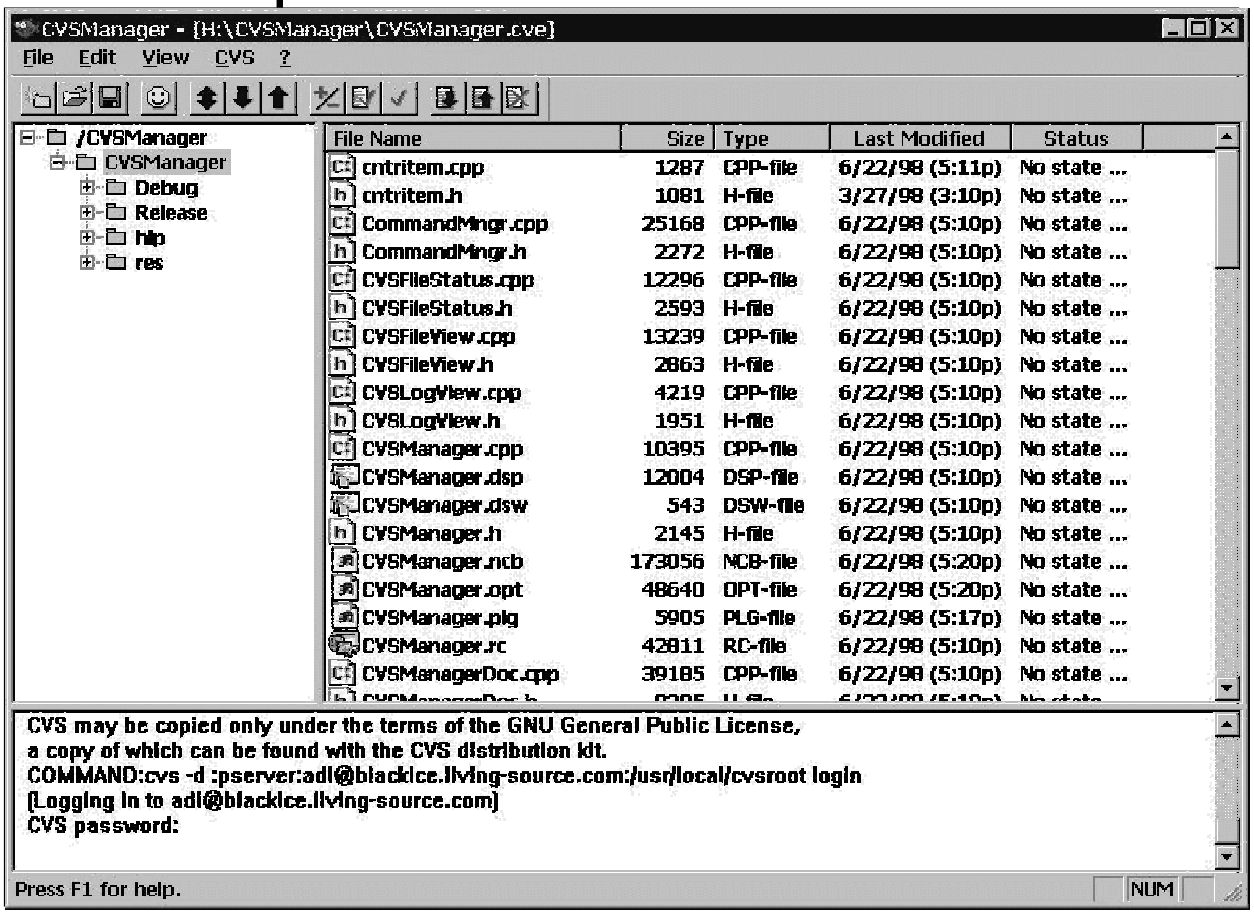
gCVS

- Unix platforms



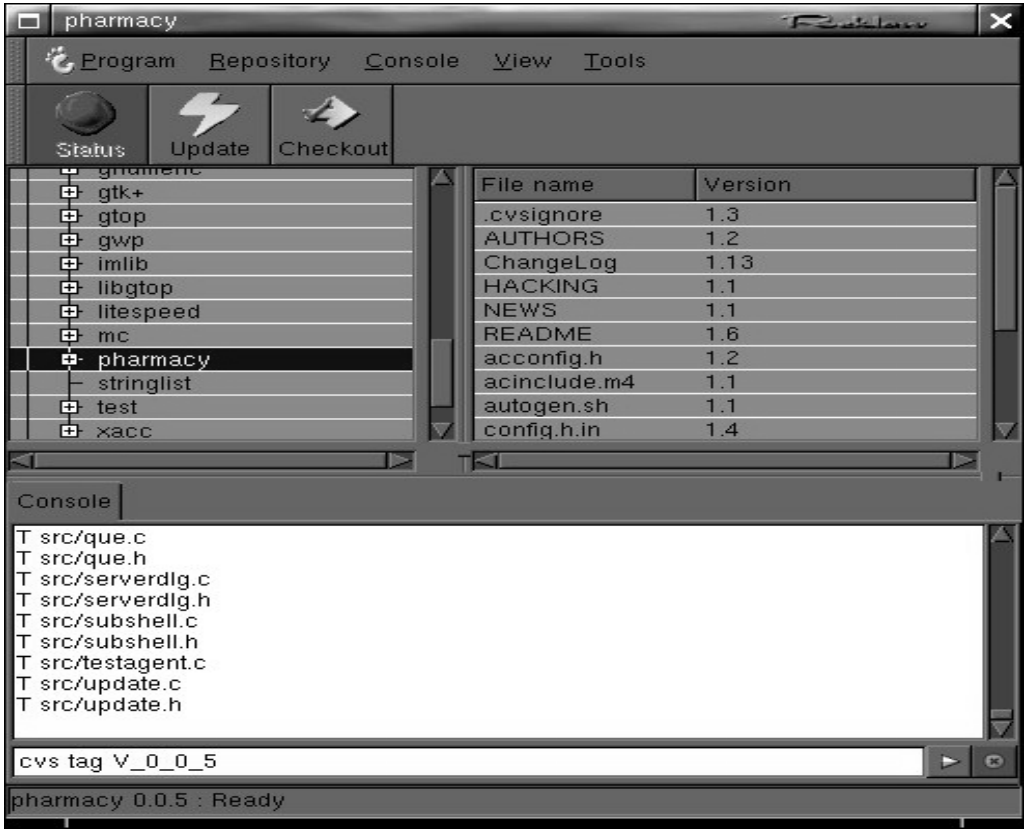
cvsmanager

- Windows platform



Pharmacy -- under construction

- Unix platform (Gnome)



Web interface

- CVSWeb
  - Browse versions
- bonsai
  - Query database
- faq-o-matic
  - Query database

Reference(s)

- www.cyclic.com