

本文内容为高索公司原创，如需转载，请保留此信息。

高索科技为您提供专业的 OpenSource 配置管理解决方案。

联系方式：E-Mail: [sz@coossa.com.cn](mailto:sz@coossa.com.cn) ; Tel: 0755-26638752 ; Fax: 0755-26638764

# CVS 服务器安装及使用简介

## 目 录

1 引言.....	2
2 几种版本维护工具的简介.....	2
2.1 ClearCase.....	2
2.2 Visual SourceSafe.....	2
2.3 CVS.....	2
3 CVS 服务器配置.....	3
3.1 ext: 和 :server: 方法.....	3
3.2 :pserver:方法.....	3
4、创建 CVS 数据仓库.....	4
5、客户端访问 CVS 数据仓库的方法.....	4
5.1 :ext:方式远程访问 CVS 数据仓库的客户端配置.....	4
5.2 :pserver:方式远程访问 CVS 数据仓库的客户端配置.....	5
6、使用 CVS 进行版本控制.....	5
6.1 项目经理确定基础版本.....	6
6.2 小组成员使用 cvs.....	6
7 . 结束语.....	7

# 1 引言

随着软件团队人员的增加，软件版本不断变化，时间的紧缺，多种平台的复杂环境，使得版本控制软件已成为当今软件开发人员（工程人员和管理者）所必须的工具。CVS 采用典型的 Client/Server 结构，开发团队能够跨越网络跨平台进行分布式操作。在软件开发环境中，CVS 自动追踪每一种对象类型（包括源代码、二进制文件、目录内容、可执行文件、文档、测试包、编译器、库文件）等变更情况，通过分支和归并功能支持并行开发。直接解决了原来开发团队所面临的难以处理的问题。

在各种具体方案应用之前，安装和配置 CVS 服务器，并学会简单的使用是第一要务。本篇就 CVS 的安装配置和简单使用做一介绍。在系列文档中，本篇为其他各文档的基础，主要针对完全没有接触过 CVS 但是希望了解和使用 CVS 的读者。

## 2 几种版本维护工具的简介

目前业界主流的版本维护工具有 ClearCase、Visual SourceSafe 以及本文将要详细介绍的 CVS，下面逐一进行简要介绍。

### 2.1 ClearCase

ClearCase 由 Rational 公司开发，它支持 UNIX 系统，混合平台开发，除了支持图形界面以外，还支持命令行格式的操作。它是一种功能强大、复杂的软件配置管理工具，版本控制是该软件的主要功能之一。ClearCase 包括两套：ClearCase LT 和 ClearCase (MultiSite)。前者可以用于在同一个局域网的开发小组，适合于中小型开发组织；ClearCase (MultiSite)则适应于分布于不同地理位置、不同局域网的开发小组，适合于大型的开发组织。ClearCase LT 由 ClearCase LT Server 和 ClearCase LT Client 两个部分组成，其中服务器部分负责数据的集中管理；客户端部分则安装在各个需要使用 ClearCase 服务的机器上，以及主要开发人员的计算机上，属于典型的 Client/Server 结构。

### 2.2 Visual SourceSafe

Visual SourceSafe 是微软公司开发的版本控制系统，可管理软件和 Web 站点的开发。Visual SourceSafe 可以处理由各种开发语言、创作工具或应用程序所创建的任何文件类型。与 CVS 相比，Visual SourceSafe 的最大特点是用户可以同时在文件和项目级进行工作。Visual SourceSafe 面向项目的特性能更有效地管理工作组应用程序开发或 Web 站点开发工作中的日常任务。

### 2.3 CVS

CVS 是自由软件社区最流行的版本控制系统，为 Netscape、\*BSD(FreeBSD,NetBSD 等)和许许多多大大小小的软件项目所采用。CVS 是并行控制系统的缩写，这是一个将一组文件放在层次目录树中以保持同步的系统。与前述的两个商业软件不同，CVS 是开放源代码的版本控制系统。但作为许多开放源码项目选用的版本管理软件，CVS 在版本控制方面与前两者相比并不逊色，可以说是各有千秋。在代码管理上，与上述的两种商业版本控

制软件的 unlock-modify-lock 的模式不同；CVS 采用的模式是 copy-modify-merge 的模式，在实际应用中，理解这一模式与其他模式的区别，才能更好的制定日常的维护和操作的准则。

## 3 CVS 服务器配置

早期的 CVS 被设计来为一组工作在同一台服务器上的开发人员使用。在文件系统上有一个 CVS 数据仓库，每位工作在某个项目上的开发人员工作需要对该数据仓库中的文件和 CVS 使用的辅助文件有读写权限。随着互联网的发展，远程访问数据仓库的需求产生了。当前，常用的远程访问 CVS 数据仓库的方法主要是如下两种：

### 3.1 ext: 和 :server: 方法

这一访问方法使用远程 shell RSH 或者其替代产品（如安全 shell 通道 SSH，仅适用于 :ext: 方法）。通过这种方式访问 CVS 数据仓库的开发人员必须在数据仓库所在的服务器上拥有一个系统帐号。此方式下，用户通过 SSH 服务连入服务器，然后相当于直接在服务器上执行 CVS 命令访问本地 CVS 数据仓库，因此，CVS 服务器上无需额外监听端口等待 CVS 客户端的连接。服务器上需要配置的是为每一位用户创建系统帐号，并创建 CVS 数据仓库（见 CVS 数据仓库创建小节）。

在我们常用的 Linux 发行套间 RedHat 中，SSH 服务器一般采用的是 OPENSSSH 软件，因此，在服务器上安装 OPENSSSH 软件，使得用户可以通过 SSH 访问服务器，就为访问 CVS 数据仓库做好了必要的网络设置。有关对 OPENSSSH 服务器的配置这里不做额外的说明，请参考其他相关文档或者参阅本文档系列内的《OpenSSH 服务器配置指南》。

### 3.2 :pserver: 方法

这一方法允许客户端通过用户名和密码验证的方式与服务器间建立连接。由于 CVS 本身没有提供 stand-alone 的独立监护进程模式，因此，通常采用在服务器上的通过精灵守护进程 inetd/xinetd 监听一个 TCP 端口（默认为 2401），然后在有连接建立时启动 cvs 命令处理连接后的数据通讯。所有客户端和服务器的通讯采用 CVS 的应用层协议。此方法不需要每位开发人员拥有服务器上的系统帐号（由 CVS 管理员需要仍然需要维护一个对数据仓库读写权限控制的列表文件）。这种方式下通讯中数据都是明文传输，包括 CVS 命令甚至是用户验证过程中的密码。

在我们常用的 Linux 发行套件 RedHat 中，使用的是 xinetd 作为守护进程，以下以 xinetd 为例简述启用 pserver 协议的方法。

以 root 用户登陆系统，在 /etc/xinetd.d/ 目录下创建一个 cvspserver 文件（xinetd.d 的服务配置文件，可任意定义服务名称），内容如下：

```
# default: off
# description: An cvs server.
service cvspserver
{
    disable = no
    socket_type = stream
```

```

        protocol = tcp
        user = root
        wait = no
        server = /usr/bin/cvs
        server_args = -f --allow-root=/cvsroot pserver
        log_on_success +=USERID
        log_on_failure += USERID
    }

```

然后修改/etc/services 文件，增加如下两行：

```

    cvspserver 2401/tcp      # CVS client/server operations
    cvspserver 2401/udp      # CVS client/server operations

```

然后执行命令 `service xinetd restart` 重新启动 xinetd 服务即可。

至此，xinetd 监听端口 2401，并在有客户端连接 2401 时，将所有来自网络的连接包转给/usr/bin/cvs 程序作为其标准输入，将 cvs 程序向标准输出输出的内容通过 socket 输出到客户端程序，建立客户端程序与 cvs 程序之间的 c/s 结构的数据通讯。我们为使用:pserver:方式远程访问 CVS 数据库做好了必要的配置。

## 4、创建 CVS 数据仓库

在服务器上建议一个系统帐户 cvsroot，主组为 cvs，每一位使用 cvs 的用户必须属于 CVS 组。以 cvsroot 登陆系统，并创建 CVS 数据仓库的目录如/cvsroot，然后修改该目录的权限，以使得 cvs 组内的用户对该木偶都具有读写权限。

```
chmod 770 /cvsroot
```

设置 CVS 数据仓库目录的 setgid 位以使得 cvs 组用户在该目录下创建的文件输入 cvs 组而不是该用户缺省的主组：

```
chmod g+s /cvsroot
```

然后使用如下命令初始化 CVS 数据仓库：

```
cvs -d /cvsroot init
```

## 5、客户端访问 CVS 数据仓库的方法

### 5.1 :ext:方式远程访问 CVS 数据仓库的客户端配置

- Linux/Unix 系统通过 CVS 客户端访问 CVS 数据仓库

首先设置环境参数 CVS\_RSH，在家目录的.bash\_profile 中增加如下行：

```
export CVS_RSH=/usr/bin/ssh
```

```
export CVSROOT=:ext:cvs@172.30.12.1:/cvsroot
```

然后 source .bash\_profile 或者从当前登陆会话中退出并重新登陆，执行 cvs 命令如：

```
cvs checkout CVSROOT
```

按提示输入 SSH 登陆所需的用户名和密码或者配置好 public key(Key Auth 方式登陆)，命令执行后执行结果如下：

```

cvs checkout: Updating CVSROOT
U CVSROOT/checkoutlist
U CVSROOT/commitinfo
U CVSROOT/config
U CVSROOT/cvsignore

```

```
U CVSROOT/cvswrappers
U CVSROOT/editinfo
U CVSROOT/loginfo
U CVSROOT/modules
U CVSROOT/notify
U CVSROOT/rcsinfo
U CVSROOT/taginfo
U CVSROOT/verifymsg
```

可见已经完成客户端的配置。用户可以使用其他 cvs 命令访问和操作服务器上的数据仓库了。与:pserver:不同，:ext:方式访问

- Windows 平台下通过 wincvs 远程访问 CVS 数据仓库  
参阅《Windows 平台上 CVS(SSH)客户端配置指南》

## 5.2 :pserver:方式远程访问 CVS 数据仓库的客户端配置

- Linux/Unix 系统  
首先设置环境参数 CVS\_RSH，在家目录的.bash\_profile 中增加如下行：  

```
export CVSROOT=:pserver:cvs@172.30.12.1:/cvsroot
```

然后 source .bash\_profile 或者从当前登陆会话中退出并重新登陆，执行 cvs 命令如：  

```
cvs login
```

按提示输入用户名和密码，然后执行命令：  

```
cvs checkout CVSROOT
```

命令执行结果与上述相同，可以使用其他 cvs 命令操作服务器上远程访问和操作 CVS 数据仓库了。
- Windows 平台 wincvs 客户端配置
  - ( 1 ) 一般选项的设置，选择 Admin->Preferences...，输入以下各项：
    - 第一、 Authentication：选择 cvs 服务器的认证方式为 pserver 方式。
    - 第二、 Path：用来配置 cvs 在服务器上的主目录路径，也就是服务器上用于进行 cvs 初始化的目录，这里为/home/cvsroot/repository；
    - 第三、 Host Address：配置 cvs 服务器所在服务器的地址，可以是 IP 地址，也可以是 DNS 名。在本次开发项目中使用 192.168.2.55；
    - 第四、 User name：配置要使用些 WinCVS 来登录 CVS 服务器的用户名 linyuan，该用户属于 cvs 用户组；
  - ( 2 ) 全局选项的设置：
    - 第一、 Checkout read-only 不选；
    - 第二、 Prune (remove) empty directories 也不要选上；
    - 第三、 对一般配置没有特殊要求的，把 Dirty files support、Supply control when adding 与 TCP/IP compression 选项选中；
  - ( 3 ) 登录服务器：  
选择 Admin->login，按对话框要求输入登录口令即可。

## 6、使用 CVS 进行版本控制

以下以 WinCVS 上的操作为例，简述在项目小组中使用 CVS 做版本管理的过程。

## 6.1 项目经理确定基础版本

在项目小组中，首先由项目经理确定软件开发的基础版本，选取建立好的项目所在文件夹，然后从菜单选择 Remote - > Import module - > continue - > 确定将项目文件检入( check in ) 到服务器并从菜单栏的 admin->>comandline，输入命令：cvs admin -l 版本号 文件全名将其锁定。随着开发计划和进度，由项目经理以相同方法定期公布最新的基础版本。同时项目经理或测试经理等小组成员也可以通过同样的方法发布项目编码规范等重要的团队开发信息。

## 6.2 小组成员使用 cvs

### 6.2.1 检出

小组成员现在就可以从 CVS 服务器上检出各自负责的模块进行进一步的开发了。在 workspace 中的 Modules 选中要存放 checkout 模块的目录，选择 Create->Checkout Modeles，分别输入要检出的文件的目录和工作目录。开发完后再把文件提交到 CVS 服务器。

### 6.2.2 提交新文件

在项目中如果有新的文件加入，要提交到服务端，应先选中文件，然后点击“添加按钮”，添加文件后，再在右键菜单中选择提交命令“Commit Selection”即可。

### 6.2.3 提交修改文件

有时小组的多个成员会对同一个文件进行修改，具体的有以下情况：

a)、只有一个小组成员对文件进行修改的情况

对 checkout 出来的文件进行修改，修改之后的文件在没有提交之前会是红色的，选中该的文件后右击选择“Commit Selection”选项，如果没有其它成员也对其进行修改并已经提交到 CVS 服务器上，一切正常的话将把该文件提交到 CVS 服务器并把图标恢复成原来的颜色。

b)、两个或两个以上的小组成员对同一文件的不同部分进行修改的情况

当两个小组成员都 checkout 了文件(假设文件名为 RecordSesClient.java)的情况下：

第一位成员修改首先提交到 CVS 服务器，一般将正常提交。

第二位成员修改后在第一位成员提交后再提交到 CVS 服务器时将提示：

```
cvs server: Up-to-date check failed for `RecordSesClient.java'
```

```
cvs [server aborted]: correct above errors first!
```

此时表明已经有成员对同一个文件进行修改并提交到 CVS 服务器，这时要先选择“Update Selection”对本地文件与 CVS 服务器上的文件进行同步与合并。所修改文件随后变成红色，合并后的文件是存放在该成员的本地机上，为了更新到 CVS 服务器还必须选中此文件并右击选择“Commit Selection”才能把修改合并后的结果提交到 CVS 服务器上。

c)、两个或两个以上的成员对同一个文件的相同部分进行修改的情况

当两个成员都下载了文件 RecordSesClient.java 并进行修改后先后提交，则第二个成



员客户端将显示如下的错误提示：

```
cvcs -z9 commit -m "update in 21:20" RecordSesClient.java (in directory C:\cvcsproject)
cvcs server: Up-to-date check failed for `RecordSesClient.java`
cvcs [server aborted]: correct above errors first!
***** CVS exited normally with code 1 *****
```

表明该成员的修改与其它成员的修改冲突，这时先选择“Update Selection”，将显示如下提示：

```
cvcs -z9 update RecordSesClient.java (in directory C:\cvcsproject)
RCS file: /home/cvcsroot/repository/CheckRecord/RecordSesClient.java,v
retrieving revision 1.9
retrieving revision 1.10
Merging differences between 1.9 and 1.10 into RecordSesClient.java
rcsmerge: warning: conflicts during merge
cvcs server: conflicts found in RecordSesClient.java
C RecordSesClient.java
***** CVS exited normally with code 0 *****
```

RecordSesClient.java 前面的 C 表示与其它成员的修改有冲突，并且文件的图标会加显示一个“C”。双击 RecordSesClient.java 将显示 RecordSesClient.java 的内容，可以看到 CVS 用明显的符号标志出成员修改有冲突的部分。这时就应该与其他成员协商决定最终版本后选择“Commit Selection”进行提交。

## 6.2.4 标记(tag)

标记是版本控制中很重要的功能。除了通常的作用，它还可以标记修改者，让大家知道，是谁提交了一个新版本。在 wincvs 中选择准备标记的文件，然后在菜单栏中选择 modify->>creat a tag on selection；马上 creat tag setting 窗体会出现，在 new tag name 中填入标记名即可。

## 6.2.5 分支管理

软件版本不断变化，而软件的缺陷不一定能够在一次升级就能发现。往往在第一个版本发现了 Bug，而升级版本已经有了好几个。因此就需要在发现 Bug 的版本上建立一个分支，在这个分支上进行开发，创建补丁。然后查看最新版本上又无此 bug；如果有，就将分支合并到主版本上，解决最新版本的问题。新建分支得实际操作如下：

在菜单栏的 creat->>creat a branch，在跳出的 creat branch setting 窗体上，new branch 中，填入分支名字；在 module to fork 中填入文件名；在 rtag options 选项卡中，选中 rev./tag/branch，填入版本号。

## 7. 结束语

利用 CVS，项目经理就能方便地协调小组，根据制定的代码管理规范，及时发布官方版本信息并充分利用标记和分支管理开发团队，能够极大提高团队地工作效率。