Preparing Linux for ClearCase Installation

A successful configuration of the vnode and MVFS modules

David M. Hyer
Technical Support Engineer 2
Escalation Engineer Tier 2
Linux SME

January 23, 2008

Introduction

The purpose of this Whitepaper is to give simple instructions so that anyone who needs to install IBM® Rational® ClearCase® on Linux®, RedHat® or SUSE®, will be able to successfully. This document is aimed at those with prior installation experience on other UNIX® platforms. The reader will be able to prepare the vnode/MVFS module for ClearCase installation. This is not an installation or troubleshooting guide.

This guide is organized into two sections with two subsections. The major sections are for the different distributions of Linux supported on ClearCase: RedHat and SUSE. The subsections are for the different kernel versions that each distribution puts out and that ClearCase supports.

Before proceeding you should review the ClearCase System Requirements technotes below for a current listing of supported operating systems, architectures, and required patches:

ClearCase 7.x System Requirements:
http://www.ibm.com/support/docview.wss?rs=984&uid=swg21239315

ClearCase 2003.06.00 System Requirements:
http://www.ibm.com/support/docview.wss?rs=984&uid=swg21136950

Note: ClearCase LT® (CCLT) does not use the MVFS, therefore these instructions do not apply to CCLT.

Linux 2.4.x Kernels

This section provides the installation preparation instructions for RedHat and SUSE 2.4.x kernels.

RedHat 2.4.x Kernel preparation

Preparation of the Vnode Layer for IBM Rational ClearCase Install or Relink

If you upgrade the Linux kernel from an officially sanctioned default kernel from RedHat or SUSE, make sure to get the kernel sources as well. If you recompile the kernel to take advantage of some new feature for hardware, software, then you will need to replay the preparation steps for creating a new ".config".

IMPORTANT: *Log in as root*. This is necessary for all of the following procedures. None of these steps will modify or taint the kernel. The source files are only necessary in order to successfully build the correctly configured vnode layer and subsequent module, ensuring the kernel symbols match exactly.

First and foremost, you will need to verify that the kernel source files are installed on the Linux system and match what is currently the loaded kernel. You can use either the *uname –a* command or *cat /proc/version*. This can be accomplished one of two ways:

A. cd into */usr/src* and run **ls -l**.
You should see something similar to the following sample directories and links:

On Red Hat 2.4.x kernel systems

```
drwxr-xr-x 2 root root 4096 Jan 24 2003 debug
lrwxrwxrwx 1 root root 14 Aug 24 17:11 linux -> linux-2.4
lrwxrwxrwx 1 root root 14 Aug 24 17:10 linux-2.4 -> linux-2.4.20-8
drwxr-xr-x 16 root root 4096 Aug 25 12:17 linux-2.4.20-8
drwxr-xr-x 7 root root 4096 Aug 23 19:40 redhat
```

B. Or you can run the command:

rpm -qa | grep kernel

You should see something similar to this:

```
[root]# rpm -qa | grep kernel
kernel-doc-2.4.21-9.EL.sgi1
kernel-source-2.4.21-4.EL
kernel-2.4.21-4.EL
kernel-utils-2.4-8.37
kernel-pcmcia-cs-3.1.31-13
```

You may see more output depending on how or what you installed. The things you need to see are the **kernel-#** and **kernel-source-#**.

Red Hat 2.4.x kernel systems

1. Verify that the source directory is there

   a. cd */usr/src/*

   b. then run **ls -l**

   c. There is a link to either a 'linux' directory or a shortened directory such as 'linux-2.4', you should see output similar to the example below:

```
drwxr-xr-x 2 root root 4096 Jan 24 2003 debug
lrwxrwxrwx 1 root root 14 Aug 24 17:10 linux-2.4 -> linux-
2.4.20-8
drwxr-xr-x 16 root root 4096 Aug 25 12:17 linux-2.4.20-8
drwxr-xr-x 7 root root 4096 Aug 23 19:40 redhat
```

The link of linux-2.4 to your kernel source directory is created upon install of the operating system. IBM Rational ClearCase scripts typically refer to the older source directory of */usr/src/linux*. Step 2 will show you how to create this new link.

2. Create link of kernel source directory link to linux link

   a. Run **ln -s linux-2.4 linux**

   b. Once you have created the new link you should see it in the directory listing:

```
ls -l

drwxr-xr-x 2 root root 4096 Jan 24 2003 debug
lrwxrwxrwx 1 root root 14 Aug 24 17:11 linux -> linux-2.4
lrwxrwxrwx 1 root root 14 Aug 24 17:10 linux-2.4 -> linux-
2.4.20-8
drwxr-xr-x 16 root root 4096 Aug 25 12:17 linux-2.4.20-8
drwxr-xr-x 7 root root 4096 Aug 23 19:40 redhat
```

3. cd linux-2.4 or linux

4. Remove any old configuration file (.config) that may have previously existed:

   make mrproper

5. Build a ".config" file to represent what is configured in your current kernel setup so the vnode can be built correctly with matching symbols.

There are two ways to get the .config file into the */usr/src/linux* directory:

- Copy the config file from the */boot* directory

  On a Red Hat 2.4.x kernel you can cd to */boot* (may not be mounted on some systems):

  A. cd to */boot*

  B. run **ls –l** to view the contents of the directory.

  C. look for a "config-{kernel#}" file (for example, "config-2.4.21-4.EL")

  D. copy this file to the top-level of the linux kernel source directory (*/usr/src/linux*, for example) and rename it to **.config**

     Example…

     ```
     cp config-2.4.21-4.EL /usr/src/linux-2.4/.config
     ```

- Compile one from the stock versions that are already installed.

  a. cd into the "configs" directory (*/usr/src/linux/configs*, for example).

     You will find output similiar to the following:

     ```
     [root]$ ls
     kernel-2.4.20-athlon.config kernel-2.4.20-i586-smp.config
     kernel-2.4.20-athlon-smp.config kernel-2.4.20-i686-
     bigmem.config
     kernel-2.4.20-i386-BOOT.config kernel-2.4.20-i686.config
     kernel-2.4.20-i386.config kernel-2.4.20-i686-smp.config
     kernel-2.4.20-i386-smp.config kernel-2.4.20-x86_64.config
     kernel-2.4.20-i586.config kernel-2.4.20-x86_64-smp.config
     ```

  b. Choose the config file that best represents your server hardware setup and copy it up one directory as **.config** (running the command "uname –a" can help determine this)

     Example...

     ```
     cp kernel-2.4.20-i686.config ../.config
     ```

  c. Then you will need to go back up to the source directory */usr/src/linux* by running the command

     cd ..

     to confirm '.config' is there

     ls –al should show you something similar to the following:

```
[root]# ls -al
total 1092
drwxr-xr-x  16    root  root  4096 Dec 31 10:54 .
drwxr-xr-x  4     root  root  4096 Dec 31 10:52 ..
drwxr-xr-x  19    root  root  4096 Dec 31 10:52 arch
-rw-r--r--  1     root  root  48628 Dec 31 10:54 .config
drwxr-xr-x  2     root  root  4096 Dec 31 10:52 configs
-rw-r--r--  1     root  root  18691 Aug  2  2002 COPYING
-rw-r--r--  1     root  root  29940 Sep  7  2005 COPYING.modules
-rw-r--r--  1     root  root  80764 Sep  7  2005 CREDITS
drwxr-xr-x  2     root  root  4096 Dec 31 10:52 crypto
-rw-r--r--  1     root  root  5073 Sep  7  2005 .depend
drwxr-xr-x  31    root  root  4096 Dec 31 10:52 Documentation
drwxr-xr-x  46    root  root  4096 Dec 31 10:52 drivers
drwxr-xr-x  48    root  root  4096 Dec 31 10:52 fs
-rw-r--r--  1     root  root  234728 Sep  7  2005 .hdepend
drwxr-xr-x  27    root  root  4096 Dec 31 10:52 include
drwxr-xr-x  2     root  root  4096 Dec 31 10:52 init
drwxr-xr-x  2     root  root  4096 Dec 31 10:52 ipc
drwxr-xr-x  2     root  root  4096 Dec 31 10:52 kernel
drwxr-xr-x  4     root  root  4096 Dec 31 10:52 lib
-rw-r--r--  1     root  root  45547 Sep  7  2005 MAINTAINERS
-rw-r--r--  1     root  root  20799 Sep  7  2005 Makefile
drwxr-xr-x  2     root  root  4096 Dec 31 10:52 mm
drwxr-xr-x  32    root  root  4096 Dec 31 10:52 net
-rw-r--r--  1     root  root  14239 Aug  2  2002 README
-rw-r--r--  1     root  root  2818 Jun 13  2003 REPORTING-BUGS
-rw-r--r--  1     root  root  9528 Sep  7  2005 Rules.make
drwxr-xr-x  4     root  root  4096 Dec 31 10:52 scripts
-rw-r--r--  1     root  root  503380 Sep  7  2005 tmp_include_depends
```

6. This next command strips down the original config file to what is actually configured for your system.

make oldconfig

7. The following makes and checks for the dependencies the vnode needs to be aware of.

This may take some time while it checks for and resolves any dependencies based on kernel settings.

make dep

If you did not encounter any errors during the above steps, you are ready to Install ClearCase.

SUSE 2.4.x Kernel Preparation

If you are using a SUSE 2.4.x kernel, and can cd to */boot* (may not be mounted on some systems).

1. cd to */boot*
2. Run **ls –l** to view the contents of the directory.
3. Look for the **vmlinuz.config** file.
4. Copy this file to the top-level of the linux kernel source directory (*/usr/src/linux-2.4*, for example) as **.config**

   Example:

   ```
   cp vmlinuz.config /usr/src/linux-2.4/.config
   ```

5. cd into the “arch” directory (*/usr/src/linux-2.4/arch*, for example).

   You will see for example:

   ```
   [root] # ls -al

   total 5

   drwxr-xr-x 20  root  root 480 Jan 9 2004 .
   drwxr-xr-x 17  root  root 744 Nov 20 01:24 ..
   drwxr-xr-x 7   root  root 288 Jan 9 2004 alpha
   drwxr-xr-x 18  root  root 680 Jan 9 2004 arm
   drwxr-xr-x 7   root  root 312 Jan 9 2004 cris
   drwxr-xr-x 8   root  root 312 Jan 9 2004 i386
   drwxr-xr-x 12  root  root 408 Jan 9 2004 ia64
   drwxr-xr-x 20  root  root 632 Jan 9 2004 m68k
   drwxr-xr-x 29  root  root 1560 Jan 9 2004 mips
   drwxr-xr-x 11  root  root 624 Jan 9 2004 mips64
   drwxr-xr-x 7   root  root 288 Jan 9 2004 parisc
   drwxr-xr-x 12  root  root 408 Jan 9 2004 ppc
   drwxr-xr-x 9   root  root 336 Jan 9 2004 ppc64
   drwxr-xr-x 7   root  root 328 Jan 9 2004 s390
   drwxr-xr-x 6   root  root 304 Jan 9 2004 s390x
   drwxr-xr-x 7   root  root 288 Jan 9 2004 sh
   drwxr-xr-x 8   root  root 312 Jan 9 2004 sparc
   drwxr-xr-x 9   root  root 336 Jan 9 2004 sparc64
   drwxr-xr-x 13  root  root 752 Jan 9 2004 um
   drwxr-xr-x 9   root  root 336 Jan 9 2004 x86_64
   ```

6. Choose the directory that matches the architecture that you are using (such as *i386*), then cd into it:

   ```
   [root]# cd i386
   ```

```
[root] # ls -al

total 76

drwxr-xr-x 8   root  root 312 Jan 9 2004 .
drwxr-xr-x 20  root  root 480 Jan 9 2004 ..
-rw-r--r-- 1   root  root 3797 Oct 21 2002 Makefile
drwxr-xr-x 4   root  root 240 Jan 9 2004 boot
-rw-r--r-- 1   root  root 15363 Oct 21 2002 config.in
-rw-r--r-- 1   root  root 46245 Oct 21 2002 defconfig
drwxr-xr-x 2   root  root 352 Jan 9 2004 kdb
drwxr-xr-x 2   root  root 1496 Nov 20 01:28 kernel
drwxr-xr-x 2   root  root 384 Nov 20 01:28 lib
drwxr-xr-x 2   root  root 1520 Jan 9 2004 math-emu
drwxr-xr-x 2   root  root 240 Nov 20 01:28 mm
-rw-r--r-- 1   root  root 1909 Oct 21 2002 vmlinux.lds.S
```

7. Copy the defconfig file to the top-level kernel source directory as .config

```
cp defconfig ../../.config
```

8. cd ../..

9. Confirm **.config** is there

```
# ls -al
total 262
drwxr-xr-x 17  root  root 664 Jan  8 16:25 .
drwxr-xr-x 4   root  root 136 Apr 21  2003 ..
-rw-r--r-- 1   root  root 46245 Jan  8 16:25 .config
-rw-r--r-- 1   root  root 11 Oct 21  2002 .kversion
-rw-r--r-- 1   root  root 18691 Aug  2  2002 COPYING
-rw-r--r-- 1   root  root 79407 Oct 21  2002 CREDITS
-rw-r--r-- 1   root  root 22315 Oct 21  2002 Makefile
-rw-r--r-- 1   root  root 14239 Aug  2  2002 README
-rw-r--r-- 1   root  root 9601 Oct 21  2002 Rules.make
drwxr-xr-x 11  root  root 320 Apr 21  2003 abi
drwxr-xr-x 20  root  root 480 Apr 21  2003 arch
drwxr-xr-x 45  root  root 1128 Apr 21  2003 drivers
drwxr-xr-x 49  root  root 2488 Apr 21  2003 fs
drwxr-xr-x 30  root  root 816 Apr 21  2003 include
drwxr-xr-x 2   root  root 168 Apr 21  2003 init
drwxr-xr-x 2   root  root 192 Apr 21  2003 ipc
drwxr-xr-x 3   root  root 312 Apr 21  2003 kdb
drwxr-xr-x 2   root  root 968 Apr 21  2003 kernel
drwxr-xr-x 2   root  root 408 Apr 21  2003 lib
drwxr-xr-x 2   root  root 696 Apr 21  2003 mm
drwxr-xr-x 29  root  root 912 Apr 21  2003 net
drwxr-xr-x 4   root  root 952 Apr 21  2003 scripts
```

Linux 2.6.x Kernels

This section provides the installation preparations instructions for RedHat and SUSE 2.6.x kernels

RedHat 2.6.x kernel Preparation

On Red Hat Enterprise 4 2.6.x kernel systems (RHEL4)

In the newer versions of RedHat Enterprise Linux (RHEL) 4, preparing the vnode or MVFS (for version 7.0 of ClearCase) is much simpler. You will not see the directory structure found in the 2.4.x preparation section. Instead you will see a directory structure similar to the following:

```
[root]# pwd
/usr/src

[root]# ls -al
total 16
drwxr-xr-x  7 root root 4096 Feb  9 19:30 kernels
drwxr-xr-x  7 root root 4096 May 30  2006 redhat
```

From here you should cd into the kernels directory, (cd kernels) where you will be presented (based on your installation of the OS) with a list of kernels to boot from or modify for your use:

```
[root]# pwd
/usr/src/kernels

[root]# ls -al
total 24
drwxr-xr-x  17  root    root 4096 Sep 20  2006 2.6.9-42.0.2.EL-i686
drwxr-xr-x  17  root    root 4096 Feb  9 19:23 2.6.9-42.0.8.EL-
hugemem-i686
drwxr-xr-x  17  root    root 4096 Feb  9 19:24 2.6.9-42.0.8.EL-i686
drwxr-xr-x  17  root    root 4096 Feb  9 19:22 2.6.9-42.0.8.EL-smp-
i686
drwxr-xr-x  18  root    root 4096 Aug  1  2006 2.6.9-5.EL-i686
```

The above directory listing shows that this system has more than 1 kernel. This is typically found in a system where the kernel was upgraded from one update level to another and the kernel was included in that update.

At this point you need to pick the directory that represents your current configuration. In this case 2.6.9-42.0.2.EL-i686

cd 2.6.9-42.0.2.EL-i686

This is a difference between 2.4.x kernels and 2.6.x kernels:

```
[root]# ls -al
total 352
drwxr-xr-x  17  root    root   4096 Sep 20  2006 .
drwxr-xr-x   7   root   root   4096 Feb  9 19:30 ..
-rw-r--r--   1       root     root  50337 Aug 17  2006 .config
-rw-r--r--   1     root       root  42215 Aug 17  2006 Makefile
-rw-r--r--   2       root     root 179585 Aug 17  2006 Module.symvers
drwxr-xr-x  24  root    root   4096 Sep  8  2006 arch
drwxr-xr-x   4   root   root   4096 Sep  8  2006 crypto
drwxr-xr-x  48  root    root   4096 Sep  8  2006 drivers
drwxr-xr-x  56  root    root   4096 Sep  8  2006 fs
drwxr-xr-x  15  root    root   4096 Sep  8  2006 include
drwxr-xr-x   2   root   root   4096 Sep  8  2006 init
drwxr-xr-x   2   root   root   4096 Sep  8  2006 ipc
drwxr-xr-x   3   root   root   4096 Sep  8  2006 kernel
drwxr-xr-x   4   root   root   4096 Sep  8  2006 lib
drwxr-xr-x   2   root   root   4096 Sep  8  2006 mm
drwxr-xr-x  33  root    root   4096 Sep  8  2006 net
drwxr-xr-x  10  root    root   4096 Sep  8  2006 scripts
drwxr-xr-x   4   root   root   4096 Sep  8  2006 security
drwxr-xr-x  15  root    root   4096 Sep  8  2006 sound
drwxr-xr-x   2  root    root   4096 Sep  8  2006 usr
```

There is already a .config file defined for you. As long as you accept the kernel in its present state, that is to modify and rebuild it, then you can use this .config to install ClearCase.

The new operating system by Red Hat is based off of the 2.6.x kernel. As of this whitepaper revision release, IBM Rational ClearCase supports the base version of RHEL4 up to update 5. Support is offered starting with patches 42 (user space) and 43 (MVFS) for version 2002.05, and starting with patches 20 (MVFS) and 21 (User Space) for version 2003.06. No patches are need for version 7.0.0.0.

There are three options to consider while installing ClearCase:

1. During the installation of ClearCase, when asked to relink the MVFS, answer NO.
This will tell ClearCase to load the precompiled modules that ship with the patches listed above. This process should work fine. If this fails, perform option 3 below.

2. During the installation of ClearCase, when asked to relink the MVFS, answer YES. This will tell ClearCase to recompile MVFS/vnode modules that ship with the patches listed above. Again, this should work fine. If this fails, perform option 3.

3. During the installation of ClearCase, choose to rebuild the MVFS, and when asked for the kernel source directory, input the following:

/lib/modules/{uname –r}/build or /usr/src/kernels/2.6.9-42.0.2.EL-i686

The first path is actually a link as shown:

```
[root]# pwd
/lib/modules/2.6.9-42.0.8.EL

[root]# ls -al
total 636
lrwxrwxrwx  1 root root     37 Feb  9 19:23 build ->
/usr/src/kernels/2.6.9-42.0.8.EL-i686
drwxr-xr-x  9     root  root   4096 Feb  9 19:16 kernel
drwxr-xr-x  2     root  root   4096 Feb 27 14:59 misc
-rw-rw-r--  1     root  root 119499 Apr 24 18:11 modules.alias
-rw-rw-r--  1     root  root     69 Apr 24 18:11 modules.ccwmap
-rw-rw-r--  1     root  root 138961 Apr 24 18:11 modules.dep
-rw-rw-r--  1     root  root     73 Apr 24 18:11 modules.ieee1394map
-rw-rw-r--  1     root  root    357 Apr 24 18:11 modules.inputmap
-rw-rw-r--  1     root  root    235 Apr 24 18:11 modules.isapnpmap
-rw-rw-r--  1     root  root 114486 Apr 24 18:11 modules.pcimap
-rw-rw-r--  1     root  root  67969 Apr 24 18:11 modules.symbols
-rw-rw-r--  1     root  root 156211 Apr 24 18:11 modules.usbmap
lrwxrwxrwx  1 root      root      5 Feb  9 19:23 source -> build
```

Either of these paths can be used. Using the **/lib** path is similar to using **/usr/src/linux** in that it normally points to the currently running kernel. In the case where you have many kernels for testing or development purposes, you may want to consider the **/usr/src** path.

Now you are ready to install ClearCase. Follow the *Installation Guide for ClearCase*. If you run into any errors or problems, you should contact Rational Support.

Typically, you will not see the unresolved symbols as the kernel sources are included in a default install of the operating system. If you do see unresolved symbols, verify that you have installed the kernel sources for your kernel. If you have not, you should install them and retry the install. If you have re-installed and still see them, contact Rational Support.

SUSE 2.6.x kernel Preparation

In the newer versions of SUSE Linux Enterprise Server (SLES) 9 and 10, preparing the vnode or mvfs (for version 7.0 of ClearCase) is much simpler. You will not see the directory structure found in the 2.4 preparation section. Instead you will see a directory structure similar to the following:

```
lrwxrwxrwx 1 root root 16 2004-11-01 16:52 linux -> linux-2.6.5-7.97
drwxr-xr-x 20 root root 4096 2004-11-02 15:36 linux-2.6.5-7.97
drwxr-xr-x 3 root root 4096 2004-11-01 12:02 linux-2.6.5-7.97-obj
lrwxrwxrwx 1 root root 20 2004-11-01 16:52 linux-obj -> linux-2.6.5-
7.97-obj
drwxr-xr-x 7 root root 4096 2004-11-01 16:24 packages
```

On SUSE, the linux source directory is not the intended directory to use, but rather the linux-{kernel#}-obj directory.

Verify that the kernel object source directory is there (ls -l) and that there is a link to either a 'linux' directory or a shortened directory such as 'linux-2.4'. You should see output similar to the example below:

1. cd /usr/src/

```
lrwxrwxrwx 1 root root 16 2004-11-01 16:52 linux -> linux-2.6.5-7.97
drwxr-xr-x 20 root root 4096 2004-11-02 15:36 linux-2.6.5-7.97
drwxr-xr-x 3 root root 4096 2004-11-01 12:02 linux-2.6.5-7.97-obj
lrwxrwxrwx 1 root root 20 2004-11-01 16:52 linux-obj -> linux-2.6.5-
7.97-obj
drwxr-xr-x 7 root root 4096 2004-11-01 16:24 packages
```

2. cd to linux-obj

```
[root]# cd linux-obj/
[root]# ls -al
total 4
drwxr-xr-x 6 root root 4096 2004-11-01 12:02 i386
```

Then, cd into the i386 directory. You will something similar to the following example:

```
[root]#  cd i386/

[root]# ls -al
total 16
drwxr-xr-x 6 root root 4096 2004-11-01 12:02 bigsmp
drwxr-xr-x 6 root root 4096 2004-11-01 12:02 debug
drwxr-xr-x 6 root root 4096 2004-11-04 18:38 default
drwxr-xr-x 6 root root 4096 2004-11-01 12:02 smp
```

Here, you would choose either the **smp** or **default** directories, depending on your current configuration (from kernel# of uname –a output).

Next, cd into the directory you chose.
For this example, we will assume a default configuration:

```
[root]# cd default

[root]# ls -al
total 156
drwxr-xr-x 6 root root 4096 2004-11-04 18:38 .
drwxr-xr-x 6 root root 4096 2004-11-01 12:02 ..
drwxr-xr-x 3 root root 4096 2004-11-01 12:02 arch
-rw-r--r-- 1 root root 55534 2004-11-04 18:38 .config
-rw-r--r-- 1 root root 4886 2004-11-04 18:38 .config.cmd
-rw-r--r-- 1 root root 55534 2004-07-02 10:45 .config.old
drwxr-xr-x 4 root root 4096 2004-11-01 12:02 include
drwxr-xr-x 2 root root 4096 2004-11-01 12:02 include2
-rw-r--r-- 1 root root 321 2004-07-02 10:45 Makefile
drwxr-xr-x 5 root root 4096 2004-11-01 12:02 scripts
```

There is already a .config file defined for you. As long as you accept the kernel in its present state, that is to modify and rebuild it, then you can use this .config to install ClearCase.

ClearCase Installation

Refer to the *IBM Rational ClearCase Installation Guide* for installation procedures. If you encounter problems or have any questions, contact Rational ClearCase Support.

If you apply any ClearCase patches to the release area, you should follow the instructions for that process found in the patch bundles. Then you will have to rebuild the vnode as explained in the **README** found in the `/var/adm/rational/clearcase/mvfs/mvfs_src` directory or follow Section 2 again.

Kernel Upgrade

If you upgrade to a supported kernel as instructed in the Support Policy for ClearCase on Linux document, you will need to redo the preparation steps described in the section 2.

References

Credits

Author: Dave Hyer (Rational ClearCase Level 2 Support Engineer, Tier 2 Escalation Engineer and Linux SME)

Special Thanks: Kevin Cool, William Penny, Rich Rakich and Keith Lefrancois

The following were used in references or as other sources of information:

- ClearCase 7.x System Requirements
  http://www.ibm.com/support/docview.wss?rs=984&uid=swg21239315
- ClearCase 2003.06.00 System Requirements
  http://www.ibm.com/support/docview.wss?rs=984&uid=swg21136950
- Support Policy for ClearCase on Linux
  http://www.ibm.com/support/docview.wss?rs=984&uid=swg21159882