Enabling ReiserFS, XFS, JFS on RedHat Enterprise Linux

Despite the Linux kernel having support for so many file systems, not all of them are enabled in RedHat Enterprise Linux by default. This might be well as some of them might not yet classify as "enterprise grade" in the eyes of RedHat, who knows... Luckily, support for missing file systems such as ReiserFS, XFS and JFS can be added easily as outlined below.

For this howto I assume you are running the stock RedHat kernel.
Get an RPM repository
First of all, you need access to an RPM repository. Usually this is your setup cd-rom or a network-accessible (FTP, HTTP or even NFS) directory. I assume you have this setup already, so let's proceed to the next step.
Install requireds packages
You'll need the toolchain, rpm-build, the glibc and kernel headers as well as the kernel source RPM from redhat. Issue this to see what kernel header version is required for you:

```
# uname -a
Linux localhost 2.6.18-53.el5 #1 SMP Wed Oct 10 16:34:19 EDT 2007 x86_64 x86_64 x86_64 GNU/Linux
```

Install them as required using yum:

```
#yum install rpm-build.x86_64 ncurses-devel.x86_64 gcc.x86_64 redhat-rpm-config unifdef
[ ... output omitted ... ]
================================================================================
 Package              Arch      Version         Repository      Size
================================================================================
Installing:
 gcc                  x86_64    4.1.2-14.el5    base            5.3 M
 ncurses-devel        x86_64    5.5-24.20060715 base            1.7 M
 redhat-rpm-config    noarch    8.0.45-22.el5   base             53 k
 unifdef              x86_64    1.171-5.fc6     base             15 k
Installing for dependencies:
 cpp                  x86_64    4.1.2-14.el5    base            2.9 M
 glibc-devel          x86_64    2.5-18          base            2.4 M
 glibc-headers        x86_64    2.5-18          base            598 k
 kernel-headers       x86_64    2.6.18-53.el5   base            814 k
 libgomp              x86_64    4.1.2-14.el5    base             77 k

Transaction Summary
================================================================================
Install      9 Package(s)
Update       0 Package(s)
Remove       0 Package(s)

Total download size: 14 M
Is this ok [y/N]:
```

This will install everything you need to get going in the first place. Now go and get the kernel source RPM, which you'll find at the RedHat FTP Site. Save it to a temporary directory and install it accordingly:

```
# mkdir /usr/src/sources && cd /usr/src/sources
# wget ftp://ftp.redhat.com/pub/redhat/linux/enterprise/5Server/en/os/SRPMS/kernel-2.6.18-53.1.6.el5.src.rpm
# rpm -ivh kernel-2.6.18-53.1.6.el5.src.rpm
```

This will throw some messages at you about a user/group named "brewbuilder" not being there. That means no harm, however you may prefer to create it first. Now everything needed should exist inside your /usr/src/redhat directory. Then run rpmbuild from the redhat tree.

```
# cd /usr/src/redhat
# rpmbuild -bp SPECS/kernel-2.6.spec
[ ... output omitted ... ]
```

Enable the file systems

Now you need to initialize the kernel build environment like this from the kernel source tree. You will also need to copy the original kernel config from your current RedHat kernel.

```
# cd /usr/src/redhat/BUILD/kernel-2.6.18/linux-2.6.18.x86_64
# cp /boot/config-`uname -r` .config
# make menuconfig
```

From "menuconfig" navigate to "File systems", where you enable the missing modules. In my case I select "Reiserfs support", "JFS" and "XFS" to be included as module ("M" marking). This is especially required if you just want to build the module and copy it over to your locally installed kernel modules directory. I'd recommend to create an RPM for proper upgrade management anyway, however in this case it actually doesn't matter (except in terms of overhead and performance of course) it you're using a module or compile it into the kernel. Afterwards I exit from "menuconfig", not without saving the changes of course.

Compiling and installing the modules the lazy way

Now this is what I call the lazy way... For easy upgrades and package management, I'd strongly recommend you create a fullblown RPM (see next section). However, if you just want to get going, you can compile the modules manually like this:

```
# cd /usr/src/redhat/BUILD/kernel-2.6.18/linux-2.6.18.x86_64
# mkdir .tmp_versions
# make fs/xfs/xfs.ko
# make fs/jfs/jfs.ko
# make fs/reiserfs/reiserfs.ko
```

Then add the modules made to your current module directory like this:

```
# mkdir /lib/modules/`uname -r`/kernel/fs/reiserfs
# mkdir /lib/modules/`uname -r`/kernel/fs/xfs
# mkdir /lib/modules/`uname -r`/kernel/fs/jfs
# cp ./fs/reiserfs/reiserfs.ko /lib/modules/`uname -r`/kernel/fs/reiserfs
# cp ./fs/xfs/xfs.ko /lib/modules/`uname -r`/kernel/fs/xfs
# cp ./fs/jfs/jfs.ko /lib/modules/`uname -r`/kernel/fs/jfs
# depmod -a
```

This will leave you with modules suiting your kernel. However, whenever you're upgrading the kernel package, they may get overwritten, so it's best to create a kernel rpm and install it as a regurlar package.

Building a full-featured RPM

Before you create your RPM package, it's recommended you edit the Makefile and replace the EXTRAVERSION header by something meaningful which makes the difference clear.

```
# cd /usr/src/redhat/BUILD/kernel-2.6.18/linux-2.6.18.x86_64
# vi Makefile
```

In this case, I've set **EXTRAVERSION = -jfsxfsreiserfs_2.6.18_53_1.6**. Then simply run this command to create the RPM package:

```
# cd /usr/src/redhat/BUILD/kernel-2.6.18/linux-2.6.18.x86_64
# make rpm
```

Like this you'll end up with both a new RPM package and a source RPM inside your /usr/src/redhat/RPMS and /usr/src/redhat/SRPMS directories, which can then be installed via **rpm** or, if integrated with a local yum repository, from **yum** itself.

Getting the tools

Of course, the modules themselves serve no practical purpose if you lack the userspace tools to create and maintain the file systems. Get them accordingly from ftp://oss.sgi.com/projects/xfs/ (XFS), http://jfs.sourceforge.net/ (JFS) and http://chichkin_i.zelnet.ru/namesys/ (ReiserFS). Again, it's recommended to create an RPM from the sources, but this is beyond the scope of this article.