Software Developer Program

Application Build Environment Manual

Software Developer Program: Application Build Environment Manual


Red Hat, Inc.

1801 Varsity Drive
Raleigh NC 27606-2072 USA
Phone: +1 919 754 3700
Phone: 888 733 4281
Fax: +1 919 754 3701
PO Box 13588
Research Triangle Park NC 27709 USA

ABE Manual(EN)-1-Print-RHI (2004-12-22)

The GPG fingerprint of the security@redhat.com key is:
CA 20 86 86 2B D6 9D FC 65 F6 EC C4 21 91 80 CD DB 42 A6 0E

Table of Contents

Chapter 1.

Application Build Environment (ABE) Overview

The Application Build Environment (ABE) is a component of the Software Developer Program. The ABE enables you to create multiple *environments* on a single computer, where an environment is a previous Red Hat Enterprise Linux release/update.

Environments enable software developers, hardware developers, and OEMs (subsequently referred to as Ecosystem Partners) to easily compile an application for the release of their choice, and to test both the deployment and functionality of the resulting build in other environments. The ABE makes it easy to test software across a wide range of Red Hat Enterprise Linux versions and architectures. In addition, you can test source code across a wide range of gcc/g++ versions, including the gcc 4 preview that has buffer overflow detection technology.

1.1. ABE Prerequisites

The ABE has the following prerequisites:

Red Hat Enterprise Linux Version: 4

When the Application Build Environment is made generally available in early 2005, it will require a host platform running Red Hat Enterprise Linux 4. However, currently you can use any Red Hat host platform.

In order for the ABE to function properly, SELinux must be disabled.

To disable SELinux, enter:
`system-config-securitylevel`

Disk Space: 4GB per Red Hat Enterprise Linux version

The ABE enables you to build your applications simply by pointing the ABE at ISO9660 images of Red Hat Enterprise Linux and running `build`. But given that the ABE requires ISO images, you will need roughly 4GB of disk space on your system for every version of Red Hat Enterprise Linux on which you want to build or test. You will need an active RHN account for access to the ISO images for each version of Red Hat Enterprise Linux that you wish to build/test against.

For example, if you want to build an application to run on Red Hat Enterprise Linux version 4, you will need to download the four primary ISO images for the operating system. It is necessary to download all four primary ISO images so that you will have all the components that make up the operating system. It is not necessary for you to download the documentation ISO images or other layered product images to build a package, although it is helpful for other reasons.

Privileges: root (or sudo)

In order to use the ABE, you must have root privileges either by switching user to root or by allowing `sudo` for the executable commands in the ABE. Until Red Hat has made necessary changes to the ISO images for Red Hat Enterprise Linux, you will need to have the ISO images for all the different versions of Red Hat Enterprise Linux in different directories. This is because the ABE attempts to determine the specific version of Red Hat Enterprise Linux by looking at the ISO image itself. Until the changes are implemented in Red Hat Enterprise Linux 4 U1, you will need separate directories.

Resources: spec files and Source RPMs

To be able to use the ABE for building packages, you must have correctly written spec files and source RPMs. (This will be described in a separate document.)

1.2. Installing the ABE Kit

The ABE tool is available in two places:

- The general public can download the tool from http://people.redhat.com/mwaite/
- Developers who have access to the release candidate for Red Hat Enterprise Linux 4 can access the ABE kit from the RHN child channels for the respective operating system: https://rhn.redhat.com/network/software/index.pxt

Once you have the necessary root privileges, simply install the ABE kit as you would any other RPM. For example:

```
[root@dhcp83-48 /]# rpm -ivh rhel-abe-4.0-2.7.noarch.rpm
Preparing...        ########################################### [100%]
   1:rhel-abe       ########################################### [100%]
[root@dhcp83-48 /]#
```

1.2.1. What the Install Script Does

The Application Build Environment RPM creates directories and installs files in the following locations:

```
/etc
/etc/abe
/etc/abe/location
/etc/abe/profiles
/etc/abe/profiles/as21
/etc/abe/profiles/rhel3
/etc/abe/profiles/rhel4
/opt/abe
/usr/bin
/usr/bin/buildpackage.pl
/usr/bin/mkchroot.pl
/usr/lib/rhel-abe
/usr/lib/rhel-abe/installpackage.pl
/usr/lib/rhel-abe/macros.solve
/usr/lib/rhel-abe/phase2.sh
/usr/share/doc
/usr/share/doc/rhel-abe
/usr/share/doc/rhel-abe/ABEkit-1.0.pdf
/usr/share/doc/rhel-abe/README
/usr/share/rhel-abe
/usr/share/rhel-abe/minimumset.as21
/usr/share/rhel-abe/minimumset.rhel3
/usr/share/rhel-abe/minimumset.rhel4
/usr/share/rhel-abe/popt-1.7.1-1.as21.i386.rpm
/usr/share/rhel-abe/rpm-4.1.1-1.as21.i386.rpm
/usr/share/rhel-abe/rpm-build-4.1.1-1.as21.i386.rpm
/usr/share/rhel-abe/rpm-devel-4.1.1-1.as21.i386.rpm
```

The contents of the `/usr/lib/rhel-abe/` directory are mostly for the internal working of the ABE itself and are not meant to be changed. Here is a brief description of the contents:

- `/usr/lib/rhel-abe/installpackage.pl` works inside the chrooted environment where it installs packages and resolves dependencies.
- `/usr/lib/rhel-abe/macros.solve` is an RPM helper (a type of configuration file).

- `/usr/lib/rhel-abe/phase2.sh` is the transition point between the active host environment and the new chrooted environment. `phase2.sh` is the switch that gives control over to the chrooted environment.
- `/usr/share/rhel-abe/minimumset.as21` holds the contents of the minimum set of packages needed to build an application for the AS 2.1 version of Red Hat Enterprise Linux.
- `/usr/share/rhel-abe/minimumset.rhel3` holds the contents of the minimum set of packages needed to build an application for Red Hat Enterprise Linux 3.
- `/usr/share/rhel-abe/popt-1.7.1-1.as21.i386.rpm`, `/usr/share/rhel-abe/rpm-4.1.1-1.as21.i386.rpm`, `/usr/share/rhel-abe/rpm-build-4.1.1-1.as21.i386.rpm`, and `/usr/share/rhel-abe/rpm-devel-4.1.1-1.as21.i386.rpm`enable you to build packages for Red Hat Enterprise Linux 2.1. These are required because of differences in the RPM database versions between major releases of Red Hat Enterprise Linux.
- `/etc/abe/profiles` is created by the configuration tool. This will store the various requirements specific to the version of Red Hat Enterprise Linux for which you are building your application.
- The final ABE component is the build root directory. The default setting of this is `/opt/abe`. You can change this setting by modifying the contents of the `/etc/abe/location` configuration file.

1.3. Configuring the ABE

The `/etc/abe/profiles/rhel3` file is the configuration file that you use to build an application. Its contents are:

```
isodir=/isos
package=/usr/share/rhel-abe/minimumset.rhel3
subdir=rhel3
32bit=1
```

where:

`isodir`

The location of the ISO images for the version of Red Hat Enterprise Linux in question. The location you specify must be on your local file system; the ISOs cannot be NFS-mounted at this time.

`package`

A pointer to the minimum package set needed to build an application for a specific version of Red Hat Enterprise Linux.

If you look at the contents of `/usr/share/rhel-abe/minimumset.rhel3`, you will see that it contains the 56 packages that are required to build an RPM for Red Hat Enterprise Linux 3. The exact packages needed to build an application against versions 2.1 and 3 for Red Hat Enterprise Linux are listed in the Appendixes of this document.

`subdir`

The location of the chrooted environment. The `subdir_base` is defined in `/etc/abe/location`, which is set by default to `/opt/abe`.

`32bit`

The architecture of the build environment. The value 1 is used to define 32bit and the value 0 is used for 64bit distributions. This configuration parameter will not be needed for Red Hat

Enterprise Linux 4U1 as the ABE will be able to detect this information directly from the ISO images themselves.

Edit the `/etc/abe/profiles/rhel3` file to have values appropriate for your system and the Red Hat Enterprise Linux version that you want to build.

Once you have configured the ABE environment, you are ready to create the build environment as described in Section 1.3.1 *Running the ABE Build Script*.

1.3.1. Running the ABE Build Script

The ABE RPM installs two perl scripts: `/usr/bin/buildpackage.pl` and `/usr/bin/mkchroot.pl`:

- `/usr/bin/mkchroot.pl` makes the build environment: it mounts the ISO images and extracts the contents of the ISO images to the location you specified in Section 1.3 *Configuring the ABE*. It installs the minimum package set and a few other internal components that are not visible to you. This process may take quite a few minutes.

Note

After the ISO images have been processed by `mkchroot.pl`, it is no longer necessary for you to keep the ISO images on the system as all the required data has been populated in the base directory for the build environment.

- `/usr/bin/buildpackage.pl` is the command that you will use to build your packages.

Run the `perl` scripts:

```
perl /usr/bin/mkchroot.pl profile_name
perl /usr/bin/buildpackage.pl profile_name path_to_source_rpm
```

This builds an RPM out of the source RPM in question, based on the profile that you select. The profile is discussed in detail in Section 1.3 *Configuring the ABE*.

The ABE places the newly created RPM in `/usr/src/redhat/RPMS/noarch` in the `buildroot` environment (as defined in the specific profile used).

So that the newly built RPMs can be easily located, the ABE creates a symbolic link from each buildroot environment to `/opt/abe/profile_name/RPMS` on your local machine. For example, `/opt/abe/rhel4//usr/src/redhat/RPMS` will be linked to `/opt/abe/rhel4/RPMS`.

1.4. Considerations When Building Applications

This section contains advice you may find useful when using the ABE.

1.4.1. If the Build Fails

Builds can fail for different reasons. First, make sure that you have the configurable parameters set up correctly. They are listed here:

- `/etc/abe/profiles/profile_specific_to_your_build_environment`
- `/etc/abe/location`
- `/usr/share/rhel-abe/minimumset.*`

Note

You may not be able to build some of the packages that ship with Red Hat Enterprise Linux 3U3 based on the packages that are included in the ISO images because of incomplete spec files.

Please report to abe-feedback@redhat.com the successes and failures you have building packages such as Java, kernel modules, and so on.

1.4.2. Building Packages that have 3rd-Party Dependencies

If you want to use the ABE to build a package that has dependencies on third-party packages that are not distributed by Red Hat, you need to take into consideration the chrooted buildroot environment of the ABE.

If you do not "chroot" into one of the correct buildroots, your dependent packages will be installed on your host platform outside of the ABE file system and you will experience errors such as:

```
[root@computer_name ~]#
rpm -ivh /tmp/sample_pkg-0.6.4-1.rf.i386.rpm
error: open of /tmp/sample_pkg-0.6.4-1.rf.i386.rpm failed:
       No such file or directory
[root@computer_name ~]#
```

To avoid this problem, follow these steps:

First, copy the dependent package into the chrooted environment:

```
cp /tmp/sample_pkg-0.6.4-1.rf.i386.rpm /opt/abe/rhel4/tmp
```

Make sure to "chroot" into the correct buildroot environment. For example: `chroot /opt/abe/rhel4`

Your packages should now install into the correct buildroot environment:

```
[root@computer_name ~]#
rpm -ivh /tmp/sample_pkg-0.6.4-1.rf.i386.rpm
Preparing.##########################################[100%]
   1:sample_pkg########################################[100%]
[root@computer_name ~]#
```

Now you are set to build your packages with dependent packages not currently provided by Red Hat in our distribution.

Tip

A similar workaround is the following:

```
rpm -ivh --aid --root /opt/abe/rhel4  /tmp/sample_pkg-0.6.4-1.rf.i386.rpm
```

1.5. Questions and Answers

I have tested the Application Build Environment and want to send feedback to Red Hat. I would also like to submit feature requests for the next release of the ABE. How do I do this?

Send feedback and feature requests to abe-feedback@redhat.com.

Will the ABE and its associated documentation be internationalized?

Yes. All tools and other enablers will be made easy to use for all Ecosystem Partners worldwide.

Is this ABE a product that Red Hat will sell?

Not at this time. The ABE is part of the larger Software Developer Program, which is a complete suite of tools that are designed to enable developers to build applications supported on Red Hat Enterprise Linux.

Will the ABE allow developers to build and test applications for other Linux distributions?

No. The ABE will not be certified by Red Hat for applications on other Linux distributions.

Can developers use the ABE to build applications for Fedora?

The ABE requires Red Hat Enterprise Linux as the host platform. Red Hat Quality Assurance has not tested the Application Build Environment with Fedora.

Will the ABE allow me to use a toolchain or compiler other than what is shipped with Red Hat Enterprise Linux?

The ABE will work with other toolchains and compilers. If you need to do this to migrate your applications, do the following:

1. Install Red Hat Enterprise Linux as your host operating system.
2. Install the Application Build Environment and configure it.
3. Create a build root as described in Section 1.3 *Configuring the ABE*.
4. Install the toolchain/compiler in the build root.
5. Build your application. The ABE will use the toolchain/compiler it finds in the build root.

Appendix A.

Required Red Hat Enterprise Linux 4 Packages

Here is the minimum package set needed to build an application for Red Hat Enterprise Linux 4:

Note

Each major version of Red Hat Enterprise Linux may require a different number of packages in its minimum package set.

basesystem	bash	beecrypt
bzip2-libs	chkconfig	coreutils
cpio	cracklib	cracklib-dicts
db4	db4-utils	dev
device-mapper	e2fsprogs	elfutils-libelf
ethtool	filesystem	findutils
gawk	glib2	glibc
glibc-common	gpm	grep
gzip	hotplug	hwdata
info	initscripts	iproute
iputils	kernel	less
libacl	libattr	libgcc
libselinux	libsepol	libstdc++
lvm2	MAKEDEV	mingetty
mkinitrd	mktemp	module-init-tools
modutils	mount	ncurses
net-tools	pam	pcre
popt	procps	psmisc
readline	redhat-release	rpm
rpmdb-redhat	rpm-libs	sed
setup	shadow-utils	sysklogd
SysVinit	tar	termcap
tzdata	udev	usbutils
util-linux	which	words
zlib		

Appendix B. Required Red Hat Enterprise Linux 3 Packages

Here is the minimum package set needed to build an application for Red Hat Enterprise Linux 3:

Note

Each major version of Red Hat Enterprise Linux may require a different number of packages in its minimum package set.

basesystem	bash	beecrypt
bzip2-libs	chkconfig	coreutils
cracklib	cracklib-dicts	db4
dev	e2fsprogs	elfutils-libelf
ethtool	filesystem	findutils
gawk	glib	glibc
glibc-common	gpm	grep
info	initscripts	iproute
iputils	laus	laus-libs
libacl	libattr	libgcc
libtermcap	mingetty	mktemp
modutils	mount	ncurses
net-tools	pam	pcre
popt	procps	psmisc
rpm	rpmdb-redhat	rpm-libs
sed	setup	shadow-utils
sysklogd	SysVinit	termcap
tzdata	util-linux	which
words	zlib	

Appendix C.

Required Red Hat Enterprise Linux 2.1 Packages

Here is the minimum package set needed to build an application for Red Hat Enterprise Linux 2.1:

Note

Each major version of Red Hat Enterprise Linux may require a different number of packages in its minimum package set.

basesystem	bash	beecrypt
bzip2-libs	chkconfig	coreutils
cracklib	cracklib-dicts	db3
db3-utils	dev	diffutils
e2fsprogs	elfutils-libelf	ethtool
file	filesystem	fileutils
findutils	gawk	glib
glibc	glibc-common	gpm
grep	gzip	info
initscripts	iproute	iputils
kernel	libacl	libaio
libattr	libgcc	libtermcap
logrotate	losetup	mingetty
mkinitrd	mktemp	modutils
mount	ncures	net-tools
pam	patch	pcre
popt	procps	psmisc
rpmdb-redhat	sed	setup
shadow-utils	sh-utils	sysklogd
SysVinit	tar	tcl
termcap	textutils	tzdata
util-linux	which	words
zlib		