Cross-Platform Building

The Jikes™ RVM build process consists of two major phases: the building of a *boot image*, and the building of a *boot loader*. The boot image is built using a Java™ program executed within a host JVM and is therefore platform-neutral. By contrast, the boot loader is written in C, and must be compiled on the target platform.

Because building the boot image can be time-consuming, you may prefer to build the boot image on a faster machine than the target platform. You may also be porting Jikes RVM to a target platform that lacks tools such or whose development environment is otherwise unpleasant. To cross-build, simply set your `host.name` and `target.name` properties to different values.

For example, to build the `prototype` configuration for AIX™ on a Linux host:

```
% cd $RVM_ROOT
% ant -Dconfig.name=prototype -Dhost.name=ia32-linux
-Dtarget.name=ppc32-aix cross-compile-host
```

The build process is then completed by building just the boot loader on an AIX host:

```
% cd $RVM_ROOT
% ant -Dconfig.name=prototype -Dhost.name=ppc32-aix cross-compile-target
```

After the script has completed successfully, you should be able to run Jikes RVM.

The building of the boot loader must occur in the same directory as the rest of the build. This can either be done transparently via a network file system, or by copying the build directory from the first host to the target.

Dependencies

To compile the boot image on the host system you will also need to have built any dependencies on the target machine and then copied them to the host machine. You will also need to add an appropriate line into your *${components.dir}components.properties* file such as the following (if the target system was pppc32-linux).

ppc32-linux.classpath.lib.dir=path/to/components/classpath/95/ppc32-linux/lib

Note
It may be possible to simply build the dependencies on the host machine, modify the *${components.dir}/components.properties* so that the dependency property for target machine maps to the same value as the dependency property on the host machine. This works at the current time but may fail in the future if classpath changes the API between platforms. i.e.

ppc32-linux.classpath.lib.dir=path/to/components/classpath/95/ia32-linux/lib