3

Compilation and GNU Make Stanzas

RTEMS is compiled using the GNU **gmake(1G)** utilities. All examples in this section are with the **gmake(1G)** command. Note that the installation procedure for GNU Make installs it as *make*. It is referred to as **gmake** in this document to distinguish it from any other make utilities which may also be on the development system.

The **GNU** Make utility uses a file that describes the relationships among the files and the operations necessary for updating each file. The **GNU** Make utility uses stanzas to specify which set of relationships to update. Each component and suite control directory contains a make control file, **Makefile**, which describes the relationships which must be checked and the associated update operations for each stanza. This facility is used to perform compilation, to remove intermediate files, to install **RTEMS**, and to maintain release and working set source and documentation notebooks. The following is a list of stanzas used by **RTEMS** make control files:

Stanza	Description
all	perform compilation but do not install
install	perform compilation if directory contains source but do not install
clean	delete most generated files and directories for the current CPU and target
clobber	delete all generated files and directories for the current CPU and target

Stanza	Description
print	print all source files that have changed since the last print was performed (on the default printer)
printall	print all source files, even if unchanged since last printing (on the default printer)
printrel	print all source files that have changed since the last release was printed (on the release printer)
printrelall	print all source files (on the release printer)

3.1 Making a Component

A single component can be compiled by changing to the directory which contains that component and performing the following command:

```
gmake
```

This is equivalent to the following command:

```
gmake all
```

Both commands will result in the **GNU** Make utility determining which files require compilation or assembly. If any files require compilation or assembly, then these operations will be performed followed by the appropriate archive or link command. Files installed are placed in subdirectories under the install point. The install point is determined by the setting of the variable **PROJECT_HOME** in the file c/make/custom/**<TARGET>**.cfg or ada/make/**<VENDOR>**/custom/**<TARGET>**.cfg.

NOTE: Whether the **Ada** implementation supports a unified makefile structure and the install point are a function of the **Ada** toolset being used.

If the current directory is not a leaf directory, then the requested operation will be performed recursively to all subdirectories under the current directory.

By specifying one of the other stanzas supported by this component control directory **makefile**, the **GNU** Make utility can be used to perform such operations as printing out of date files in a component (print stanza) or removing all automatically generated files in a component (clean stanza).

NOTE: For many components it is not possible to compile them until other components have been installed.

3.2 Optional Manager Support

RTEMS allows the C applications developer to build images that only contain those components of the executive that are needed, leaving out those that will not be utilized. This is accomplished by the **RTEMS** Makefile system linking in the "stub" versions of the optional managers in the place of those managers not needed by the specific application. The application Makefile sets the system variable $(MANAGERS) list to contain those managers that are required by the application. The **RTEMS** Makefile system then is able to build a list of managers that are unwanted, effectively linking in the stubbed versions of these managers before the RTEMS library is built.

For more information and implementation details refer to the following files:

- *c/make/leaf.cfg,*
- *a Makefile for a test or sample application, and*
- *a compiler description file from c/make/compilers*

These files demonstrate the use of $(MANAGERS) and how the unwanted managers are handled.