5

RTEMS Specific Utilities

This section describes the additional commands available within the **RTEMS Development Environment**. Although some of these commands are of general use, most are included to provide some capability necessary to perform a required function in the development of the **RTEMS** executive, one of its support components, or an **RTEMS** based application. The commands have been classified into the following categories for clarity:

- *C Language Specific Utilities*
- *Ada Language Specific Utilities*

Some of the commands are implemented as C programs. However, most commands are implemented as Bourne shell scripts. Even if the current user has selected a different shell, the scripts will automatically invoke the Bourne shell during their execution lifetime.

The commands are presented in **UNIX™** manual page style for compatibility and convenience. A standard set of paragraph headers were used for all of the command descriptions. If a section contained no data, the paragraph header was omitted to conserve space. Each of the permissible paragraph headers and their contents are described below:

Header	Description
SYNOPSIS	describes the command syntax
DESCRIPTION	a full description of the command
OPTIONS	describes each of the permissible options for the command
NOTES	lists any special noteworthy comments about the command
ENVIRONMENT	describes all environment variables utilized by the command
EXAMPLES	illustrates the use of the command with specific examples
FILES	provides a list of major files that the command references
SEE ALSO	lists any relevant commands which can be consulted

Most environment variables referenced by the commands are defined for the **RTEMS Development Environment** during the login procedure. During login, the user selects a default **RTEMS** environment through the use of the **Modules** package. This tool effectively sets the environment variables to provide a consistent development environment for a specific user. Additional environment variables within the **RTEMS** environment were set by the system administrator during installation. When specifying paths, a command description makes use of these environment variables.

When referencing other commands in the **SEE ALSO** paragraph, the following notation is used: command(code). Where command is the name of a related command, and code is a section number. Valid section numbers are as follows:

Section	Description
1	Section 1 of the standard **UNIX™** documentation
1G	Section 1 of the **GNU** documentation
1R	a manual page from this document, the **RTEMS Development Environment Guide**

For example, **ls(1)** means see the standard ls command in section 1 of the **UNIX™** documentation. **gcc020(1G)** means see the description of **gcc020** in section 1 of the **GNU** documentation.

5.1 C Language Specific Utilities

The C language utilities provide a powerful set of tools which combine to allow operations within the **RTEMS Development Environment** to be consistent and easy to use. Much effort was devoted to providing as close to the standard **UNIX™** and **GNU** style of operations as possible. Each of these utilities are described in the section below.

5.1.1 packhex - Compress Hexadecimal File

SYNOPSIS

```
packhex <source >destination
```

DESCRIPTION

packhex accepts Intel Hexadecimal or Motorola Srecord on its standard input and attempts to pack as many contiguous bytes as possible into a single hexadecimal record. Many programs output hexadecimal records which are less than 80 bytes long (for human viewing). The overhead required by each unnecessary record is significant and **packhex** can often reduce the size of the download image by 20%. **packhex** attempts to output records which are as long as the hexadecimal format allows.

OPTIONS

This command has no options.

EXAMPLES

Assume the current directory contains the Motorola Srecord file download.sr. Then executing the command:

```
packhex <download.sr >packed.sr
```

will generate the file packed.sr which is usually smaller than download.sr.

CREDITS

The source for packhex first appeared in the May 1993 issue of *Embedded Systems* magazine. The code was downloaded from their BBS. Unfortunately, the author's name was not provided in the listing.

5.1.2 unhex - Convert Hexadecimal File into Binary Equivalent

SYNOPSIS

```
unhex [-valF] [-o file] [file [file ...] ]
```

DESCRIPTION

unhex accepts Intel Hexadecimal, Motorola Srecord, or TI 'B' records and converts them to their binary equivalent. The output may sent to standout or may be placed in a specified file with the -o option. The designated output file may not be an input file. Multiple input files may be specified with their outputs logically concatenated into the output file.

OPTIONS

This command has the following options:

Code	Description
v	Verbose
a *base*	First byte of output corresponds with *base* address
l	Linear Output
o *file*	Output File
F *k_bits*	Fill holes in input with 0xFFs up to *k_bits* * 1024 bits

EXAMPLES

The following command will create a binary equivalent file for the two Motorola S record files in the specified output file binary.bin:

```
unhex -o binary.bin downloadA.sr downloadB.sr
```

5.1.3 size_rtems - Report RTEMS Size Information

SYNOPSIS

size_rtems

DESCRIPTION

size_rtems analyzes **RTEMS** and determines all of the critical sizing information which is reported in the related documentation.

EXAMPLES

To generate the **RTEMS** size report for the currently configured processor, execute the following command:

```
size_rtems
```

Although the actual size information will differ, a report of the following format will be output:

```
        RTEMS SIZE REPORT

                  CODE  DATA    BSS
                  ==================
MANAGERS:         15988 0       0
CORE    :         4568  0       0
CPU     :         364   0       0
OVERALL :         20556 0       0
MINIMUM :         8752  0       0

init    :         1592  0       0
tasks   :         2440  0       0
intr    :         64    0       0
clock   :         2252  0       0
sem     :         876   0       0
msg     :         1624  0       0
event   :         604   0       0
signal  :         212   0       0
part    :         872   0       0
region  :         844   0       0
dpmem   :         532   0       0
timer   :         424   0       0
io      :         288   0       0
fatal   :         40    0       0
rtmon   :         764   0       0
mp      :         2984  0       0

sem     :         4     0       0
msg     :         4     0       0
event   :         4     0       0
signal  :         4     0       0
part    :         4     0       0
region  :         4     0       0
timer   :         4     0       0
dpmem   :         4     0       0
io      :         4     0       0
rtmon   :         4     0       0
mp      :         8     0       0
```

SEE ALSO

gsize020(1G), gsize386(1G), gsize960(1G)

5.2 Ada Language Specific Utilities

The Ada language utilities provide a powerful set of tools which combine to allow operations within the **RTEMS Development Environment** to be consistent and easy to use. Much effort was devoted to providing as close to the standard **UNIX™** and **GNU** style of operations as possible. Each of these utilities are described in the section below.

5.2.1 scope - Start Tartan AdaScope

SYNOPSIS

```
scope [a|b|A|B] [deb]
scopea
scopeb
```

DESCRIPTION

scope invokes the Tartan AdaScope debugger on the executable image for the program in the current directory. The executable is downloaded to the current target.

The second argument may be an 'A' or 'B' with case being irrelevant. This forces scope to start AdaScope on a particular serial port. **scopea** and **scopeb** are equivalent to the commands **scope a** and **scope b**, respectively.

If the third argument is "deb", then the file **$RTEMS_SRC_BASE/tmp/brkpts.cmd** is read at startup time by AdaScope. This allows one to conditionally load a temporary set of breakpoints. The second argument is required when using this feature.

EXAMPLES

To invoke AdaScope and download the executable for Test 1 of the **Single Processor Test Suite**, execute the following commands:

```
cd $RTEMS_SRC_BASE/ada/sptest/sp1
scope
```

NOTES

This command assumes that a windowing environment is used and is typically invoked as a background process.

The specific serial ports used by AdaScope must be tailored for the local site.

FILES

$RTEMS_SRC_BASE/ada/bin/menu.cmd
obj/m68020/tartan/test.cmd
$RTEMS_SRC_BASE/ada/bin/resize.cmd
$RTEMS_SRC_BASE/tmp/break.cmd
$RTEMS_SRC_BASE/tmp/brkpts.cmd

SEE ALSO

Tartan Development System 68xxx v4.3.1 AdaScope

5.2.2 packhex - Compress Hexadecimal File

SYNOPSIS

```
packhex <source >destination
```

DESCRIPTION

packhex accepts Intel Hexadecimal or Motorola Srecord on its standard input and attempts to pack as many contiguous bytes as possible into a single hexadecimal record. Many programs output hexadecimal records which are less than 80 bytes long (for human viewing). The overhead required by each unnecessary record is significant and **packhex** can often reduce the size of the download image by 20%. **packhex** attempts to output records which are as long as the hexadecimal format allows.

OPTIONS

This command has no options.

EXAMPLES

Assume the current directory contains the Motorola Srecord file download.sr. Then executing the command:

```
packhex <download.sr >packed.sr
```

will generate the file packed.sr which is usually smaller than download.sr.

CREDITS

The source for packhex first appeared in the May 1993 issue of *Embedded Systems* magazine. The code was downloaded from their BBS. Unfortunately, the author's name was not provided in the listing.

5.2.3 size_rtems - Report RTEMS Size Information

SYNOPSIS

```
size_rtems
```

DESCRIPTION

size_rtems analyzes **RTEMS** and determines all of the critical sizing information which is reported in the related documentation.

EXAMPLES

To generate the **RTEMS** size report for the currently configured processor, execute the following command:

```
size_rtems
```

Although the actual size information will differ, a report of the following format will be output:

```
                RTEMS SIZE REPORT

                     CODE        DATA   CONST  BSS
                     ==========================
MANAGERS           : 26842       1082   1438   0
CORE               : 20300       1680   978    0
CPU                :  1192       1060   0      0
OVERALL            : 48334       3822   2416   0
MINIMUM            : 27236       2896   978    0

rtems              : 280         508    0      0
initialization     : 356         0      0      0
rtems_tasks        : 2930        8      266    0
interrupts         : 66          0      0      0
clock              : 212         0      0      0
semaphore          : 2094        48     143    0
message_queue      : 2782        72     276    0
event              : 1088        9      40     0
signals            : 316         8      40     0
partition          : 1456        48     164    0
region             : 1214        48     168    0
dual_ported_memory: 904          40     0      0
timer              : 1390        44     46     0
io                 : 1544        48     132    0
fatal              : 278         0      0      0
rate_monotonic     : 1928        44     41     0
MP                 : 8004        157    122    0
```

SEE ALSO

tdump68xxx in the **Tartan Ada Development System 68xxx v4.3.1 Tool Set**