**`The Unpacker'**

tu is a program to read RSR binary dump files, decode them, and produce another file more to the user's liking. Usually, this is an ASCII file where each line represents an RSR record, and each word in the line represents the date, time, or one of the quantities measured by the RSR.

The behavior of the program is governed by a set of *processing parameters*. Each parameter has a default value, chosen to provide 'typical' output. However, the user has the ability to change each parameter to provide specialized functionality. To override the default values, the user may either enter a new value in a configuration file, or type the new value on the command line. Entries in the configuration file take precedence over the defaults, and command line flags have priority over all others. Usually, the configuration file is prepared with permanent, or rarely changed, parameters. Then, when unusual circumstances arise, command-line flags are used to create a specialized output file.

**Configuration file**

The config file is a simple list of the parameters the user wishes to change from their default settings. The configuration file is **not required** for the operation of tu; the functionality it represents is provided to simplify the RSR file conversion process. If a configuration file is used, it must be located in the directory that the program is invoked from. For Unix users, this is different from the usual method of locating configuration files in the user's home directory. On Unix machines, the default name for the config file is "**.turc"**, and on DOS machines, it's "**tu.ini**". However, an alternate config file may be specified on the command line with the -f flag. Note that any number of processing parameters may be specified in the config file. Unrecognized parameters are **ignored,** usually with a warning message (no message is produced when the quiet [-q] flag is given). Comments in the config file are introduced with the '#' character and continue until the end of line. Blank lines are allowed. Each non-comment, non-blank line in the file is assumed to be parameter specification statement, the form of which is

**param-name = value**

where param-name is one of the processing parameters (enumerated below), and value is the new value to override the default. Space characters surrounding the '=' character are optional: white space is ignored in the file. Legal values are defined for each parameter and are discussed below. Here is an example:

# not real parameters, just for show; this would be ignored

separator = ,

Since white space is ignored, the effect of the preceding example would be to set the parameter 'separator' to the new value ','.

**Command Line Flags**

The second method of overriding default behavior is through the use of command-line flags. These are extra commands the user gives to the command interpreter (command.com or the shell) when the program is invoked. Each flag is introduced with the '-' character and is followed immediately by a single character (the 'flag') that indicates which parameter is being addressed. In most cases, the new value of the parameter follows the flag. Several flags may be given, and they may appear in any order. For example, a user might run the program like this

C> tu -t hms -o rsr.dat download.rsr

in a DOS environment. This command line has 2 flags which override default behavior. The first is the '-t' flag, it's new value is 'hms', and the second is '-o', with value 'rsr.dat'. If any flags are unknown or in error, the program halts immediately with a message indicating the problem. The sole exception to this rule is the '-q' flag. The command line is first scanned to check for the presence of '-q', and if present, no messages are emitted for the duration of the program. The implication of this is that the user can only detect the success or failure of the program by querying its return status, or by examining the output file.

**Usage**

To execute tu, the user must specify RSR download files to convert. This is also done via the command line. All words on the command line beginning with the first non-flag are taken to be the names of RSR files to convert. At least one filename is required. If more are provided, they are processed in sequence using the same processing parameters. In general, the program usage is

tu [optional flags] filename [optional additional filenames]

**Processing parameters**

The following table describes all the available processing parameters, what their default value is, the configuration file name, the command line flag, and the allowed values they may assume.

**Parameter\_Default\_Config File Param. Name\_Command Line Flag\_Allowed Values\_**

Quiet\_No\_-\_-q\_None. If given, the program emits no diagnostic messages. The user must determine program status from the environment.

Help\_No\_-\_-h\_None. If given, a usage message is printed, and the program exits immediately, abandoning processing.\_\_

Version\_No\_-\_-V (upper case)\_None. If provided, the program version number is printed and the program exits immediately, abandoning processing.

Config file\_.turc (Unix)

tu.ini (DOS)\_-\_-f\_The word following the flag is taken to be the name of an alternate configuration file. Processing is identical to that of the default.

Verbosity\_Normal\_verbosity\_-v (lower case)\_Acceptable values are “quiet:, "normal" and "verbose"

Time Zone\_0.0 (GMT)\_timezone\_-z\_A number representing the number of hours (west positive) from Greenwich for this RSR file. Any floating point value is accepted.

Night time processing\_-9999\_nighttime\_A string to insert in the output at night for each field that represents a 'daytime only' channel. The special string 'omit' is recognized to remove daytime only channels from the output when the sun is below the horizon.

Separator character\_' ' (1 space character)\_separator\_-s\_A character or string to insert between each field in the output file. The special string 'format' is recognized to provide tabular output.

Date format\_DOY1 (yy doy)\_date\_-d\_This parameter determines how the date is formatted in the output file. Several values are defined:

Value Effect

day1 (year) (day of year starting from 1)

mdy mm dd yy

dmy dd mm yy

unix seconds since 1/1/1970

rsr days since 1/1/1900

joe (days since 1900).(fractional day)\_\_Time format\_hours (hh.frac)\_time\_-t\_This parameter determines how the record time is formatted in the output file.

Value Effect

hms hh mm ss

hours (hours since midnight).(fractional hour)

**note: if the date format is 'unix' or 'joe', then no**

**time field is produced**

Output file\_the screen (stdou t)\_outfile\_The name of a file to write converted data to.

Gap value\_-9998\_gap\_-g

Header in output file\_no\_header\_-H\_Valid values are 'yes' and 'no'. If 'yes,' the first line of the output file lists the values of the RSR file header.

Report interval\_end\_reptime\_-r\_Set the output time interval. Valid parameters are 'start', 'middle', and 'end.'. These correspond to the beginning, midpoint, and end of the sampling period, respectively. When set to 'start', the time reported for each record is the beginning of the sample period; the midpoint for 'middle', and the end for 'end'.

Cosine correction file\_None\_-c\_The name of a 'SolarInfo'-formatted file to read cosine corrections for this RSR file from. If this file is successfully read, the relevant data is cosine corrected before being output.

Extra info\_None\_extra\_-x\_A string composed of the characters a, e, z, h, d, m, s and t. Lower case characters output data in degrees, upper case in radians. The fields are produced after the RSR data, in the order given in the string.

a - solar azimuth

e - solar elevation

z - solar zenith angle

h - hour angle

d - declination

s - relative solar distance

t - true solar time

**note: air mass is not defined when elevation < 0;**

**therefore, air mass obeys the same rules as 'daytime only' data. all others are regarded as 'all the time' quantities.**