GTV User Guide

P.Valiron

Document probably older than you think

1 GTVDEF Language Internal Help

1.1 Language

GTVDEF\ Command Language Summary

```
DEFAULT   : Resets parameters to default values
DELETE    : Deletes a terminal definition
DIRECTORY : Directory of terminal definitions in the current file
LIST      : Lists terminal parameters (local or binary file)
NAME1     : Primary terminal name
NAME2     : Secondary terminal name
READ      : Reads terminal parameters from current binary file
RENAME    : Renames a terminal definition in the current file
USE       : Selects/creates a Binary Terminal Definition File
WRITE     : Writes terminal parameters to current binary file
```

1.1.1 Language NEWS

1. Preliminary version in use since 7-August-1986

2. Special features: TEST command not yet fully debugged, consult HELP TEST; WAIT_ERASE is an integer; specify a negative value for RXY to disable the packing of Tektro bytes; 12 bit adresses in the Tektro GIN cursor sequence are not supported...

3. In project: more TEST facilities, selection of default plotter configurations and procedures, HP-GL and GKS protocols, ...

1.2 DEFAULT

```
[GTVDEF\]DEFAULT
```

DEFAULT resets the parameters of the local table to their default values.

The primary and secondary terminal names and the terminal protocol (NAME1, NAME2 and PROTOCOL) are set to an "undefined" state. DEFOUT is set to "TT".

Escape sequences are disabled (count -1), and most other parameters are set to zero.

1.3 DELETE

```
[GTVDEF\]DELETE name1 [name2]
```

Deletes a terminal definition in the current Binary Terminal Definition

File (see USE).

Arguments specify the primary terminal name and optionally the secondary name. These names can be abbreviated as far as they remain non ambiguous.

1.4 DIRECTORY

[GTVDEF\]DIRECTORY [/OUTPUT OutputFile]

DIRECTORY lists the names of the terminal definitions available in the current Binary Terminal Definition File (see USE).

The output list is directed to STDOUT by default, or to the file in argument if specified.

1.5 LIST

[GTVDEF\]LIST [name1 [name2]] [/OUTPUT File]

Lists terminal parameters.

The option /OUTPUT allows to redirect the output to a specified file (default is STDOUT). An output file can be used later as a command procedure for GTVDEF. For clarity, default parameters values are not listed.

Three modes of operation are selected by the arguments of the command.

1. With no argument, lists the local parameters.

2. If a primary terminal name (and optionally a secondary name) are specified, lists the associated definitions in the current Binary Terminal Definition File (see USE). These names can be abbreviated as far as they remain non ambiguous.

3. If the argument is a star (*), lists ALL definitions in the current BTDF (Binary Terminal Definition File) file. This feature is designed to facilitate the migration of existing terminal definitions to a new version of the GTVIRT environment.

1.6 NAME1

[GTVDEF\]NAME1 [name1]

NAME1 sets the primary terminal name. A secondary terminal name can be also defined to discriminate between several modes of operation for the same terminal (see NAME2).

Primary and secondary terminal names are used by the GTVIRT library and by the associated softwares to retrieve a terminal definition in the Binary Terminal Definition File (see USE). They appear for instance as argument of the command DEVICE to select the current terminal in LAS, GreG and Gildas.

To modify a name, just enter the command with no argument. The current name is returned with an error condition, and the line keypad editing mode of SIC is triggered.

1.7 NAME2

```
[GTVDEF\]NAME2 [name2]
```

NAME2 sets the secondary terminal name. A secondary terminal name may be specified to discriminate between several modes of operation for the same terminal (see NAME1).

To modify a name, just enter the command with no argument. The current name is returned with an error condition, and the line keypad editing mode of SIC is triggered.

1.8 READ

```
[GTVDEF\]READ name1 [name2]
```

READ reads terminal parameters from the currently selected Binary Terminal Definition File (see USE) into the local parameter table in core memory.

The terminal is identified by its primary name and optionally by its secondary name as specified in argument(s). These names can be abbreviated as far as they remain non ambiguous.

1.9 RENAME

```
[GTVDEF\]RENAME old1 [old2] /AS new1 [new2]
```

Renames a terminal definition in the current Binary Terminal Definition File (see USE).

Arguments of the command refer to the old primary and secondary names. Arguments of the option /AS refer to the new names.

Old names can be abbreviated as far as they remain non ambiguous. New names are checked against any ambiguity with other terminal names.

1.10 USE

```
[GTVDEF\]USE [File [/CREATE]]
```

USE selects a Binary Terminal Definition File (BTDF) to be used in connection with commands DIRECTORY, READ, WRITE, LIST, RENAME and DELETE. The option /CREATE creates an empty BTDF file. Remember that the GTVDEF library accesses the BTDF file pointed to by the logical name GAG_TERMINAL.

USE with no argument lists the name of the current BTDF file.

The GTVDEF utility allows to manipulate a local terminal definition table which is volatile (in core memory). The local table can be initialised with the command DEFAULT or can be read from an external file using the commands USE and READ. The target terminal name is defined by the commands NAME1 and NAME2.

The language PARAMS\ is designed to set up or to modify the current terminal parameters in the local table.

Do not forget to issue the command WRITE in order to save the local table into some external BTDF file of your choice.

Other commands of language GTVDEF\ allow to maintain BTDF files. Command LIST is very useful to dump the contents of a BTDF file in a readable format suitable for later input by the GTVDEF utility.

1.11 WRITE

```
[GTVDEF\]WRITE
```

WRITE writes terminal parameters from the local parameter table to the current Binary Terminal Definition File (BTDF, see USE).

Current primary and secondary terminal names are used (see NAME1 and NAME2), and checked against any ambiguity with existing terminal names in the BTDF file. If the terminal name is already present, its definition is updated.

The total number of terminal definitions in a BTDF is limited by the NTMX built-in parameter (currently 32).

2 PARAMS Language Internal Help

2.1 Language

PARAMS\ Command Language Summary

```
ALPHA       : Sequence to toggle into VT100-like alpha plane
CURS1       : Sequence to request the graphic cursor
CURS2       : Sequence to reset the terminal when the cursor has exited
COLOUR      : Sequence(s) to select colours
DASH        : Sequence(s) to select hardware dashed lines
DEFOUT      : Default logical name pointing to the terminal device
ERASE       : Sequence to clear the graphic screen
FORMAT      : Select a format to specify and list escape sequences
GRAPHIC     : Sequence to toggle to graphic plane
INIT1       : Initialisation sequence (GTOPEN)
INIT2       : Reset sequence (GTCLOS)
PHEIGHT     : Height of hardware characters (pixels)
PIXEL_RANGE : Limits of the screen in pixels
PROTECT     : Screen protection
PROTOCOL    : Graphic protocol supported by the terminal
PWIDTH      : Width of hardware characters (pixels)
RXY         : Aspect ratio of the screen
TEXT1       : Sequence to initiate a hardware character generator
TEXT2       : Sequence to terminate hardware characters
WAIT_ERASE  : Specify a wait time after ERASE
WEIGHT      : Sequence(s) to select hardware weightened lines
```

2.1.1 Language NEWS

See HELP GTVDEF NEWS

2.2 FORMAT

[PARAMS\]FORMAT [format]

FORMAT selects a format to specify and list the character strings that define the escape sequences. Available formats are DECIMAL, HEXADECIMAL and ASCII.

1. Decimal format is in the form "<d1><d2>..." where d1, d2,... are decimal unsigned byte values enclosed in brackets.

2. Hexadecimal format is in the form "<h1><h2>..." where h1, h2,... are hexadecimal unsigned byte values enclosed in brackets.

3. Ascii format is in the form "c1c2..." where c1, c2,... are either ordinary ascii characters (SIC separators and brackets excluded) or dec-

imal unsigned byte values enclosed in brackets (as for decimal format) or usual mnemonics for control characters enclosed in brackets (such as <ESC>, <US>, etc.).

To list or modify the format, just enter FORMAT with no argument. The current format is returned with an error condition, and the line keypad editing mode of SIC is triggered.

Escape sequences are always preceded by a decimal count corresponding to the number of bytes of the string. A count -1 or 0 suppresses the escape sequence. By convention, a count value of -1 generally means that the corresponding action is not available (e.g. ALPHA, GRAPHIC if no alpha plane is available, CURS1 if no cursor is available, etc...), and 0 means that the corresponding action is available but is managed internally for the current protocol.

To list or modify an escape sequence, just enter the associated command with no argument. The current sequence is returned with an error condition, and the line keypad editing mode of SIC is triggered.

Index

DEFAULT, 2
DELETE, 2
DIRECTORY, 3

FORMAT, 6

Language, 2, 6
    NEWS, 2, 6
LIST, 3

NAME1, 3
NAME2, 4

READ, 4
RENAME, 4

USE, 5

WRITE, 5