


```
GGGGGGGG  CCCCCCCC  AAAAAA
GGGGGGGG  CCCCCCCC  AAAAAA
GG         CC         AA     AA
GG         CC         AA     AA
GG         CC         AA     AA
GG         CC         AA     AA
GG         CC         AA     AA
GG         CC         AA     AA
GG  GGGGGG  CC  AAAAAAAAAA
GG  GGGGGG  CC  AAAAAAAAAA
GG         GG  CC  AA     AA
GG         GG  CC  AA     AA
GGGGGG    CCCCCCCC  AA     AA
GGGGGG    CCCCCCCC  AA     AA
```

```
....
....
....
....
```

```
RRRRRRRR  EEEEEEEEE  QQQQQQ
RRRRRRRR  EEEEEEEEE  QQQQQQ
RR         RR  EE         QQ     QQ
RR         RR  EE         QQ     QQ
RR         RR  EE         QQ     QQ
RR         RR  EE         QQ     QQ
RRRRRRRR  EEEEEEEEE  QQ     QQ
RRRRRRRR  EEEEEEEEE  QQ     QQ
RR  RR     EE         QQ  QQ  QQ
RR  RR     EE         QQ  QQ  QQ
RR         RR  EE         QQ     QQ
RR         RR  EE         QQ     QQ
RR         RR  EEEEEEEEE  QQQQ  QQ
RR         RR  EEEEEEEEE  QQQQ  QQ
```

GCA.REQ Definitions pertaining to the General Control Area (GCA)
IDENT 026

*
* COPYRIGHT (c) 1978, 1980, 1982, 1984 BY *
* DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS. *
* ALL RIGHTS RESERVED. *
*

* THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *
* ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE *
* INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER *
* COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *
* OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY *
* TRANSFERRED. *
*

* THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE *
* AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT *
* CORPORATION. *
*

* DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS *
* SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL. *
*
*
*

◆◆
FACILITY: DSR (Digital Standard RUNOFF) / DSRPLUS

ABSTRACT:
Definitions for the GCA (Global Control Area). Initialized by
DOOPTS from command-line information.

ENVIRONMENT: Transportable BLISS

AUTHOR: Rich Friday

CREATION DATE: 1978

MODIFIED BY:

026 REM00026 Ray Marshall 16-May-1983
Added support for outputting output lines numbers by adding
another macro: GCA_CMD_050

025 REM00025 Ray Marshall 16-November-1983
Added macros so that Ken's SAVRES module can load cells with
them instead of using numeric offsets. They have the string
"_f_" within them.

024 KFA00024 Ken Alden 10-Oct-1983
Put gca_cmd_pages in the right place.

023 KFA00023 Ken Alden 06-Oct-1983
Added gca_cmd_pages to signal the /pages qualifier.

```

022 KFA00022 Ken Alden 27-Jul-1983
    Added gca_flag_cmd to SAVE list.

021 KFA00021 Ken Alden 15-Jul-1983
    Added gca_do_cont_already, gca_do_index_already.

020 KFA00020 Ken Alden 27-Jun-1983
    Added gca_cref_errcnt to the top eight bits of black_box.

019 KFA00019 Ken Alden 13-Jun-1983
    Made use of GCA[45] for flagging /down line skipping. Used in
    TOPPAG and USKIPL.

018 KAD00018 Keith Dawson 12-May-1983
    Use 16 bits each for page, index, contents, cref counts.

017 KAD00017 Keith Dawson 21-Mar-1983
    Add bits for LNO1 output options.

016 KFA00016 Ken Alden 16-Mar-1983
    Save/restore now permitted in DSR.

015 KAD00015 Keith Dawson 07-Mar-1983
    Global edit of all modules. Updated module names, idents,
    copyright dates. Changed require files to BLISS library.

```

MACRO

```

gca_com_start      = GCA[ 00 ]%,    !A CHSPTR to the first character of the
                    !current command (usually it's a '.')
gca_autopara       = .GCA[ 01 ]%,    ! (SAVEALL) True if user said .AUTOPARAGRAPH
gca_f_autopara     = GCA[ 01 ]%,
gca_autosubt       = .GCA[ 02 ]%,    ! (SAVEALL) True if user said .AUTOSUBTITLE
gca_f_autosubt     = GCA[ 02 ]%,
gca_autotabl       = .GCA[ 03 ]%,    ! (SAVEALL) True if user said .AUTOTABLE
gca_f_autotabl     = GCA[ 03 ]%,
gca_autojust       = .GCA[ 04 ]%,    ! (SAVEALL) True if user said .AUTOJUSTIFY
gca_f_autojust     = GCA[ 04 ]%,

gcaBars            = (GCA[05])<BAR_>%, !GCA BARS ENAB & GCA BARS ON.
gcaBars_enab       = (GCA[06])<BAR_>%, !Controlled by .ENABLE/.DISABLE BAR.
gcaBars_on         = (GCA[07])<BAR_>%, !Controlled by .BEGIN/.END BAR.
gcaEverBars        = (GCA[08])<BAR_>%, !True if change bars ever enabled.
gcaDraft_f         = GCA[ 09 ]%,    !If user said /DEBUG, the character to be used as draft flag (from .VARIABLE command).
gcaChapt           = GCA[ 10 ]%,    !True if document contains chapters/appendices
gcaTwidth_36       = GCA[ 11 ]%,    !TOPS-20 terminal wrap width (DSRPLUS only).
gcaConcat          = GCA[ 12 ]%,    !True if .NO SPACE is pending.
gcaCmd_msg         = GCA[ 13 ]%,    !Routing code for error messages
gcaLit_count       = GCA[ 14 ]%,    !If a counted literal, the # of lines.
gcaPchax           = GCA[ 15 ]%,    !True if a .NUMBER CHAPTER or .NUMBER APPENDIX command has already set up NPAGEN.
gcaTescape_36     = GCA[ 16 ]%,    !TOPS-20 terminal escape characteristics (DSRPLUS only)

gcaCmd_nbits       = GCA[ 17 ]%,    !Ultimate controls for miscellaneous functions.
gcaLit_flag        = GCA[ 18 ]%,    !status of .GCA_FLAG_CMD during literal processing.
gcaCmd_lstbts     = GCA[ 19 ]%,    !Ultimate controls for misc functions.

```

```

gca_literal      = GCA[ 20 ]%, !True between .LITERAL and .END LITERAL
gca_xcase       = .GCA[ 21 ]%, !(SAVEALL)Case rules for index entries.
gca_f_xcase     = GCA[ 21 ]%,

gca_page_cnt    = GCA[ 22 ]%, !Number of pages generated so far.
gca_contents_cnt = GCA[ 23 ]%, !Number of table-of-contents entries written to the .BRN file.
gca_index_cnt   = GCA[ 24 ]%, !Number of index entries written to the .BRN file.
gca_old_cref_cnt = GCA[ 25 ]%, !Number of cross-reference entries read from the old .BRN file.
gca_new_cref_cnt = GCA[ 26 ]%, !Number of cross-reference entries written to the .BRN file.

gca_orange_cnt  = GCA[ 27 ]%, !Count of number of page ranges specified on /PAGE_RANGE switch.
gca_skip_out    = GCA[ 28 ]%, !True if skipping output because of a /PAGE_RANGE specification.
gca_debug_index = (GCA[29])<0,1>%, !True if debugging the index.
gca_debug_toc   = (GCA[29])<1,1>%, !True if debugging the table of contents.
gca_debug_cnd   = (GCA[29])<2,1>%, !True if debugging conditionals (.IF, etc.).
gca_debug_fil   = (GCA[29])<3,1>%, !True if want file names listed.
gca_debug_cref  = (GCA[29])<4,1>%, !True if want cross-reference errors flagged inline.
gca_debug_save  = (GCA[29])<5,1>%, !True if want save/restore information echoed.
gca_max_astn    = GCA[ 30 ]%, !Result of last .AUTOSUBTITLE command. Remembered during .NO AUTOSUBTITLE.
gca_btc         = (GCA[31])<0,1>%, !True if a binary table of contents is wanted.
gca_cmd_btc     = (GCA[31])<1,1>%, !True iff user said /INTERMEDIATE.
gca_bix         = (GCA[31])<2,1>%, !True if a binary index is wanted.

gca_case        = .GCA[ 32 ]%, !(SAVEALL)Set by .UPPER/LOWER CASE
gca_f_case      = GCA[ 32 ]%,
gca_flag_cmd    = .GCA[ 33 ]%, !(SAVE)Last .FLAGS ALL/.NO FLAGS ALL command
gca_f_flag_cmd  = GCA[ 33 ]%,

gca_pwidth      = GCA[ 34 ]%, !Physical page width

gca_lwidth      = .GCA[ 35 ]%, !(SAVE)Logical page width
gca_f_lwidth    = GCA[ 35 ]%,

gca_clit        = GCA[ 36 ]%, !True for counted literals.
gca_note_prim   = GCA[ 37 ]%, !Primary margin adjust for .NOTE
gca_note_alt    = GCA[ 38 ]%, !Alternate margin adjust for .NOTE
gca_skipping    = (GCA[39])<0,1>%, !True if in the FALSE branch of an .IF command.
gca_nia         = GCA[ 40 ]%, !Count of attempted negative indents.
gca_xmarg       = GCA[ 41 ]%, !Count of crossed margin attempts.
gca_normal_xtn  = GCA[ 42 ]%, !Next transaction number for index entry NOT in a footnote.
gca_foot_xtn    = GCA[ 43 ]%, !Next transaction number for index entry in a footnote.
gca_index_fmt   = GCA[ 44 ]%, !If True, generate dots in index. (Not currently set anywhere in the code.)
gca_down_flag   = (GCA[45])<0,1>%, !True if the unconditional lines are for /down.
gca_max_require = GCA[ 46 ]%, !Same as MAX REQUIRE.
gca_req_depth   = GCA[ 47 ]%, !Current depth of .REQUIRE (usually 0, i.e., not a .REQUIRE file).
gca_fehler      = GCA[ 48 ]%, !True if any errors have been found.
gca_errcnt      = GCA[ 49 ]%, !Count of lines of error messages.
gca_err_dir     = GCA[ 50 ]%, !Indicates where errors are to be reported.

gca_keep        = .GCA[ 51 ]%, !(SAVE)True if user said .KEEP.
gca_f_keep      = GCA[ 51 ]%,

gca_io_flags    = GCA[ 52 ]%, !General I/O control flags.
gca_cmd_quick   = (GCA[52])<0,1>%, !True if /QUICK or /NOOUTPUT.
termination_log = (GCA[52])<1,1>%, !True if /LOG.

```

```

gca_flip           = (GCA[52])<2,1>% , !True if /DEC=FLIP.
gca_s_output       = (GCA[52])<3,1>% , !True if /OUTPUT was explicitly specified.
gca_op_dev         = (GCA[52])<4,4>% , !Type of output device (see OPDEV.REQ).

gca_ln01_ital_under = (GCA[52])<8,1>% , ! For LN01 output: set=italics, clear=underlining.
gca_ln01_port_land  = (GCA[52])<9,1>% , ! For LN01 output: set=portrait, clear=landscape.
gca_cmd_pages       = (GCA[52])<10,1>% , !True if /PAGES.

gca_diag1          = GCA[ 53 ]% , !First 16 diagnostic control flags.
gca_diag2          = GCA[ 54 ]% , !Second 16 diagnostic control flags.

gca_xp_ever        = GCA[ 55 ]% , !True if user gave any .XP/.YP commands since last .DX or .PX.
gca_line_pend      = GCA[ 56 ]% , !Used for OUTHDR in determining if the header has been processed and output.

gca_black_box      = GCA[ 57 ]% , !True if user said /AUTOMATIC.
gca_cref_errcnt    = (GCA[57])<8,8>% , !Number of unresolved cross references.
gca_cross_reference = (GCA[58])<0,1>% , !True if user said /CROSS_REFERENCE (it's the default).
gca_expand_cref    = (GCA[58])<1,1>% , !True if cross references should be expanded.
gca_old_brn_exists = (GCA[58])<2,1>% , !True if an old .BRN file exists and was read.
gca_cref_changed   = (GCA[58])<3,1>% , !True if any cross-reference changed in the course of this run.
gca_do_cont_already = (GCA[58])<4,1>% , !True if a do contents has been processed in this run.
gca_do_index_already = (GCA[58])<5,1>% , !True if a do index has been processed in this run.
gca_rerun_count    = (GCA[59])<0,8>% , !Number of times DS+PLUS must be rerun for complete and correct /AUTOMATIC processing.
gca_pass_count     = (GCA[59])<8,8>% , !Number of the current /AUTOMATIC rerun pass.

```

LITERAL

```

gca_size = 64; ! Number of cells in GCA. As of version V04-000 there are
           ! 5 free cells.

```

```

!The following literals define where errors are to be reported
!NOTE: The bit representation of these literals is used by DOOPTS.

```

LITERAL

```

report_err_none    = 0, !Report no error messages
report_err_file    = 1, !Report in output file only.
report_err_std     = 2, !Report on standard error log only.
report_err_both    = 3; !Report errors in both output file and on standard error device.

```

```

!Definition of diagnostic control flags. The names may be later changed,
!but for now, these names will at least provide easy access for initial use.

```

MACRO

```

diag1_0 = (GCA_DIAG1)<0,1>% ,
diag1_1 = (GCA_DIAG1)<1,1>% ,
diag1_2 = (GCA_DIAG1)<2,1>% ,
diag1_3 = (GCA_DIAG1)<3,1>% ,
diag1_4 = (GCA_DIAG1)<4,1>% ,
diag1_5 = (GCA_DIAG1)<5,1>% ,
diag1_6 = (GCA_DIAG1)<6,1>% ,
diag1_7 = (GCA_DIAG1)<7,1>% ,
diag1_8 = (GCA_DIAG1)<8,1>% ,
diag1_9 = (GCA_DIAG1)<9,1>% ,
diag1_10 = (GCA_DIAG1)<10,1>% ,
diag1_11 = (GCA_DIAG1)<11,1>% ,
diag1_12 = (GCA_DIAG1)<12,1>% ,
diag1_13 = (GCA_DIAG1)<13,1>% ,
diag1_14 = (GCA_DIAG1)<14,1>% ,
diag1_15 = (GCA_DIAG1)<15,1>% ;

```

MACRO

```
diag2_0 = (GCA_DIAG2)<0,1>%
diag2_1 = (GCA_DIAG2)<1,1>%
diag2_2 = (GCA_DIAG2)<2,1>%
diag2_3 = (GCA_DIAG2)<3,1>%
diag2_4 = (GCA_DIAG2)<4,1>%
diag2_5 = (GCA_DIAG2)<5,1>%
diag2_6 = (GCA_DIAG2)<6,1>%
diag2_7 = (GCA_DIAG2)<7,1>%
diag2_8 = (GCA_DIAG2)<8,1>%
diag2_9 = (GCA_DIAG2)<9,1>%
diag2_10 = (GCA_DIAG2)<10,1>%
diag2_11 = (GCA_DIAG2)<11,1>%
diag2_12 = (GCA_DIAG2)<12,1>%
diag2_13 = (GCA_DIAG2)<13,1>%
diag2_14 = (GCA_DIAG2)<14,1>%
diag2_15 = (GCA_DIAG2)<15,1>%;
```

MACRO

```
!
! The following are true if user:
gca_cmd_bar = (GCA_CMD_LSTBTS)<BAR_-->% ! did not forbid change bars on the command line.
gca_cmd_isq = (GCA_CMD_LSTBTS)<ISQ_-->% ! requested input sequence numbers.
gca_cmd_osq = (GCA_CMD_LSTBTS)<OSQ_-->% ! requested output line numbers.
gca_cmd_rit = (GCA_CMD_LSTBTS)<RIT_-->% ! said /RIGHT.
gca_cmd_und = (GCA_CMD_NBITS)<UND_-->% ! said /UNDERLINING.
gca_cmd_bld = (GCA_CMD_NBITS)<BLD_-->% ! said /BOLDING.
gca_cmd_ovr = (GCA_CMD_NBITS)<OVR_-->% ! said /OVERSTRIKING.
gca_cmd_ind = (GCA_CMD_NBITS)<IND_-->% ! said /INDEXING.
```

MACRO

```
gca_definition =
VECTOR [gca_size]
%;
```

