IBM i
7.2

Programming
DDS for Display files



Note

Before using this information and the product it supports, read the information in “Notices” on page 271.

This edition applies to IBM i 7.2 (product number 5770-SS1) and to all subsequent releases and modifications until otherwise indicated in new editions. This version does not run on all reduced instruction set computer (RISC) models nor does it run on CISC models.

This document may contain references to Licensed Internal Code. Licensed Internal Code is Machine Code and is licensed to you under the terms of the IBM License Agreement for Machine Code.

© Copyright International Business Machines Corporation 1999, 2013.
US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

DDS for display files 1
PDF file for DDS for display files 1
Defining a display file for DDS 1
Conventions and terminology used in the DDS information 2
Positional entries for display files (positions 1 through 44) 3
Positional entries for display files (positions 1 through 7) 3
Condition for display files (positions 7 through 16) 3
Specifying a condition for a field or for more than one keyword 4
Display size condition names 4
Type of name or specification for display files (position 17) 7
Reserved for display files (position 18) 7
Name for display files (positions 19 through 28) 7
Reference for display files (position 29) 8
Length for display files (positions 30 through 34) 10
Data type and keyboard shift for display files (position 35) 11
Keyboard types 11
Valid entries for display files 12
Decimal positions for display files (positions 36 and 37) 24
Usage for display files (position 38) 24
Location for display files (positions 39 through 44) 26
Line (positions 39 through 41) 27
Position (positions 42 through 44) 27
Beginning attribute character 27
Ending attribute character 27
Overlapping fields 27
Display length 28
DDS keyword entries for display files (positions 45 through 80) 29
ALARM (Audible Alarm) keyword for display files 29
ALIAS (Alternative Name) keyword for display files 29
ALTHELP (Alternative Help Key) keyword for display files 30
ALTNAME (Alternative Record Name) keyword for display files 31
ALTPAGEDWN/ALTPAGEUP (Alternative Page Down/Alternative Page Up) keyword for display files 31
ALWGPH (Allow Graphics) keyword for display files 33
ALWROL (Allow Roll) keyword for display files 34
ASSUME (Assume) keyword for display files 35
AUTO (Auto) keyword for display files 36
BLANKS (Blanks) keyword for display files 37
BLINK (Blink) keyword for display files 39
BLKFOLD (Blank Fold) keyword for display files 39
CAnn (Command Attention) keyword for display files 39
CFnn (Command Function) keyword for display files 41
CHANGE (Change) keyword for display files 43
CHCACCEL (Choice Accelerator Text) keyword for display files 44
CHCAVAIL (Choice Color/Display Attribute when Available) keyword for display files 45
CHCCTL (Choice Control) keyword for display files 47
CHCSLT (Choice Color/Display Attribute when Selected) keyword for display files 49
CHCUNAVAIL (Choice Color/Display Attribute when Unavailable) keyword for display files 50
CHECK (Check) keyword for display files 52
CHGINPDFT (Change Input Default) keyword for display files 61
CHKMSGID (Check Message Identifier) keyword for display files 63

CHOICE (Selection Field Choice) keyword for display files..... 64
CHRID (Character Identifier) keyword for display files.....66
CLEAR (Clear) keyword for display files.....66
CLRL (Clear Line) keyword for display files.....67
CMP (Comparison) keyword for display files.....69
CNTFLD (Continued-Entry Field) keyword for display files.....69
COLOR (Color) keyword for display files.....70
COMP (Comparison) keyword for display files..... 74
CSRINPONLY (Cursor Movement to Input-Capable Positions Only) keyword for display files.....75
CSRLOC (Cursor Location) keyword for display files..... 76
DATE (Date) keyword for display files.....77
DATFMT (Date Format) keyword for display files..... 78
DATSEP (Date Separator) keyword for display files..... 79
DFT (Default) keyword for display files.....80
DFTVAL (Default Value) keyword for display files..... 81
DLTCHK (Delete Check) keyword for display files..... 82
DLTEDT (Delete Edit) keyword for display files..... 82
DSPATR (Display Attribute) keyword for display files..... 83
DSPMOD (Display Mode) keyword for display files..... 90
DSPRL (Display Right to Left) keyword for display files..... 91
DSPSIZ (Display Size) keyword for display files..... 92
DUP (Duplication) keyword for display files..... 97
Programming for the Dup key.....98
EDTCDE (Edit Code) keyword for display files..... 99
EDTMSK (Edit Mask) keyword for display files..... 104
EDTWRD (Edit Word) keyword for display files..... 104
ENTFLDATR (Entry Field Attribute) keyword for display files..... 109
ERASE (Erase) keyword for display files.....110
ERASEINP (Erase Input) keyword for display files..... 111
ERRMSG (Error Message) and ERRMSGID (Error Message Identifier) keywords for display files...112
ERRSFL (Error Subfile) keyword for display files.....115
FLDCSRPRG (Cursor Progression Field) keyword for display files.....116
FLTFIXDEC (Floating-Point to Fixed Decimal) keyword for display files..... 116
FLTPCN (Floating-Point Precision) Keyword for Display Files..... 117
FRCDTA (Force Data) keyword for display files..... 118
GETRETAIN (Get Retain) keyword for display files 119
HELP (Help) keyword for display files..... 119
HLPARA (Help Area) keyword for display files..... 120
HLPBDY (Help Boundary) keyword for display files.....123
HLPCLR (Help Cleared) keyword for display files.....124
HLPCMDKEY (Help Command Key) keyword for display files.....124
HLPDOC (Help Document) keyword for display files 125
HLPEXCLD (Help Excluded) keyword for display files 127
HLPFULL (Help Full) keyword for display files..... 127
HLPID (Help Identifier) keyword for display files..... 128
HLPPNLGRP (Help Panel Group) keyword for display files.....128
HLPRCD (Help Record) keyword for display files..... 129
HLPRTN (Help Return) keyword for display files.....130
HLPSCHIDX (Help Search Index) keyword for display files.....131
HLPSEQ (Help Sequencing) keyword for display files..... 132
HLPTITLE (Help Title) keyword for display files.....132
HOME (Home) keyword for display files.....133
HTML (Hypertext Markup Language) keyword for display files..... 134
INDARA (Indicator Area) keyword for display files.....135
INDTXT (Indicator Text) keyword for display files..... 135
INVITE (Invite) keyword for display files..... 136
INZINP (Initialize Input) keyword for display files..... 137
ERASEINP(*ALL) keyword..... 139

INZRCD (Initialize Record) keyword for display files....140
KEEP (Keep) keyword for display files....141
LOCK (Lock) keyword for display files....141
LOGINP (Log Input) keyword for display files....142
LOGOUT (Log Output) keyword for display files....142
LOWER (Lower) keyword for display files....143
MAPVAL (Map Values) keyword for display files....143
MDTOFF (Modified Data Tag Off) keyword for display files....144
MLTCHCFLD (Multiple-Choice Selection Field) keyword for display files....145
MNUBAR (Menu Bar) keyword for display files....147
MNUBARCHC (Menu-Bar Choice) keyword for display files....148
MNUBARDSP (Menu-Bar Display) keyword for display files....151
MNUBARSEP (Menu-Bar Separator) keyword for display files....152
MNUBARSW (Menu-Bar Switch Key) keyword for display files....155
MNUCNL (Menu-Cancel Key) keyword for display files....156
MOUBTN (Mouse Buttons) keyword for display files....157
MSGALARM (Message Alarm) keyword for display files....159
MSGCON (Message Constant) keyword for display files....160
MSGID (Message Identifier) keyword for display files....161
MSGLOC (Message Location) keyword for display files....163
NOCCSID (No Coded Character Set Identifier) keyword for display files....164
OPENPRT (Open Printer File) keyword for display files....164
OVERLAY (Overlay) keyword for display files....165
OVRATR (Override Attribute) keyword for display files....166
OVRDTA (Override Data) keyword for display files....167
PAGEDOWN/PAGEUP (Page Down/Page Up) keywords for display files....167
PASSRCD (Passed Record) keyword for display files....168
PRINT (Print) keyword for display files....169
PROTECT (Protect) keyword for display files....171
PSHBTNCHC (Push Button Field Choice) keyword for display files....172
PSHBTNFLD (Push Button Field) keyword for display files....173
PULLDOWN (Pull-Down Menu) keyword for display files....175
PUTOVR (Put with Explicit Override) keyword for display files....176
PUTRETAIN (Put-Retain) keyword for display files....178
RANGE (Range) keyword for display files....179
REF (Reference) keyword for display files....180
REFFLD (Referenced Field) keyword for display files....181
RETKEY (Retain Function Keys) and RETCMDKEY (Retain Command Keys) keywords for display files....182
RETLCKSTS (Retain Lock Status) keyword for display files....182
RMVWDW (Remove Window) keyword for display files....182
ROLLUP/ROLLDOWN (Roll Up/Roll Down) keywords for display files....183
RTNCSRLOC (Return Cursor Location) keyword for display files....184
RTNDTA (Return Data) keyword for display files....186
SETOF (Set Off) keyword for display files....187
SETOFF (Set Off) keyword for display files....187
SFL (Subfile) keyword for display files....188
SFLCHCCTL (Subfile Choice Control) keyword for display files....189
SFLCLR (Subfile Clear) keyword for display files....190
SFLCSRPRG (Subfile Cursor Progression) keyword for display files....191
SFLCSRRRN (Subfile Cursor Relative Record Number) keyword for display files....191
SFLCTL (Subfile Control) keyword for display files....191
SFLDLT (Subfile Delete) keyword for display files....193
SFLDROP (Subfile Drop) keyword for display files....194
SFLDSP (Subfile Display) keyword for display files....195
SFLDSPCTL (Subfile Display Control) keyword for display files....195
SFLEND (Subfile End) keyword for display files....196
SFLENTER (Subfile Enter) keyword for display files....199

SFLFOLD (Subfile Fold) keyword for display files....200
SFLINZ (Subfile Initialize) keyword for display files....201
SFLLIN (Subfile Line) keyword for display files....202
SFLMLTCHC (Subfile Multiple Choice Selection List) keyword for display files....203
SFLMODE (Subfile Mode) keyword for display files....205
SFLMSG (Subfile Message) and SFLMSGID (Subfile Message Identifier) keywords for display files....206
SFLMSGKEY (Subfile Message Key) keyword for display files....208
SFLMSGRCD (Subfile Message Record) keyword for display files....209
SFLNXTCHG (Subfile Next Changed) keyword for display files....210
SFLPAG (Subfile Page) keyword for display files....211
SFLPGMQ (Subfile Program Message Queue) keyword for display files....213
SFLRCDNBR (Subfile Record Number) keyword for display files....215
SFLRNA (Subfile Records Not Active) keyword for display files....216
SFLROLVAL (Subfile Roll Value) keyword for display files....217
SFLRTNSEL (Subfile Return Selected Choices) keyword for display files....219
SFLSCROLL (Subfile Scroll) keyword for display files....219
SFLSIZ (Subfile Size) keyword for display files....220
SFLSNGCHC (Subfile Single Choice Selection List) keyword for display files....222
SLNO (Starting Line Number) keyword for display files....224
SNGCHCFLD (Single-Choice Selection Field) keyword for display files....225
SYSNAME (System Name) keyword for display files....227
TEXT (Text) keyword for display files....228
TIME (Time) keyword for display files....228
TIMFMT (Time Format) keyword for display files....229
TIMSEP (Time Separator) keyword for display files....230
UNLOCK (Unlock) keyword for display files....231
USER (User) keyword for display files....232
USRDFN (User-Defined) keyword for display files....232
USRDSPMGT (User Display Management) keyword for display files....233
USRRSTDSP (User Restore Display) keyword for display files....233
VALNUM (Validate Numeric) keyword for display files....234
VALUES (Values) keyword for display files....234
VLDCMDKEY (Valid Command Key) keyword for display files....235
WDWBORDER (Window Border) keyword for display files....236
WDWTITLE (Window Title) keyword for display files....239
WINDOW (Window) keyword for display files....242
WRDWRAP (Word Wrap) keyword for display files....245
DDS for 3270 remote attachment....246
System/36 environment considerations for display files....247
Keyword considerations for display files used in the System/36 environment....247
ALTNAME (Alternative Record Name) keyword....248
CHANGE record-level keyword....248
HELP and HLPRTN keyword....249
MSGID keyword....249
PRINT(*PGM) keyword....250
RETKEY (Retain Function Keys) and RETCMDKEY (Retain Command Keys) keywords....251
RETKEY keyword....251
RETCMDKEY keyword....251
Considerations for specifying RETKEY and RETCMDKEY keywords....251
USRDSPMGT (User Display Management) keyword....252
Unicode considerations for display files....253
Positional entry considerations for display files that use Unicode data....253
Keyword considerations for display files that use Unicode data (positions 45 through 80)....254
CCSID (Coded Character Set Identifier) keyword....254
Double-byte character set considerations for DDS....256
Positional entry considerations for display files that use DBCS....256
Length (positions 30 through 34)....256

Data type (position 35)....256
Decimal positions (positions 36 and 37)....257
Keyword considerations for display files that use DBCS....257
CNTFLD (Continued-Entry Field) keyword....258
GRDATR (Grid Attribute) keyword....259
GRDBOX (Grid Box) keyword....260
GRDCLR (Grid Clear) keyword....262
GRDLIN (Grid Line) keyword....264
GRDRCD (Grid Record) keyword....266
IGCALTTYP (Alternative Data Type) keyword....266
IGCCNV (DBCS Conversion) keyword....267
Additional considerations for describing display files that contain DBCS data....268

Notices....271
Programming interface information....272
Trademarks....272
Terms and conditions....273

Index....275

DDS for display files

You can use data description specifications (DDS) to define display files. This topic collection provides the information you need to code the positional and keyword entries that define these display files.

This information is only a quick reference for display file coding. For detailed information and examples about DDS for display files, see the Application Display Programming book. This book is helpful if you are just getting started with DDS for display files.

Note: By using the code examples, you agree to the terms of the "Code license and disclaimer information" on page 269.

PDF file for DDS for display files

You can view and print a PDF file of this information.

To view or download the PDF version of this document, select DDS for Display files.

Saving PDF files

To save a PDF on your workstation for viewing or printing:

1. Right-click the PDF link in your browser.
2. Click the option that saves the PDF locally.
3. Navigate to the directory in which you want to save the PDF.
4. Click **Save**.

Downloading Adobe Reader

You need Adobe Reader installed on your system to view or print these PDFs. You can download a free copy from the Adobe Web site (www.adobe.com/products/acrobat/readstep.html).

Defining a display file for DDS

When you specify positional entries for display files, you need to follow some specific rules for filling in positions 1 through 44 of the data description specifications (DDS) form.

Procedure

- Specify the entries in the following order to define a display file:
  1. File-level entries
  2. Record-level entries
  3. Help-level entries
  4. Field-level entries
- Specify at least one record format in the file.

  The maximum number of record formats in a display file is 1024. The maximum number of fields in any one record format is 32 763. The maximum number of fields that can be displayed per record is 4095. The maximum combined length of all named fields and indicators in a record format is 32 763 bytes, regardless of the usage (I, O, B, M, H, P). For more information, see "Usage for display files (position 38)" on page 24. Also, see the Application Display Programming book for the maximum number of input-capable fields.

© Copyright IBM Corp. 1999, 2013

Note: Specify the file name through the Create Display File (CRTDSPF) command, not through DDS.

Results

You can find an explanation of file-level, record-level, help-level, and field-level entries as well as syntax rules for specifying DDS keywords in Rules for DDS keywords and parameter values.

The following figure shows a display file example.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00100A* DISPLAY FILE EXAMPLE
00101A*
00102A                                      REF(PAYROLL)
00103A          R MENU
00104A          H                           HLPARA(1 1 12 80)
00105A                                      HLPRCD(RECORD1 FILEA)
00106A N01
00107AO 02        FLDA          20I 2O  2  2DSPATR(HI)
00108A            FLDB          22N 2B  3  2
00109A  72 73
00110AO 60 61 62
00111AA 63                                  DSPATR(HI)
00112A            FLDC           7Y 0B  7 20DSPATR(RI PC)
00113A  42 43
00114AO 60 61
00115AO 62                                   9  2'Constant'
00116A            FLDD      R           11  2
```

Figure 1. Display file example

Related reference

DDS keyword entries for display files (positions 45 through 80)
You type the keyword entries that define display files in positions 45 through 80 (functions).

Keyword considerations for display files used in the System/36 environment
You cannot specify some keywords in display files that contain the USRDSPMGT keyword.

Related information

Example: DDS for each file type

Conventions and terminology used in the DDS information

DDS information uses these conventions and terminology.

- A *keyword* is a name that identifies a function.
- A *parameter* is an argument shown between the parentheses on a keyword that identifies a value or set of values you can use to tailor the function the keyword specifies.
- A *value* is an actual value that you can use for a parameter.
- In the keyword descriptions, *this field* or *this record format* means the field or record format you are defining.
- The expression *use this file- or record-level keyword* means the keyword is valid only at the file or record level.
- *To specify a keyword* means to code the keyword in the DDS for a file. This contrasts with *to select a keyword* or *when a keyword is in effect*, which both mean that any conditioning (such as one or more option indicators) is satisfied when an application program issues an input or output operation.
- *Current source* or *source you are defining* means the DDS that together make up the description of one file.
- In sample displays, character fields are shown as Xs and numeric fields are shown as Ns.
- The 5250 Workstation Feature is a feature of the OS/2 communications manager that allows the personal computer to perform like a 5250 display station and use functions of IBM® i.
- *Logical file* includes join logical files, simple logical files, and multiple-format logical files.

- *Page* means to move information up or down on the display. *Roll* means the same as page. *Paging keys* are the same as *roll keys*. The PAGEDOWN keyword is the same as the ROLLUP keyword. The PAGEUP keyword is the same as the ROLLDOWN keyword.

Positional entries for display files (positions 1 through 44)

You specify positional entries in the first 44 positions of the data description specifications (DDS) form for display files.

Related reference

DDS keyword entries for display files (positions 45 through 80)

You type the keyword entries that define display files in positions 45 through 80 (functions).

Positional entries for display files (positions 1 through 7)

You can specify the sequence number, the form type, and comments in positions 1 through 7.

Sequence number for display files (positions 1 through 5)

Use these positions to specify a sequence number for each line on the form.

The sequence number is optional and is for documentation purposes only.

Form type for display files (position 6)

You can type an A in this position to designate this as a DDS form.

The form type is optional and is for documentation purposes only.

Comment for display files (position 7)

You can type an asterisk (*) in this position to identify this as a comment.

Use positions 8 through 80 for comment text. A blank line (no characters specified in positions 7 through 80) is also treated as a comment. Comments can appear anywhere in DDS and are kept only in the source file. Comments are printed on the source computer printout but are not printed on the expanded source computer printout.

Condition for display files (positions 7 through 16)

Positions 7 through 16 are a multiple-field area in which you can specify option indicators.

Option indicators are 2-digit numbers from 01 to 99. Your program can set option indicators on (hex F1) or off (hex F0) to select a field or keyword. You can use option indicators to select fields to display different data on different output operations instead of defining a different record format for each combination of fields.

A condition is a grouping by AND of two through nine indicators that must all be in effect before the field or keyword is selected. An AND condition is set off if N is specified and it is set on if N is not specified. You can specify a maximum of nine indicators for each condition and nine conditions for each field or keyword. Therefore, a maximum of 81 indicators can be specified for each field or keyword. An AND condition occurs when you specify a condition that requires more than one indicator must be on or off before the condition is satisfied. You can join the first indicator with the second, and the third, and so on, by AND to form a condition. These indicators must all be in effect before the condition is satisfied and the field or the keyword is selected. You must specify the field or the keyword on the same line as the last (or only) set of indicators specified.

You can also specify several conditions for a field or keyword such that if any one of them is satisfied, the field or the keyword is selected. This is called an OR relationship. You can join the first condition with the second condition, and the third condition, and so on, by OR to form an OR relationship. Conditions within the OR relationship can consist of just one indicator or of several indicators joined by AND. Indicators can

be joined by AND to form a condition. Conditions can be joined by OR to give your program several ways to select the field or keyword.

Position 7 (AND)
If you need more than three indicators to form an AND condition, specify the indicators on the next line or lines. You can specify an A in position 7 on the second or following lines to continue the AND condition, or you can leave it blank because A is the default.

Position 7 (OR)
If you specify several conditions that are to be joined by OR, each condition must start on a new line and each condition, except the first, must have an O in position 7. An O specified for the first condition produces a warning message, and that position is assumed to be blank.

Position 8, 11, 14 (NOT)
If you want an indicator to be off instead of on to satisfy a condition, specify an N in the position just preceding the indicator (position 8, 11, or 14).

Related reference
DSPSIZ (Display Size) keyword for display files
You use this file-level keyword to specify the display size to which your program can open the display file.

Specifying a condition for a field or for more than one keyword

If you specify a condition for a field, the field name (or the constant) and the last (or the only) indicator must be on the same line.

If you do not select the field for an output operation, no keywords specified for that field are in effect, regardless of how the condition for the keywords is specified. For example, in the following figure, FLDA is selected if either indicator 01 is off or indicator 02 is on. If FLDA is not selected, any keyword associated with that field, such as DSPATR(HI), is ignored.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00100A* DISPLAY FILE EXAMPLE
00101A*
00102A                                        REF(PAYROLL)
00103A          R MENU
00104A          H                             HLPARA(1 1 12 80)
00105A                                        HLPRCD(RECORD1 FILEA)
00106A N01
00107AO 02        FLDA          20I 2O  2  2DSPATR(HI)
00108A            FLDB          22N 2B  3  2
00109A  72 73
00110AO 60 61 62
00111AA 63                                    DSPATR(HI)
00112A            FLDC           7Y 0B  7 20DSPATR(RI PC)
00113A  42 43
00114AO 60 61
00115AO 62                                 9  2'Constant'
00116A            FLDD      R           11  2
```

Figure 2. Display file coding example

If you want to specify a condition for one or more keywords, the last (or only) indicator must appear on the same line as the keywords. If the condition applies to keywords on more than one line, you must use keyword continuation for the indicators to apply to all keywords. You can use a plus (+) or a minus (–) sign as a continuation character. For more information about keyword continuation, see Rules for DDS keywords and parameter values.

Display size condition names

If you want your program to open this file to display devices with display sizes other than 24 lines x 80 characters, specify the DSPSIZ (Display Size) keyword at the file level. You can then specify a condition for the use of keywords and the location of fields with the display size condition names specified for the DSPSIZ keyword.

If you do not specify the DSPSIZ keyword, your program can only open this file to display devices with a 24 x 80 display.

The following table shows the display size condition name for each display device.

Device	Display size	Display size condition name (see Note)
3179 3180 3196 3197 (Models C1 and C2) 3476 3487 (Models HA, HC, HG, and HW) 3488 (depending on the monitor that is attached to the display device) 3486 (Models BA and BG) 5251 (Models 11 and 12) 5291 5292	24 x 80 characters (1920 characters)	*DS3
3180 3197 (Models D1, D2, W1, and W2) 3477 (Models FA, FC, FD, and FG) 3487 (Models HA, HC, HG, and HW)	27 x 132 characters (3564 characters)	*DS4

Note: You can specify a user-defined display size condition name instead of *DS3 or *DS4. See "DSPSIZ (Display Size) keyword for display files" on page 92 for an explanation of how to specify user-defined condition names.

Figure 3 on page 5 shows how to specify the DSPSIZ keyword and display size condition names.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A                                                  1           2
00010A                                   DSPSIZ(27 132 *LARGE 24 80 *NORMAL)
00020A          R RECORDA
00030A            FIELDA        10  0   1  2
00040A            FIELDB        10  0   1120
00050A  *NORMAL                         1 49
00060A            FIELDC        10  0  27  1
00070A  *NORMAL                        15  1
     A
```

Figure 3. Specifying the DSPSIZ keyword and display size condition names

In Figure 3 on page 5, the display size condition name for the primary display size is defined as *LARGE **1** (column 52 to 64). The display size condition name for the secondary display size is defined as *NORMAL **2** (column 66 to 75). FIELDA appears on line 1, position 2 for both display sizes. FIELDB appears on line 1, position 120 for the primary display size (*LARGE by default), and on line 1, position 49 for the secondary display size (*NORMAL specified in positions 9 through 16). FIELDC appears on line 27, position 1 for the primary display size, and on line 15, position 1 for the secondary display size. Only secondary display sizes (in this example, *NORMAL) can be used to specify a condition for field locations.

Use display size condition names similar to the way you use option indicators, except that display size condition names do not appear in your program and do not appear in the output record. A display size condition is on if the display file is opened to the corresponding display size. When you use display size condition names, the following rules apply:

- Specify the DSPSIZ keyword to designate the primary display size and the secondary display size. If you do not specify the DSPSIZ keyword, the default is DSPSIZ(*DS3).
- You can specify only one display size condition name for a condition. You cannot specify AND or OR with other display size condition names or option indicators.
- The display size condition name must start in position 9.
- The display size condition name can be user-defined. See the keyword description for DSPSIZ (Display Size) keyword for display files for more details.
- You can specify N in position 8 to designate a NOT condition (for the primary display size).

  Note: Specifying N in position 8 implies an OR relationship between the remaining display size condition names. For example, N*DS4 implies *DS3 when *DS3 is specified as a secondary display size on the DSPSIZ keyword.

- You must not use display size condition names that alter the line or position sequence of a field within a record. Fields are ordered in the display file by primary locations. A severe error occurs at file creation time if the secondary location alters this primary sequence.

  For example, FLD1 and FLD2 are on the primary display. FLD1 is located on line 2, position 2 and FLD2 on line 4, position 2. You cannot use a display size condition name to display FLD2 before FLD1 on the display (on line 1) for a secondary display size.
- When you specify the location of a field on a secondary display size, you can only specify positions 8 through 16 (condition) and 39 through 44 (location).
- If you do not specify a condition name for a keyword for which condition names are valid, the primary condition name specified on the DSPSIZ keyword is the default.

Table 1 on page 6 shows the correct and incorrect combinations of display size condition names and primary display sizes, when both display sizes are specified on the DSPSIZ keyword and the first one specified varies.

Table 1. Valid display size condition specifications

Display size condition name[1]	**24 x 80 DSPSIZ(*DS3...) or DSPSIZ(24 80...) primary display size**	**27 x 132 DSPSIZ(*DS4...) or DSPSIZ(27 132...) primary display size**
*DS3	Error[2]	Valid
*DS4	Valid	Error[2]
N*DS3	Valid	Error[3]
N*DS4	Error[3]	Valid

Notes:

1. See the DSPSIZ keyword description for user-defined names for these display size condition names.
2. The display size condition names are in error because that display size is the primary display size.
3. These display size condition names are in error because a primary and a secondary location are implied for the same display size. A condition name specified with the NOT condition implies an OR relationship. For example, N*DS4 implies *DS3.

Figure 4 on page 6 and Figure 5 on page 6 show a display size condition for a keyword (in this case, MSGLOC, Message Location).

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00030A                                  DSPSIZ(*DS3 *DS4)
00040A  *DS4                            MSGLOC(26)
     A
```

Figure 4. Display size condition (example 1)

In Figure 4 on page 6, the display size condition name *DS4 is specified, so that the message line is line 26 for a 27 x 132 display and line 25 (the default) for a 24 x 80 display.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00080A                                  DSPSIZ(*DS4 *DS3)
00081A                                  MSGLOC(26)
     A
```

Figure 5. Display size condition (example 2)

In Figure 5 on page 6, the message line is also line 26 for the 27 x 132 display and line 25 (the default) for the 24 x 80 display, even though no display size condition name is specified, because the primary display size (*DS4) specified with the DSPSIZ keyword is the default.

Type of name or specification for display files (position 17)

You can specify a value in this position to identify the type of name in positions 19 through 28.

The valid entries for display files are:

Entry
Meaning

R
Record format name

H
Help specification

Blank
Field name

The example in "Defining a display file for DDS" on page 1 shows how to code the name type.

Related concepts
Name for display files (positions 19 through 28)
You use these positions to specify record format names and field names.

Related reference
HELP (Help) keyword for display files
You use this file-level or record-level keyword to enable the Help key.

Reserved for display files (position 18)

This position does not apply to any file type. Leave this position blank unless you use it for comment text.

Name for display files (positions 19 through 28)

You use these positions to specify record format names and field names.

The names must begin in position 19.

The example in "Defining a display file for DDS" on page 1 shows you how to specify record format names and field names.

Record format name

When you specify R in position 17, the name specified in positions 19 through 28 is a record format name. You can specify more than one record format for a display file, but each record format name must be unique within that file.

Field name

When position 17 is left blank, the name you specified in positions 19 through 28 is a field name. Field names must be unique within the record format.

Constant fields

Constant fields are unnamed fields (positions 19 through 28 must be blank). The following rules apply to constant fields:

- Positions 17 through 38 must be blank.
- The location of the field is required (positions 39 through 44).
- The field can be conditioned using option indicators (positions 7 through 16).
- You can specify secondary display locations using display size condition names (positions 8 through 16). Only the display size condition name and location can be specified. That is, positions 7, 17 through 38, and 45 through 80 must be blank.

- The constant itself is defined in positions 45 through 80 using one of the following entries:
  - Explicit DFT keyword (specify the value within single quotation marks with the DFT keyword)
  - Implicit DFT keyword (specify the value within single quotation marks without the DFT keyword)
  - DATE keyword (specify no value; see the DATE keyword description)
  - TIME keyword (specify no value; see the TIME keyword description)
  - SYSNAME keyword (specify no value; see the SYSNAME keyword description)
  - USER keyword (specify no value; see the USER keyword description)
  - MSGCON keyword (specify the message description, the message file, the library name, and the length of the message description)

How to determine the order of fields in a record format

The order of the name fields that you specify in a record format is the order in which the name fields appear in your program when it is compiled. (Unnamed fields do not appear in your program.)

The locations of named and unnamed fields you specify in positions 39 through 44 determine the order that the fields appear in the display. Hidden fields (H in position 38) and Program-to-System fields (P in position 38) do not appear on the display.

Related concepts

Location for display files (positions 39 through 44)
You use these positions to specify the exact location on the display where each field begins.

Related reference

Type of name or specification for display files (position 17)
You can specify a value in this position to identify the type of name in positions 19 through 28.

Rules for DDS keywords and parameter values

Reference for display files (position 29)

You can specify R in this position to use the reference function of the IBM i operating system. This function copies the attributes of a previously defined, named field (called the *referenced field*) to the field you are defining.

The referenced field can be previously defined in either the display file you are defining or a previously created database file (the database file to be referred to is specified with the REF or REFFLD keyword). The field attributes referred to are the length, data type, and decimal positions of the field, as well as the following keywords:

- ALIAS (alternative name)
- CCSID (Coded Character Set Identifier)
- FLTPCN (floating-point precision)
- TEXT
- DATFMT
- DATSEP
- TIMFMT
- TIMSEP
- Editing and validity checking keywords

If you do not specify R, you cannot use the reference function for this field and you must specify field attributes for this field.

Position 29 must be blank at the file, record, and help levels.

If the name of the referenced field is the same as the field you are defining, you need only specify R in position 29 (in addition to specifying the name of the field you are defining in positions 19 through 28). If

the name of the field you are defining is different from the name of the referenced field you must specify the name of the referenced field with the REFFLD (Referenced Field) keyword.

You can specify the name of the file defining the referenced field as a parameter value with the REF (Reference) or the REFFLD keyword.

You do not need to copy all attributes from the previously defined field to the field you are defining. To override specific attributes of the referenced field, specify those attributes for the field you are defining as follows:

- To override the EDTCDE (Edit Code) or EDTWRD (Edit Word) keywords, specify EDTCDE or EDTWRD for the field you are defining. You can delete these keywords by specifying the DLTEDT (Delete Edit) keyword for the field you are defining.
- To override the CHECK (Check), COMP (Comparison), RANGE (Range), and VALUES (Values) validity checking keywords and the CHKMSGID (Check Message Identifier) keyword, specify any validity checking keyword for the field you are defining. You can delete these keywords by specifying the DLTCHK (Delete Check) keyword for the field you are defining.

When you override some specifications, others are also affected:

- If you specify keyboard shift attribute, field length, or decimal positions for the field you are defining, neither editing nor validity checking keywords are copied from the referenced field.
- If you override the previously defined data type to character (by specifying M, A, X, or W in position 35), decimal positions are not copied. However, if you specify N, D, or I in position 35 and leave blanks in positions 36 and 37 (decimal positions), the field you define has the decimal positions of the referenced field. For D, the decimal positions must be zero.
- Packed decimal and binary fields are not supported for display files. Therefore, when you refer to fields of these types, the data type assigned is zoned decimal with a keyboard shift as follows:
  - If editing is in effect for the field you are defining, the keyboard shift is numeric only (Y in position 35).
  - If no editing is in effect for the field you are defining, the keyboard shift is signed numeric (S in position 35).
- When the referenced field contains the REFSHIFT (Reference Shift) keyword, the value specified for REFSHIFT is used as the display file keyboard shift. However, if the data type specified for the new field is not compatible with the keyboard shift specified on the REFSHIFT keyword, the keyword is not copied to the new field.

Note: After the display file is created, you can delete or change the referenced file without affecting the field descriptions in the display file. Delete and create the display file again to incorporate changes made in the referenced file.

Related concepts

Length for display files (positions 30 through 34)
You must specify a length for each named field unless you are copying the length from a referenced field.

Related reference

REF (Reference) keyword for display files
You use this file-level keyword to specify the name of a file from which field descriptions are to be retrieved. You can also use this keyword when you want to duplicate descriptive information from several fields in a previously described record format.

REFFLD (Referenced Field) keyword for display files
You use this field-level keyword to refer to a field when the name, record format, file, or library of the referenced field differs from its equivalent in positions 19 through 28.

Related information

When to specify REF and REFLD keywords for DDS files

Length for display files (positions 30 through 34)

You must specify a length for each named field unless you are copying the length from a referenced field.

The length is the number of bytes of data to pass to or receive from your program when I/O operations are done for the field. This is called the *program length* of the field.

The length of a field when it appears on the display is called the *display length*. The display length is greater than or equal to the program length. The display length of a field is determined by the keyboard shift (specified in position 35) and other field specifications, such as decimal positions (positions 36 and 37), and editing functions.

The display length does not include beginning and ending attribute characters of a field. However, you must consider these attribute characters when planning the display layout for field locations. Within a record, the ending attribute character of a field can overlap the beginning attribute character of the next field, requiring only one space between fields.

The maximum length of a character field is equal to the display size minus one. (This allows space for the beginning attribute character.) The maximum length of a numeric (zoned decimal) field is 63 positions. The maximum length of a single-precision floating-point field is 9 digits. The maximum length of a double-precision floating-point field is 17 digits.

You must not specify a field length for a constant field. See "DATE (Date) keyword for display files" on page 77, "DFT (Default) keyword for display files" on page 80, "MSGCON (Message Constant) keyword for display files" on page 160, or "TIME (Time) keyword for display files" on page 228 for information about the lengths of constant fields.

If you specify length, it must be right-aligned. Leading zeros are optional. Figure 6 on page 10 shows incorrect and correct field length specifications.

```
|...+....1....+....2....+....3....+....4....+....5
00010A            FIELD1       7
     A
00020A            FIELD2         7
     A
00030A            FIELD3    R   +7
     A
```

Figure 6. Incorrect and correct length specifications

Note: FIELD1 shows the field length specified incorrectly. FIELD2 and FIELD3 show the field length specified correctly.

If you use a referenced field, you can override the length of the field by specifying a new value or by specifying the increase or decrease in length. To increase the length, specify *+n*, where n is the increase. To decrease the length, specify *-n*, where n is the decrease. For example, an entry of +4 for a numeric field indicates a field that is 4 digits longer than the referenced field.

In some cases, some keywords specified with the field in the database file are not included in the display file if you specify a value for length.

A field cannot occupy the first position on the display. The first position is reserved for an attribute character. For example, on a 24 by 80 display, an entry of 1 in positions 39 through 41 (line), and 1 in positions 42 through 44 (position) is not allowed for a signed numeric field since the field starts in position 1.

Related concepts

Data type and keyboard shift for display files (position 35)
The entry you make in position 35 is the data type and keyboard shift attribute for display files.

Location for display files (positions 39 through 44)
You use these positions to specify the exact location on the display where each field begins.

Reference for display files (position 29)

You can specify R in this position to use the reference function of the IBM i operating system. This function copies the attributes of a previously defined, named field (called the *referenced field*) to the field you are defining.

Data type and keyboard shift for display files (position 35)

The entry you make in position 35 is the data type and keyboard shift attribute for display files.

This entry does not determine the data type of the field used in your program. The entry in positions 36 and 37 (decimal positions) determines the data type of the field.

The keyboard shift attribute automatically shifts 5250 workstations with data-entry keyboards and can, for all keyboards, limit what the workstation user can type into a field. However, the keyboard shift attribute does not shift 5250 workstations with the typewriter-like keyboard. Nor does the keyboard shift attribute restrict in any way what your program can write to a field. Your program can write alphabetic characters to a numeric field and, in most cases, read that field and receive those characters. Any restrictions are enforced solely by the programming language used for your program.

Related concepts

Length for display files (positions 30 through 34)
You must specify a length for each named field unless you are copying the length from a referenced field.

Decimal positions for display files (positions 36 and 37)
You use these positions to specify the decimal placement within a zoned decimal field and also to specify the data type of the field as it appears in your program.

Keyboard types

There are two types of keyboards on the IBM i platform: a *typewriter-like keyboard* and a *data-entry keyboard*.

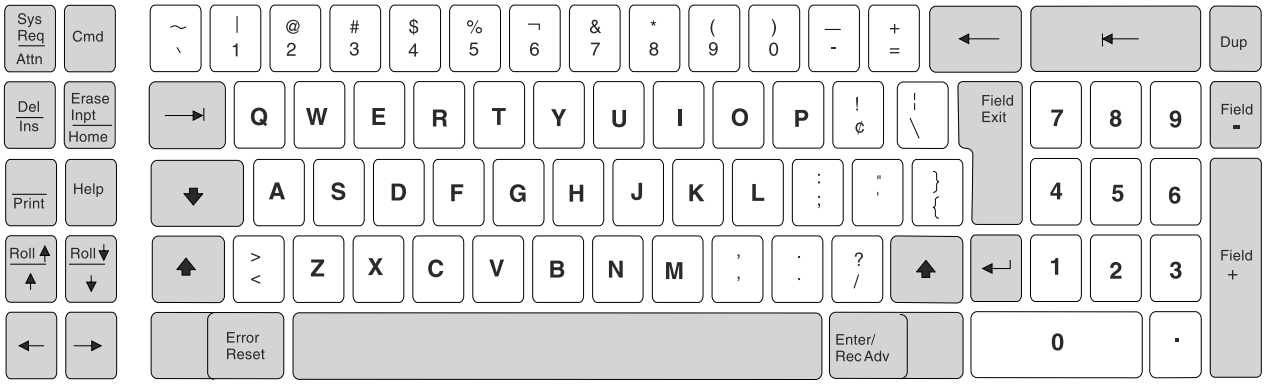
Display stations can have either the typewriter-like or the data-entry keyboard.

Typewriter-like keyboard
The typewriter-like keyboard functions in either uppershift or lowershift.

Type the upper symbol (for the keys with two symbols) when the keyboard is in uppershift. Type the lower symbol (for the keys with two symbols) when the keyboard is in lowershift. Type uppercase characters for alphabetic keys (which have only one symbol) when the keyboard is in uppershift. Type uppercase alphabetic characters when the keyboard is in lowershift, unless the Check Lowercase (CHECK(LC)) keyword is specified. If the CHECK(LC) keyword is specified, and you place (or leave) the keyboard in lowershift, you can type lowercase a through z characters.

Note: None of the keyboard shift attributes causes an automatic uppershift for the typewriter-like keyboard.

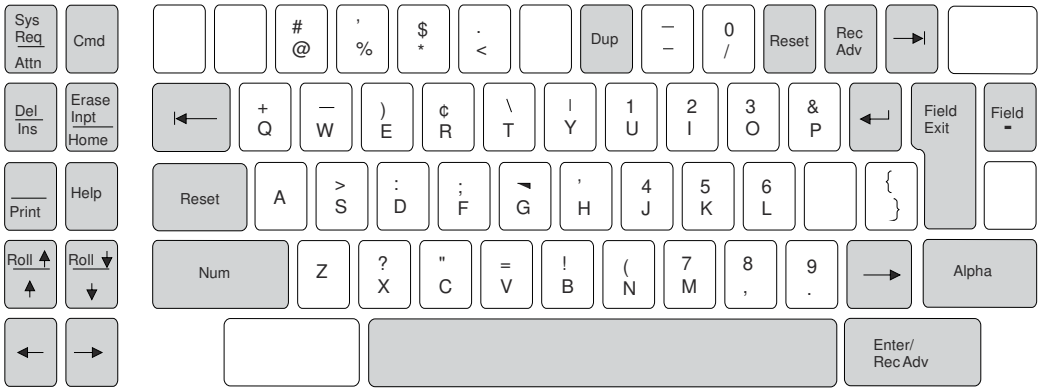
The following figure shows the typewriter-like keyboard.



Data-entry keyboard
The data-entry keyboard functions in either numeric shift (upper) or alphabetic shift (lower).

The upper symbol (for the keys with two symbols) is typed when the keyboard is in uppershift. On this keyboard, the numbers 0 through 9 are the upper symbols on alphabetic keys. The lower symbol (for the keys with two symbols) is typed when the keyboard is in lowershift. The alphabetic characters A through Z are the lower symbols and are always in uppercase. The data-entry keyboard does not support lowercase characters a through z, even if you specify CHECK(LC) keyword.

The following figure shows the data-entry keyboard.



RSLL616-0

Note that you type the numbers 0 through 9 by using the lowershift on the typewriter-like keyboard and by using the uppershift (numeric) on the data-entry keyboard. Therefore, when a field has one of the numeric keyboard shift attributes (numeric shift or numeric only), the typewriter-like keyboard is in lowershift and the data-entry keyboard is in uppershift. In both cases, you can type numeric characters without pressing a shift key.

Valid entries for display files
These entries are valid for display files.

Entry keyboard shifts	**Meaning**	**Data type permitted**
Blank	Default	
X	Alphabetic only	Character
A	Alphanumeric shift	Character
N	Numeric shift	Character or numeric
S	Signed numeric	Numeric
Y	Numeric only	Numeric
W	Katakana (for Japan only)	Character
I	Inhibit keyboard entry	Character or numeric
D	Digits only	Character or numeric
M	Numeric only character	Character
Data type (see note)		
F	Floating point	Numeric
L	Date	
T	Time	
Z	Timestamp	

Note: The data types J (only), E (either), O (open), and G (graphic) support DDS display files that use DBCS. The G (graphic) data type also supports DDS display files that use UTF-16 and UCS-2.

The examples in "Defining a display file for DDS" on page 1 and "Date (L), Time (T), and Timestamp (Z)" on page 19 show how to specify the keyboard shift attribute.

The keyboard shift attributes are defined in detail in the following topics.

Related concepts

Positional entry considerations for display files that use Unicode data
Be aware of these positional entry considerations for display files that use Unicode data. Positions not mentioned have no special considerations for Unicode.

Related reference

Data type (position 35)
You can specify the data type in this position by typing J, E, O, G.

Default (blank)
If you leave position 35 blank, the entry in positions 36 and 37 (decimal positions) determines the data type of the field.

- If you make a valid entry in positions 36 and 37, the data type is zoned decimal and the keyboard shift attribute is signed numeric (S) unless you also specify an editing keyword. The keyboard shift attribute is numeric only (Y) when you also specify an editing keyword.
- If you make no entry in positions 36 and 37, the data type is character and the keyboard shift attribute is alphanumeric shift (A).

If you specify the REFSHIFT keyword for a referenced field, the specified value is used. Otherwise, a data type of packed or binary is converted to zoned decimal in the display file. Conversion to or from packed or binary can occur within your program.

Alphabetic only (X)
Both types of keyboards are in lowershift. Only the characters A through Z, comma (,), period (.), dash (–), and space ( ) can be typed in.

When you type lowercase characters a through z, uppercase characters are sent to the program. See "CHECK (Check) keyword for display files" on page 52 for information about how to permit typing in lowercase characters for the typewriter-like keyboard on the 5250 workstation.

Alphanumeric shift (A)
Both types of keyboards are in lowershift. All characters are valid for entry.

Numeric shift (N)
To allow numeric entry, you must use uppershift for the data-entry keyboard and lowershift for the typewriter-like keyboard. All characters are valid for entry.

The display length for a numeric shift field is one more than the length coded in positions 30 through 34 when the following conditions occur:

- The field is an un-edited, input-capable field.
- The value in the decimal positions field is greater than zero.

The extra position in the display length is for the decimal point.

Note: Numeric-shift fields with specified decimal positions (in positions 36 and 37) are processed as numeric-only fields by data management during input, except that editing is not supported.

Related concepts

Numeric only (Y)

You can only type the numbers 0 through 9, plus (+), minus (-), period (.), comma (,), and space ( ) into the field. You can press any key to leave the field.

Signed numeric (S)
You can only type the numbers 0 through 9 into the field (no blanks, no plus sign, no minus sign).

To leave the field, press the Field Exit key, the Field+ key, the Field- key, or a cursor movement key. If you do not type any data into the field, you can press the Enter key.

You should consider the following differences when you choose between signed numeric (S) and numeric only (Y):

- Signed numeric restricts the characters that you can type into the field to the numbers 0 through 9.
- You cannot specify S in position 35 if you also specify the EDTCDE or EDTWRD keyword.
- Numeric-only performs character removal to remove nonnumeric characters; signed numeric prevents you from typing in these characters at all.

For input-capable fields only, the display length for the field is one more than the length specified in positions 30 through 34. The farthest right position on the display is reserved for a minus sign.

The following considerations apply when the IBM i operating system passes the contents of a signed numeric field to your program:

- Your program always sees a numeric, right-aligned, zero-filled field.
- The field is displayed as a right-aligned, blank-filled field unless you specify CHECK(RZ). If you specify CHECK(RZ), the field is displayed as right-adjusted and zero-filled.
- The IBM i operating system does not perform decimal alignment.
- The IBM i operating system does not remove characters from the field (as it does for numeric only fields).

When an input-capable signed numeric field displays and you do not specify CHECK(RZ), the IBM i operating system performs zero suppression by default (the EDTCDE and EDTWRD keywords are not valid for signed numeric fields).

Negative numbers are handled as follows:

- On input, you must type the number and press the Field- key. The number is right-aligned in the displayed field with a minus sign in the farthest right position. The IBM i operating system converts the farthest right significant digit to hex Dn where n is the significant digit, before passing the number to your program. For example, if you type 12345 and press the Field- key, 12345- is displayed and your program sees X'F1F2F3F4D5'.
- On output, the IBM i operating system converts hex D in the farthest right digit to hex F. This changes the negative number to a positive number for display purposes and displays a minus sign in the farthest right (additional) position in the displayed field. For example, if your program sees X'F1F2F3F4D5', the number appears on the display as 12345-.

For examples of signed numeric fields and sample data typed into them, see the example in “Date (L), Time (T), and Timestamp (Z)” on page 19.

Numeric only (Y)
You can only type the numbers 0 through 9, plus (+), minus (-), period (.), comma (,), and space ( ) into the field. You can press any key to leave the field.

The display length for a numeric-only field is one more than the program length when both of the following conditions occur (the program length is specified in positions 30 through 34):

- The field is an unedited, input-capable field.
- The value in positions 36 and 37 (decimal positions) is greater than zero.

The extra position in the display length is for the decimal point.

When the IBM i operating system passes the contents of the field to your program, the following considerations apply:

- Your program sees a numeric, decimally aligned field.
- To type digits to the right of the decimal, positions 36 and 37 must be greater than zero and you must type the decimal character.
- You cannot type the maximum number of digits, a decimal character, and a sign character, because the display length of the field equals only the program length plus one. You can press the Field+ key or the Field- key to avoid typing a sign character.
- The IBM i operating system removes all characters except 0 through 9 (whether typed or supplied through the EDTWRD keyword) and the sign.
- The IBM i operating system converts embedded blanks (hex 40) to zeros (hex F0) before decimal alignment. (Embedded blanks are blanks between any significant digits in the field.) Leading blanks, trailing blanks, zeros, plus signs, and minus signs are not treated as significant digits. Embedded ampersands in an edit word are also converted to zeros before decimal alignment.
- All nonnumeric characters are removed before decimal alignment and validity checking for the RANGE, COMP, CMP, VALUES, CHECK(VN), CHECK(M10), CHECK(M11), and CHECK(VNE) keywords. Numeric characters (0 through 9) supplied by the EDTWRD keyword are not removed. Validity checking for the CHECK(M10F) and CHECK(M11F) keywords is performed before the nonnumeric characters are removed.
- The field length in the input buffer is the program length.

When the IBM i operating system displays a numeric-only field, the EDTCDE or the EDTWRD keyword, if specified, applies. You can specify EDTCDE and EDTWRD only for numeric-only fields. The display length equals the program length plus the editing characters from the specified edit code or edit word.

Negative numbers are handled as follows:

- The user can type a negative number on input in one of two ways:
  - Type the digits, then a minus. The minus sign (-) appears (hex 60) on the display where it was typed in.
  - Type the digits, then press the Field- key.

    If you do not specify CHECK(RZ) or CHECK(RB), a brace (}) is displayed in the farthest right position. This causes an error message to appear at the workstation if you specify decimal positions other than zero in positions 36 and 37. If you specify CHECK(RZ) or CHECK(RB), the digits typed in are right-aligned. No minus sign appears in either case.

    If you specify an EDTCDE keyword that displays a minus sign and you do not specify CHECK(RZ) or CHECK(RB), a brace (}) is displayed in the farthest right position. This does not cause an error message to appear at the workstation. A minus sign appears in the farthest right position on output.

    If you specify an EDTCDE keyword that displays a minus sign and you also specify CHECK(RZ) or CHECK(RB), the farthest right significant digit is displayed as hex Dn (negative). A minus sign appears on the output.

    When a negative number passes to your program, the IBM i operating system converts the farthest right significant digit from hex Fn (positive) to hex Dn (negative), where n is the significant digit.
- The sign appears in the farthest right display position on output and takes up one of the positions in the display length.

Note: The IBM i program examines each character of a numeric-only field to remove the nonnumeric characters plus sign (+), minus sign (-), comma (,), and decimal point(.) and nonsignificant digits, and to convert embedded blanks to zeros. This examination and removal can delay response time if a significant number of fields must be processed on an input operation.

Related concepts

Numeric shift (N)

To allow numeric entry, you must use uppershift for the data-entry keyboard and lowershift for the typewriter-like keyboard. All characters are valid for entry.

Related reference

EDTCDE (Edit Code) keyword for display files
You use this field-level keyword to edit output-capable numeric fields.

EDTWRD (Edit Word) keyword for display files
You use this field-level keyword to specify an edit word if you cannot obtain the editing that you want through the EDTCDE keyword.

Katakana (W)
This field attribute designates the Japanese Katakana keyboard shift. All characters are valid for entry.

Inhibit keyboard entry (I)
A field with this keyboard shift attribute does not accept data typed in from the keyboard, and an error is issued if you press any keys.

You can press the field advance key to position the cursor at the start of the field. This field can be used to allow input from feature devices such as a light pen. The Field+, Field Exit, and Dup keys are valid for a field with this attribute, and function the same as if pressed in any input field for which the Display Attribute Protect (DSPATR(PR)) keyword is not in effect.

The display length for an inhibit keyboard entry field is one position greater than the length coded in positions 30 through 34 when the following conditions occur:

- The field is an un-edited, input-capable field.
- The value in the decimal positions field is greater than zero.

The extra position in the display length is for the decimal point.

Digits only (D)
To allow numeric entry, you must use uppershift for the data-entry keyboard and lowershift for the typewriter-like keyboard.

The numbers-only keyboard shift defines a character or numeric field that allows you to key only the digits 0 through 9 into the field. You cannot key special characters or blanks.

The numbers-only keyboard shift is supported only on devices that are configured on the 6040 or 6041 local controller or the 5294 or 5394 Control Unit. When a digits-only field is sent to a device that is not configured on a valid controller type, the field is processed as alphanumeric (keyboard shift A). Because you can type any alphanumeric character into the field, a decimal data error can result in the application program.

The Field Exit, Field+ Exit, and Dup keys are allowed. The Field+ Exit is processed as an unsigned Field Exit. The Field- Exit key is not allowed.

Blank and zero are the only supported values for decimal positions (DDS positions 36 and 37). If positions 36 and 37 are blank, the field is considered a character field. If you specify zero, the field is considered numeric.

You can only enter a positive integer value into a D field.

The display length of a digits-only field is always the field length as specified in positions 30 through 34.

Zero suppression is not supported for digits-only fields. EDTCDE and EDTWRD keywords are not valid, and the IBM i operating system does not perform zero suppression by default, as it does for signed-numeric fields.

You cannot enter blanks into the field. However, you can move the cursor out of the field after entering part of it. If you move the cursor after entering part of the field, when the IBM i operating system passes the contents of the field to a program, the following considerations apply:

- For a numeric field, leading blanks are converted by the IBM i operating system to zero, and the field is right-aligned before the programme passes the field contents to the application program.

- For a character field, blanks are passed to the application program as hex 40, and the field is not right-aligned.
- The field length in the input buffer is the program length.

In a database file, you can specify the D keyboard shift on the REFSHIFT keyword if the field data type is numeric or character (S, B, P, or A). For a numeric field, the number of decimal positions must be zero.

Numeric only character (M)
The M keyboard shift defines a character field that allows you to type only the digits 0 through 9, plus (+), minus (-), comma (,), period (.), and blank into the field.

The Field Exit, Field+ Exit, Field- Exit, and Dup keys are allowed. The Field+ Exit is processed as an unsigned Field Exit. The Field- Exit is processed as follows:

- If you do not specify CHECK(RZ) or CHECK(RB), the farthest right position is changed to a brace.
- If you specify CHECK(RZ) or CHECK(RB), the last character you type into the field must be a digit (or a keyboard error is issued). The farthest right digit, n, is converted from hex Fn (positive) to hex Dn (negative).

The display length for an M field is the length coded in positions 30 through 34. You must include any additional positions needed for a sign character or decimal point in the field length.

The field displays as a blank-filled field when you do not specify any keyword. If you specify a CHECK(RZ) keyword, the field displays as right-aligned zero. If you specify a CHECK(RB) keyword, the field displays as right-aligned blank filled.

When the IBM i operating system passes the contents of the field to a program, the following considerations apply:

- The program always sees a character field.
- The field length in the input buffer is the program length.
- The field contents are passed directly to the program. The IBM i operating system neither converts embedded blanks to zeros nor removes nonnumeric characters, such as sign characters and decimal points.

In a database file, you can specify the M keyboard shift on the REFSHIFT keyword if the field data type is character (A).

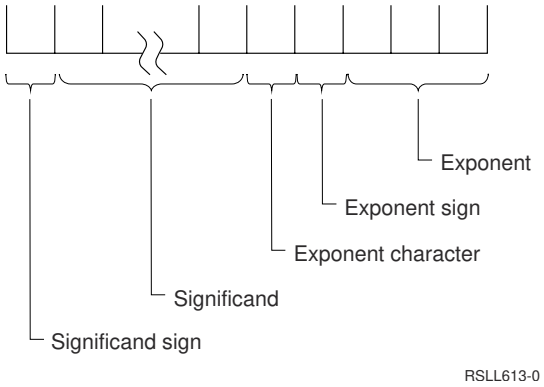
Floating point (F)
You can type any combination of characters in a floating-point field (but only the digits 0 through 9, sign characters (+ or -), E or e, decimal point (.), and comma (,) are valid).

An error message is issued if you type any other character in a floating-point field.

A floating-point value consists of five parts:

- Significand sign
- Significand
- Exponent character
- Exponent sign
- Exponent

The following figure shows the five parts of a floating-point value.



The parts in the figure are as follows:

- Significand sign
  - Use + for a positive value and - for a negative value.
  - On output, the significand sign is not displayed for a positive value.
  - On input, the significand sign is optional. If you do not type a + or -, the significand is assumed to be positive.
- Significand
  - The digits 0 through 9 and a decimal point (.) or a comma (,) are valid.
  - On output, the number of digits in the significand is determined by the length specified (positions 30 through 34). The location of the decimal point or the comma is determined by the decimal positions you specify (positions 36 and 37).
  - On input, you must type the significand. Only digits 0 through 9 are valid. The decimal point or comma is optional. If you do not specify a decimal point, a decimal point is assumed on the right.
- Exponent character
  - E or e are valid.
  - On output, the exponent character always displays.
  - On input, you must type an exponent character if the floating-point value includes an exponent.
- Exponent sign
  - Use + for a positive value and - for a negative value.
  - On output, the exponent sign always displays.
  - On input, the exponent sign is optional. If you do not type a + or a -, the exponent is assumed to be positive.
- Exponent
  - The digits 0 through 9 are valid.
  - On output, the exponent is always 3 digits.
  - On input, you must type at least 1 digit if you type the exponent character (E or e). You can type a maximum of 3 digits.

Notes:

1. When a floating-point value displays, embedded blanks are removed. On input, you can type blanks before or after a floating-point value. Within a floating-point value, blanks are allowed between the significand and the exponent character.
2. If you do not type a value in a displayed floating-point field, a positive zero is assumed.
3. A value of negative zero is valid in a floating-point field. Only the first zero to the left of the decimal point displays. A minus sign displays to the left of the first zero.

4. A value of positive zero is valid in a floating-point field. The significand sign (+) does not display. Only the first zero to the left of the decimal point displays.
5. You can type a fixed-point value in a floating-point field.

The display length for a floating-point field is seven positions greater than the length specified in positions 30 through 34. The seven extra positions are for the significand sign, the decimal point or comma, the exponent character, the exponent sign, and the three exponent digits. "Date (L), Time (T), and Timestamp (Z)" on page 19 describes how the data you type is passed to your program.

Date (L), Time (T), and Timestamp (Z)
Both types of keyboards are in lowershift. All characters are valid for entry.

The field length (DDS positions 30 and 34) for these data types are always blank. The following rules determine the field length:

- For the date (L) data type, the format specified on the DATFMT keyword dictates the length of the field. If you do not specify the DATFMT keyword, then the format is set to *ISO as default, which has a field length of 10. If you specify DATFMT(*JOB), the field length will always be 10, even if the Job Date Format Definition Attribute displays an 8 character date.
- For the time (T) data type, the format specified on the TIMFMT keyword dictates the length of the field. All formats for the TIMFMT keyword, including the default of *ISO, have field lengths of 8.
- For the timestamp (Z) data type, the field length is 26. The format of a timestamp field is

  ```
  yyyy-mm-dd-hh.mm.ss.mmmmmm
  ```

  Where yyyy = year, mm = month, dd = day, hh = hour, mm = minute, ss = second, and mmmmmm = microsecond.

Decimal positions (DDS positions 36 and 37) support only values of period (.). Valid field usage (DDS position 38) can be O, B, or I.

It is the responsibility of the high-level language and the application to format the date, time, and timestamp fields correctly on output. The system does not format fields on output. Date and time fields should be formatted according to the formats of the DATFMT and TIMFMT keywords and should use the separators specified for the DATSEP and TIMSEP keywords. You should use the standard timestamp format (*yyy-mm-dd-hh.mm.ss.mmmmmm*) for timestamp fields.

The system validates date-, time-, and timestamp-capable fields on input when the modified data tag (MDT) for a field is set to the on position. You can turn on the MDT for a field by either typing into the field or by specifying DSPATR(MDT) on the field. If the MDT for a field is turned off, the saved contents of the field return to the application. When the MDT is on for a field, date and time fields are evaluated according to the following items:

- The format specified on the DATFMT and TIMFMT keywords.
- The separators specified on the DATSEP and TIMSEP keywords.

Timestamp fields are evaluated according to the standard timestamp format *(yyyy-mm-dd-hh.mm.ss.mmmmmm)*.

You can enter date, time, and timestamp field values with or without separators. When you enter a value without separators, leading zeros are inserted when necessary. The system includes the separators in the data that are passed back to the application. When you enter a value with separators, leading zeros are inserted up to the first separator when necessary. A value that is entered with separators cannot start with a separator. Leading and trailing blanks are ignored.

You can enter timestamp field values with or without separators. The system inserts leading or trailing zeros for timestamp fields. If you enter the field with separators, you must enter 20 digits and 6 separator characters.

You can enter the following field level keywords with these data types:

ALIAS	ERRMSGID

CHANGE

CHGINPDFT

CHRID

COLOR

DATFMT (L)

DATSEP (L)

DFT

DFTVAL

DLTEDT

DLTCHK

DSPATR

ENTFLDATR

ERRMSG

FLDCSRPRG

INDTXT

MAPVAL

NOCCSID

OVRATR

OVRDTA

PUTRETAIN

REFFLD

SFLCSRPRG

TEXT

TIMFMT (T)

TIMSEP (T)

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00100A*
00200A* KEYBOARD SHIFT ATTRIBUTES
00300A            CHARA          5   I  2  2
00400A            CHARB          5   I  3  2CHECK(RB)
00500A            CHARC          5   I  4  2CHECK(RZ)
00600A            CHARD          5X  I  4 30
00700A            CHARE          5M  I  4 40
00800A            CHARF          5D  I  4 50
00900A            SIGN1          5  OI  5  2
01000A            SIGN2          5  2B  5 30
01100A            NBR1           5Y OI  6  2
01200A            NBR2           5Y 2I  6 15
01300A            NBR3           5Y 2B  6 30EDTCDE(L)
01400A            NBR4           5N 2I  6 40
01500A            NBRZ           5   H
01600A            FLPT           7F 4I  7  2
01700A            DATE            L  B  7 30DATFMT(*JUL) DATSEP('/')
01750A            DATE1           L  B  7 40DATFMT(*MDY) DATSEP('/')
01800A            TIME            T  B  8  2
01900A            TSTMP           Z  I  7 30
     A
```

Figure 7. Data type and keyboard shift coding

Three special characters are used in Table 2 on page 21.

- _ means that you did not type in any character.
- X indicates a blank.
- } is represented internally as hex D0.

Except where indicated, you enter the data only by pressing a command function key. The fourth entry under SIGN1 is an exception. You make that entry by pressing the Field Exit key.

The following table refers to fields defined in Figure 7 on page 20.

Table 2. Example data type and keyboard shift coding instructions

Field name (keyboard shift)	As typed in by the workstation user	As passed to your program
CHARA (Alphanumeric shift)	1. _ _ _ _ _ 2. A B C _ _ 3. A _ C _ _ 4. _ _ _ D E	• x x x x x (X'40') • A B C x x • A x C x x • x x x D E
CHARB (Alphanumeric shift)	1. _ _ _ _ _ 2. A B C _ _ 3. A _ C _ _ 4. _ _ _ D E	• x x x x x • x x A B C • x x A x C • x x x D E
CHARC (Alphanumeric shift)	1. _ _ _ _ _ 2. A B C _ _ 3. A _ C _ _ 4. _ _ _ D E	• 0 0 0 0 0 • 0 0 A B C • 0 0 A x C • 0 0 0 D E
CHARD (Alphabetic only)	1. _ _ _ _ _ 2. A B C _ _ 3. 4 _ _ _ _ 4. A B C. $ _	• x x x x x • A B C x x • error message • error message
CHARE (Numeric only character)	1. _ _ _ _ _ 2. 516.7 3. 5, 2 _ _ 4. A _ _ _ _	• x x x x x • 516.7 • 5, 2 x x • error message
CHARF (Digits only)	1. _ _ _ _ _ 2. 2 3 _ 5 _ 3. 1 2 _ _ _ 4. A _ _ _ _	• x x x x x • 2 3 x 5 x • 1 2 x x x • error message
SIGN1 (Signed numeric)	1. _ _ _ _ _ 2. 1 2 3 _ _ _ 3. 1 _ 3 _ _ _ 4. _ _ _ 4 5 _ (Field Exit key) 5. _ _ _ 4 5 (Field + key) 6. _ _ _ 4 5 _ (Field - key) 7. 1 2 3 4 5 _	• 0 0 0 0 0 • 0 0 1 2 3 • 0 0 1 0 3 • 0 0 0 4 5 • 0 0 0 4 5 • 0 0 0 4 N(X'F0F0F0F4D5') • 1 2 3 4 5
SIGN2 (Signed numeric)	1. _ _ _ _ _ 2. 1 2 3 4 _ _ 3. 1 2 _ _ _ _ 4. 1 2 _ _ _ _ (Field - key)	• 0 0 0 0 0 • 0 1 2 3 4 • 0 0 0 1 2 • 0 0 0 1 K (X'F0F0F0F1D2')

Table 2. Example data type and keyboard shift coding instructions (continued)

Field name (keyboard shift)	As typed in by the workstation user	As passed to your program
NBR1 (Numeric only)	1. _ _ _ _ _ 2. 0 0 0 0 5 3. 0 0 0 5 _ 4. 0 0 2 _ _	• 0 0 0 0 0 • 0 0 0 0 5 • 0 0 0 0 5 • 0 0 0 0 2
NBR2 (Numeric only)	1. _ _ _ _ _ _ 2. 0 0 0 0 5 _ 3. 0 0 5 _ _ _ 4. 5 _ _ _ _ _ 5. 0 5 0 0 _ _ 6. 5 x 5 _ _ 7. 5 x x _ _ _ 8. 5 5 . 1 _ _ 9. 5 0 0 0 0 . _ 10. 5 0. 0 0 0 11. 5 5 – _ _ _ 12. 5 _ – – _ _ 13. 5 _ + _ – _	• 0 0 0 0 0 • 0 0 5 0 0 • 0 0 5 0 0 • 0 0 5 0 0 • 5 0 0 0 0 • 5 0 5 0 0 • 0 0 5 0 0 • 0 5 5 1 0 • error message (use of decimal • error message not valid) • 0 5 5 0 } • 0 0 5 0 } • 0 0 5 0 }
NBR3 (Numeric only)	Input processing is the same as for NBR2.	
NBR4 (Numeric shift)	1. _ _ _ _ _ _ 2. 5 – _ _ _ _ 3. _ 5 _ – _ _ 4. 5 _ + _ _ _ 5. 5 _ A B C _ 6. 5 _ K K _ _ 7. 5 _ K A K _ 8. 5 _ K K A _ 9. 1 0 E + 0 3	• 0 0 0 0 0 • 0 0 5 0 } • 0 0 5 0 } • 0 0 5 0 0 • 0 0 5 0 0 • 5 0 2 0 } See note. • 5 0 2 0 } See note. • 0 0 5 0 0 See note. • 1 0 0 3
NBRZ (Hidden field)	This is a hidden field and does not appear on the display.	
FLPT (Floating point)	1. _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ 2. – 9 9 . 2 _ _ _ _ _ _ _ _ _ 3. – 9 9 E 0 2 _ _ _ _ _ _ _ _ 4. + 9 9 9 . 9 9 9 9 E + 0 0 3 5. A B C _ _ _ _ _ _ _ _ _ _ _ 6. _ _ 9 9 0 _ e _ _ _ _ _ _ _	• + 0 • – 9 9 . 2 • – 9 9 0 0 . • + 9 9 9 9 9 9 . 9 • error message • error message

Table 2. Example data type and keyboard shift coding instructions (continued)

Field name (keyboard shift)	As typed in by the workstation user	As passed to your program
DATE (Date)	1. _ _ _ _ _ _	• 4 0 / 0 0 1
	2. 0 0 0 0 1 _	• 0 0 / 0 0 1
	3. _ _ 1 _ _ _	• 0 0 / 0 0 1
	4. 0 0 / 0 0 1	• 0 0 / 0 0 1
	5. _ 0 / 0 0 1	• 0 0 / 0 0 1
	6. / 0 0 1 _ _	• error message
	7. 0 0 — 0 0 1	• error message
	8. A _ _ _ _ _	• error message
DATE1 (Date)	1. _ _ _ _ _ _ _ _	• 0 1 / 0 1 / 4 0
	2. 0 6 0 2 9 7 _ _	• 0 6 / 0 2 / 9 7
	3. 6 0 2 9 7 _ _ _	• 0 6 / 0 2 / 9 7
	4. 0 6 / 0 2 / 9 7	• 0 6 / 0 2 / 9 7
	5. _ 6 / 2 / 9 7 _	• 0 6 / 0 2 / 9 7
	6. 0 6 — 0 2 — 9 7	• error message
	7. 6 / 9 7 _ _ _ _	• error message
	8. 6 / / 9 7 _ _ _	• error message
	9. 1 3 / 2 / 9 7 _	• error message
	10. 6 / 3 1 / 9 7 _	• error message
	11. A / 2 / 9 7 _ _	• error message
TIME (Time)	1. _ _ _ _ _ _ _ _	• 0 0 . 0 0 . 0 0
	2. 1 2 3 4 5 6 _ _	• 1 2 . 3 4 . 5 6
	3. _ _ 1 2 3 4 5 6	• 1 2 . 3 4 . 5 6
	4. 1 2 _ _ _ _ _ _	• 1 2 . 0 0 . 0 0
	5. 1 2 3 4 _ _ _ _	• 1 2 . 3 4 . 0 0
	6. 1 2 . 3 4 . 5 6	• 1 2 . 3 4 . 5 6
	7. 1 . 2 . 3 _ _ _	• 0 1 . 0 2 . 0 3
	8. 0 1 : 0 0 : 0 0	• error message
	9. 1 2 3 _ _ _ _ _	• error message
	10. 1 2 3 4 5 _ _ _	• error message
	11. 1 . 0 0 0 0 _ _	• error message
TSTMP (Timestamp)	1. 2000–01–01–01.00.00.000000	• 2000–01–01–01.00.00.000000
	2. 20000101010000000000____ __	• 2000–01–01–01.00.00.000000
		• error message
	3. 2000/01/01/01.00.00.000000	• error message
	4. 0000–00–00–00.00.00.000000	

Note: The internal representation of K is hex D2. All nonnumeric characters (including those with hex D in the zone portion) are deleted with no place value. For example, 5_KAK_ becomes 5020}.

Decimal positions for display files (positions 36 and 37)

You use these positions to specify the decimal placement within a zoned decimal field and also to specify the data type of the field as it appears in your program.

If you leave these positions blank, the IBM i operating system assigns a data type of character for the field. If you type in a number in these positions, the IBM i operating system assigns a data type of zoned decimal for the field. The number you specify is the number of positions to the right of the decimal point. The entry must be less than or equal to the field length, with a maximum of 63 positions.

Packed decimal and binary fields are not supported for display files. Therefore, when you refer to fields of these types using the reference function, the data type assigned is zoned decimal with a keyboard shift as follows:

- If editing is in effect for the field you are defining, the keyboard shift is numeric only (Y in position 35).
- If no editing is in effect for the field you are defining, the keyboard shift is signed numeric (S in position 35).

If you are using a referenced field, you can override or change these positions. Specify the new value to override decimal positions. To change decimal positions, specify the amount you want the field increased or decreased and precede that number with either a plus (+) or minus (-) sign. For example, an entry of +4 indicates four more digits to the right of the decimal point than in the referenced field.

The example in "Date (L), Time (T), and Timestamp (Z)" on page 19 shows how to specify the decimal positions field.

Related concepts

Data type and keyboard shift for display files (position 35)
The entry you make in position 35 is the data type and keyboard shift attribute for display files.

Usage for display files (position 38)

You use this position to specify that a named field is an output-only, input-only, input/output (both), hidden, program-to-system, or message field.

Make no entry in this position for a constant (unnamed) field.

The valid entries for display files are:

Entry
Meaning

Blank or O
Output only

Output-only fields pass data from a program to the device when the program writes a record to a display. You can use the DFT (Default) keyword to specify an initial value for a named output field if you also specify the OVRDTA keyword for the field. If the OVRDTA keyword is not in effect, the initial value for the field is used. If the OVRDTA keyword is in effect, the data for the field is used and the data is taken from the output buffer.

I
Input only

Input-only fields pass data from the device to a program when the program reads a record. Input fields can be initialized with a default value (specified on the DFT keyword). If you do not change the field and the field is selected for input, the default value is passed to the program. Input fields are, by default, underlined on the display. You can use the Change Input Default (CHGINPDFT) keyword or the Display Attribute Underline (DSPATR(UL)) keyword to prevent underlining.

B
Input/output (both)

Input/output fields are passed from a program when the program writes a record to the display, and are passed to a program when the program reads a record from the display and the field is selected

for input. Input/output fields are typically used when the program displays data that you can change. An initial value can be specified for the field on the DFT keyword. When DFT is specified, the OVRDTA keyword is also required and indicates whether the data displayed in the field is taken from the output buffer (OVRDTA in effect) or from the DFT keyword (OVRDTA not in effect). Input/output fields are, by default, underlined on the display.

H

Hidden (special input/output field)

A hidden field is a named, numeric, alphanumeric, date. time, or timestamp field that does not appear on the display. Your program can send data to the field with an output operation, and it can retrieve data from the field with an input operation, but you cannot see or change the contents of the field.

The following rules apply to hidden fields:

- Hidden fields are always named.
- Locations are not valid for hidden fields.
- Specify length, data type, and decimal positions as you do for other named fields.
- You can specify more than one hidden field for a display file.

Since hidden fields are not displayed, they are not considered input-capable or output-capable fields, even though your program can send and receive data from them.

The DATFMT and DATSEP keywords can be used on a date hidden field, and the TIMFMT and TIMSEP keywords can be used on a time hidden field. However, no formatting is done by the system when the record containing the hidden date or time field is written or read.

Hidden fields are useful in applications involving subfiles. For example, a subfile record can contain record key information in a hidden field. You cannot see the hidden field, but the field is returned to the program with the subfile record so that the program can return the record to the database.

M

Message (special output field)

A message field is a named, output-only, character field.

The following rules apply to message fields:

- You can use option indicators to select message fields, but during processing, only one message can be displayed at a time. The message from the first message field selected is displayed, and all others are ignored for that operation.
- When a message field displays, all other fields you specify for that record are processed in the normal manner. The device goes into an error condition (locked keyboard, blinking cursor, and message displayed with the high intensity (HI) display attribute). When you press the Reset key, normal processing continues.
- The text of the message is established when your program moves a value to the message field.
- The location of the message on the display is the message line (the last line on the display unless the MSGLOC keyword is in effect).
- The length you specify for the message field should be less than 79 positions for 24 x 80 workstations, or less than 131 positions for the 27 x 132 workstation. Any message text that occupies more than 78 positions on the 24 x 80 workstations, or more than 130 positions on a 27 x 132 workstation is truncated to fit the message line.
- The Help key is not supported for message fields. Message help for the message is not displayed when Help is pressed.
- Only the following keywords are valid for a message field:

ALIAS	REFFLD
INDTXT	TEXT
OVRDTA	

- You cannot specify M in position 38 for a field if the field is part of the subfile record format.

Note: It is valid to send an input operation to a record that contains no input-capable fields. This permits pressing a function key as a response to an output record.

P

Program-to-system (special output field)

A program-to-system field is a named, numeric, or alphanumeric output-only field that is used to pass data between the program and the system. The field does not appear on the display. Your program can send data to the field with an output operation, but the workstation user cannot see the contents of the field. Because program-to-system fields are not displayed, they are not considered output-capable fields, even though the program can send data to them.

The following rules apply to program-to-system fields in display files:

- Program-to-system fields are always named.
- Locations are not valid for program-to-system fields.
- Specify length, data-type, and decimal positions as you do for other named fields.
- The program-to-system field must be specified as a parameter on a CHCACCEL, CHCCTL, CHKMSGID, CHOICE, ERRMSGID, GRDATR, GRDBOX, GRDCLR, GRDLIN, HTML, MNUBARCHC, MSGID, PSHBTNCHC, SFLCHCCTL, SFLMSGID, SFLSIZ, WDWTITLE, or WINDOW keyword within the same record format. The P-usage field is not valid as a parameter on any other keyword. A severe error is sent if the field is not specified on at least one of these keywords.
- Unlike the P-usage fields in ICF files, P-usage fields in display files can appear anywhere in the buffer. In ICF files, P-fields must be specified after all the data fields (B-usage fields).
- A P-usage field can be specified as the message-identifier, message-file, or library name on a MSGID keyword, provided the field is defined with the proper attributes, such as length.
- The record containing the P-usage field must be written before the data contained within the P-usage field is known to the system.

The only keywords allowed on a program-to-system field are:

ALIAS	TEXT
INDTXT	REFFLD

Notes:

1. Input-only and input/output fields are input-capable fields.
2. Output-only and input/output fields are output-capable fields.
3. Output-only is the default if you leave the position blank.

Location for display files (positions 39 through 44)

You use these positions to specify the exact location on the display where each field begins.

You cannot specify a location for hidden, program-to-system, or message fields. The validity of the location is based on the DSPSIZ keyword and the display size condition names.

Related concepts

Name for display files (positions 19 through 28)
You use these positions to specify record format names and field names.

Length for display files (positions 30 through 34)
You must specify a length for each named field unless you are copying the length from a referenced field.

Related reference

DSPSIZ (Display Size) keyword for display files

You use this file-level keyword to specify the display size to which your program can open the display file.

Line (positions 39 through 41)

You use these positions to specify the line on which the field begins.

The entry must be right-aligned. Leading zeros are optional. The maximum number of lines is 24 or 27. See the table in "Display size condition names" on page 4 for more information.

Position (positions 42 through 44)

You use these positions to specify the starting position of the field within the line you specified.

Your entry must be right-aligned. Leading zeros are optional. The maximum position is 132 for the 3180 device and 80 for all of the remaining display devices.

For fields other than the first field within the record, you can specify the location by specifying a plus value (+n) for positions 42 through 44. The plus value indicates the number of spaces to be left between the end of the previous field and the start of the field you are defining. The plus value must be in the range of 1 through 99. (A plus value of zero is not valid.)

Beginning attribute character

Each field displayed has one attribute character that defines the display attribute of the field on the display.

This attribute character is not displayed, but occupies one position on the display immediately preceding the field. Because of the beginning attribute character, you cannot specify that a field is to begin in the first position of the display (line 1, position 1), unless the SLNO (Starting Line Number) keyword is specified and the start line number is greater than 1.

When a field begins in position 1 the beginning attribute character occupies the last position of the preceding line. If such a field is the first field of a record, the preceding line is a part of the record area and displayed as a blank line. Any record format using that line cannot be displayed at the same time as the other record. The last one to be displayed causes the other one to be deleted (unless CLRL(*NO) is specified for the last displayed record).

Ending attribute character

The end of a field on the display is indicated by an ending attribute character, unless there is only one space between that field and the next field.

In that case, the beginning attribute character of the second field serves as the ending attribute character of the first field. In any case, there must always be at least one space left between fields in a record. When the last position occupied by a field of a record is the last position in a line, the ending attribute character for the field is in position 1 of the next line. However, the next line is not considered part of the first record, and records can be displayed on both lines at once.

Overlapping fields

Within a record format, you can define fields to overlap portions of other fields or their attribute characters; however, only one of those fields is shown on the display at a time.

At run time, when processing overlapping fields within a record, the IBM i program looks at the fields in line and position sequence. When the IBM i operating system finds a field whose conditioning is satisfied or that does not have an option indicator specified, it selects that field for display and ignores the remaining overlapping fields. The first overlapping field that does not have an option indicator specified always stops the search, and any subsequent overlapping fields are never displayed. In the following example, if indicator 01 is set on, FIELD1 is the only field displayed. If indicator 01 is off and indicator 02 is on, FIELD2 is the only field displayed. FIELD3 is displayed when neither of the others is selected.

Figure 8 on page 28 shows how to define overlapping fields.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A  01        FIELD1        10      1  5TEXT('ONE')
00020A  02        FIELD2         5      1  5TEXT('TWO')
00030A            FIELD3         2  0   1  5TEXT('THREE')
     A
```

Figure 8. Specifying overlapping fields

If used incorrectly, this capability can result in problems in user and program communication. In the following example, only one input field (FIELD4) is specified for the record, and according to the field location specification, this field overlaps a preceding output field. The workstation user is not able to type in any data because FIELD1 is always the field selected for display. The other three fields, including FIELD4, are never displayed.

Figure 9 on page 28 shows an example of incorrect field specification entry.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A            FIELD1        10      1  5TEXT('ONE')
00020A  21        FIELD2         5      1  5TEXT('TWO')
00030A  12        FIELD3         2  0   1  5TEXT('THREE')
00040A            FIELD4         5Y 2I  1  5TEXT('FOUR')
     A
```

Figure 9. Incorrect field specification

Display length

The display length is increased for certain types of fields and must be considered when laying out the display.

The display length is increased by the IBM i operating system in the following situations:

- For numeric-only fields with editing, the display length is determined from the edit word or the program length and the edit code.
- For input-capable signed numeric fields, the display length is one more than the program length.
- The display length is one more than the program length for numeric shift fields and for numeric-only fields without editing when these fields are input-capable and have decimal positions greater than zero.
- The display length for floating-point fields is 7 more than the length specified in positions 30 through 34. The 7 extra positions are for the significand sign, the decimal point or comma, the exponent character, the exponent sign, and the 3 exponent digits.

For an unsigned numeric field (like FIELD4 in Figure 1) with a nonzero decimal position, the system requires a decimal character to be typed into the field when decimal values are typed in as data. In Figure 2, 123 in FIELD4 does not require a decimal character, but 1234 does (123.4). For this field, the display length is 6.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A            FIELD1        10      1  5TEXT('ONE')
00020A  21        FIELD2         5      1  5TEXT('TWO')
00030A  12        FIELD3         2  0   1  5TEXT('THREE')
00040A            FIELD4         5Y 2I  1  5TEXT('FOUR')
     A
```

Figure 10. Incorrect field specification

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A  01        FIELD1        10      1  5TEXT('ONE')
00020A  02        FIELD2         5      1  5TEXT('TWO')
00030A            FIELD3         2  0   1  5TEXT('THREE')
     A
```

Figure 11. Specifying overlapping fields

DDS keyword entries for display files (positions 45 through 80)

You type the keyword entries that define display files in positions 45 through 80 (functions).

Related concepts

Positional entries for display files (positions 1 through 44)
You specify positional entries in the first 44 positions of the data description specifications (DDS) form for display files.

Related tasks

Defining a display file for DDS
When you specify positional entries for display files, you need to follow some specific rules for filling in positions 1 through 44 of the data description specifications (DDS) form.

Related reference

Keyword considerations for display files that use DBCS
Some DDS keywords should be avoided in double-byte character set (DBCS) data fields, and others should be used with caution.

Related information

Rules for DDS keywords and parameter values

ALARM (Audible Alarm) keyword for display files

You use this record-level keyword to specify that the IBM i operating system is to activate the audible alarm when this record is displayed. The alarm is of short duration.

This keyword has no parameters.

If the Error Message (ERRMSG) or Error Message Identifier (ERRMSGID) keyword is in effect, the ALARM keyword has no effect, even if also selected.

To sound the audible alarm when an active ERRMSG or ERRMSGID keyword is on the record, see "MSGALARM (Message Alarm) keyword for display files" on page 159.

Option indicators are valid for this keyword.

Example

The following example shows how to specify the ALARM keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00005A          R CUST
00010A  01                                ALARM
     A
```

ALIAS (Alternative Name) keyword for display files

You use this field-level keyword to specify an alternative name for a field.

When the program is compiled, the alternative name is brought into the program instead of the DDS field name. The high-level language compiler in use determines if the ALIAS name is used. Read the appropriate high-level language reference manual for information about ALIAS support for that language.

The format of the keyword is:

```
ALIAS(alternative-name)
```

The alternative-name must be different from all other alternative names and from all DDS field names in the record format. If a duplicate name is found, an error appears on the field name or alternative name.

An alternative name cannot be used within DDS or any other IBM i function (for example, as a key field name, as the field name specified for the REFFLD keyword, or as a field name used in the Copy File (CPYF) command).

When you refer to a field that has the ALIAS keyword, the ALIAS keyword is copied in unless the ALIAS keyword is explicitly specified on the referencing field.

Option indicators are not valid for this keyword.

Example

The following example shows how to specify the ALIAS keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00070A            FIELDA        25A     1  2ALIAS(CUSTOMERNAME)
     A
```

The alternative name for FIELDA is CUSTOMERNAME.

Related information

DDS naming conventions

ALTHELP (Alternative Help Key) keyword for display files

You use this file-level keyword to assign a command attention (CA) key as an alternative Help key.

When either the Help key or the CA key is pressed, the help function is called.

The format of the keyword is:

```
ALTHELP[(CAnn)]
```

The valid values for the optional parameter are CA01 through CA24. If the parameter is not specified, CA01 is the default.

The HELP keyword must also be specified, either at the file level or on at least one record in the file. ALTHELP applies only to records for which the HELP keyword also applies. If HELP is specified at the file level, it applies to all records in the file; thus, ALTHELP also applies to all records in the file. If HELP is specified at the record level, ALTHELP applies only to those records that have the HELP keyword specified.

If you specify a response indicator on the HELP keyword, the response indicator is set on and control is returned to your program when either the Help key or the CAnn key is pressed.

If you specify option indicators on the HELP keyword, the same option indicators apply to the ALTHELP keyword. That is, both the Help key and the CAnn key are active when the indicator condition is true.

The following keywords cannot be specified in a file with an ALTHELP keyword that has no parameter (CA01 default):

ALTPAGEDWN(CF01)
ALTPAGEUP(CF01)
CA01
CF01
MNUCNL(CA01)
MNUBARSW(CA01)
MOUBTN(...CF01)
PSHBTNCHC(...CF01)
SFLDROP(CA01)
SFLDROP(CF01)
SFLENTER(CA01)
SFLENTER(CF01)
SFLFOLD(CA01)
SFLFOLD(CF01)

Similarly, the following keywords cannot be specified in a file with ALTHELP(CA*nn*) (where *nn* is the same number):

ALTPAGEDWN(CFnn)	PSHBTNCHC(...CFnn)
ALTPAGEUP(CFnn)	SFLDROP(CAnn)
CAnn	SFLDROP(CFnn)
CFnn	SFLENTER(CAnn)
MNUCNL(CAnn)	SFLENTER(CFnn)
MNUBARSW(CAnn)	SFLFOLD(CAnn)
MOUBTN(...CFnn)	SFLFOLD(CFnn)

You cannot specify RETKEY or RETCMDKEY in a file with the ALTHELP keyword.

Option indicators are not valid for this keyword.

Example

The following example shows how to specify the ALTHELP keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A                                     ALTHELP
     A                                     HELP(01 'HELP KEY PRESSED')
     A          R RECORD
     A            FIELD1        20A     5  5
     A
```

The CA01 key is an alternative Help key. If the user presses the CA01 key or the Help key, response indicator 01 is set on and control returns to the application program.

Related reference

HELP (Help) keyword for display files
You use this file-level or record-level keyword to enable the Help key.

ALTNAME (Alternative Record Name) keyword for display files

You use this record-level keyword to specify an alternative name for a record.

The alternative name can be specified for I/O operations when using program-described files. The syntax of the alternative record name must be valid for the high-level language compiler in use.

The format of the keyword is:

```
ALTNAME('alternative-name')
```

See “System/36 environment considerations for display files” on page 247 for information about how to specify the ALTNAME keyword.

ALTPAGEDWN/ALTPAGEUP (Alternative Page Down/Alternative Page Up) keyword for display files

You use these file-level keywords to assign command function (CF) keys as alternative Page Down/Page Up keys.

When either the Page keys or the CF keys are pressed, the page function is called. The alternative Page Down and Page Up keys function only on the displays defined in the file with the ALTPAGEDWN or ALTPAGEUP keyword. They do not function on system displays, such as message help.

The format for each of these keywords is:

```
ALTPAGEDWN[(CFnn)]
ALTPAGEUP[(CFnn)]
```

The valid values for the optional parameters are CF01 through CF24. If the parameter is not specified, CF08 is the default for ALTPAGEDWN and CF07 is the default for ALTPAGEUP.

The PAGEDOWN/PAGEUP keyword should also be specified if you want your program to handle any situation where the user has pressed a Page key or the alternative CFnn key and the IBM i program cannot move the text lines on the display. If you do not specify the PAGEDOWN/PAGEUP keyword, a message indicating the key is not valid is displayed when either a Page key or the alternative CFnn key is pressed and the IBM i operating system cannot move the text lines on the display.

Note: Throughout this keyword description, PAGEDOWN means the PAGEDOWN or the ROLLUP keyword. PAGEUP means the PAGEUP or the ROLLDOWN keyword.

If you specify a response indicator on the PAGEDOWN or PAGEUP keyword, the response indicator is set on and returned to your program when either the Page key or the alternative CFnn key is pressed and the IBM i operating system cannot move the text lines on the display.

If you specify option indicators on the PAGEDOWN or PAGEUP keyword, the same option indicators apply to the ALTPAGEDWN or ALTPAGEUP keyword, in the order given. That is, when the indicator condition is true, control will be returned to your program when either the Page key or the CFnn key is pressed and the IBM i operating system cannot move the text lines on the display.

The following keywords cannot be specified in a file with an ALTPAGEDWN keyword that has no parameter (CF08 default):

ALTHELP(CA08)
ALTPAGEUP(CF08)
CA08
CF08
MNUCNL(CA08)
MNUBARSW(CA08)
MOUBTN(...CA08)
PSHBTNCHC(...CA08)
SFLDROP(CA08)
SFLDROP(CF08)
SFLENTER(CA08)
SFLENTER(CF08)
SFLFOLD (CA08)
SFLFOLD (CF08)

The following keywords cannot be specified in a file with an ALTPAGEUP keyword that has no parameter (CF07 default):

ALTHELP(CA07)
ALTPAGEDWN(CF07)
CA07
CF07
MNUCNL(CA07)
MNUBARSW(CA07)
MOUBTN(...CA07)
PSHBTNCHC(...CA07)
SFLDROP(CA07)
SFLDROP(CF07)
SFLENTER(CA07)
SFLENTER(CF07)
SFLFOLD (CA07)
SFLFOLD (CF07)

Similarly, the following keywords cannot be specified in a file with ALTPAGEDWN(CF*nn*) or ALTPAGEUP(CF*nn*) (where *nn* is the same number):

ALTHELP(CA*nn*)
CA*nn*
CF*nn*
MNUCNL(CA*nn*)
MNUBARSW(CA*nn*)
MOUBTN(...CA*nn*)
PSHBTNCHC(...CA*nn*)
SFLDROP(CA*nn*)
SFLDROP(CF*nn*)
SFLENTER(CA*nn*)
SFLENTER(CF*nn*)
SFLFOLD (CA*nn*)
SFLFOLD (CF*nn*)

Also, you cannot specify the same command function (CF) key on both the ALTPAGEDWN and the ALTPAGEUP keywords.

You cannot specify RETKEY or RETCMDKEY in a file with the ALTPAGEDWN or ALTPAGEUP keyword.

The ALTPAGEDWN and ALTPAGEUP keywords are allowed only in files containing a pageable area (either a subfile or a PAGEDOWN/PAGEUP keyword).

Option indicators are not valid for these keywords.

Example

The following example shows how to specify the ALTPAGEDWN and ALTPAGEUP keywords.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A                                      ALTPAGEUP
     A                                      ALTPAGEDWN
     A          R SUBFILE                   SFL
     A            FIELD1        20A    5   5
     A          R CONTROL                    SFLCTL(SUBFILE)
     A                                       SFLSIZ(30)
     A                                       SFLPAG(10)
     A                                       SFLDSP
     A
```

The CF07 and CF08 keys are alternative page keys. During display of the subfile, if the user presses the CF07/CF08 key or a Page key, the IBM i operating system pages the subfile. If the IBM i operating system cannot page the subfile without going beyond the end or beyond the beginning, a message is displayed indicating the key is not valid at that time (PAGEDOWN/PAGEUP not specified).

Related reference

PAGEDOWN/PAGEUP (Page Down/Page Up) keywords for display files
You use these file-level or record-level keywords to specify that your program handles any situation where the workstation user has pressed the Page Down or Page Up keys and the IBM i operating system cannot page through the display.

ALWGPH (Allow Graphics) keyword for display files

This file- or record-level keyword allows graphics and alphanumeric contents to be displayed by the record format on a 5292 Model 2 Color Display Station at the same time.

The keyword is ignored if it is specified for a file displayed on any other type of display.

This keyword has no parameters.

When a record with this keyword in effect is written to a 5292 Model 2 Color Display Station, the device is placed in graphics display mode, if it was not already in that mode. This is done whether the keyword is in effect for any other record also displayed on the device. The device remains in graphics display mode as long as there is any record displayed with the ALWGPH keyword selected. Displaying a record that does not have the ALWGPH keyword will not cause graphics display mode to end.

To turn the graphics display off, you must delete all records with the ALWGPH keyword in them from the display (or option off the keyword in those records). When the graphics display is turned off, any graphics already on the display *are not* deleted, but are no longer displayed. Provided that the graphics display is not deleted (through the use of Graphical Data Display Manager (GDDM) functions), it will be displayed again the next time a record with ALWGPH is displayed, including one from a secondary interactive job called from the System Request Menu.

In the graphics display mode:

- The device is automatically placed in reduced line spacing mode, with less space between lines. This is done regardless of the workstation user's choice of this mode from the keyboard (in local mode), and it cannot be overridden by the user.
- Any graphics that are already on the display when the DDS record format is displayed remain on the display as a background to the alphanumeric characters displayed by the record format.

Option indicators are valid for this keyword.

This keyword cannot be specified with the SFL or the USRDFN keywords.

Example 1

The following example shows how to specify the ALWGPH keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A          R RECORD1                  ALWGPH
00020A                                   23  2'Enter account number:'
00030A            ACCT           5   B    +2
     A
```

In the example, RECORD1 can be displayed with graphics. The constant *Enter account number:* and the field ACCT appear on the display. Any graphics also displayed (through the use of GDDM routines) appear *behind* the alphanumerics. In other words, when a graphics line or pattern crosses the alphanumerics, a portion of the graphics is covered by the alphanumerics.

Example 2

The following example shows how RECORD2 can be displayed with graphics only if option indicator 01 is on.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A          R RECORD2
00020A  01                                 ALWGPH
00030A                                   1 34'Sample Title'
     A
```

In the example, if option indicator 01 is off, any graphics already displayed are turned off (not deleted) and only the alphanumerics specified through DDS (such as *Sample Title*) appear on the display. On a later output operation, with option indicator 01 on, the graphics reappear with the alphanumerics.

ALWROL (Allow Roll) keyword for display files

This record-level keyword enables your program to page through data in a window on the display when the system is displaying the record format you are defining.

The window consists of display lines between and including a start line and end line defined in your program. The number of lines to be paged through and the direction in which to page through them are defined in your program.

This keyword has no parameters.

When your program sends an output operation to this record format, the IBM i operating system pages through data already in the window up or down the display and then displays the record format. Data paged past the start line or end line is lost. After you page through, your program cannot send an input operation to record formats that were either partially or completely within the window before the page.

To use the ALWROL function in COBOL, use the WRITE ROLLING statement.

The ALWROL keyword does not allow the display station user to page through data; it only allows your program to page through data on the display. To allow paging of data by the display station user, specify the ROLLUP and ROLLDOWN keywords or specify a subfile with subfile page not equal to subfile size.

To prevent deleting paging records, specify the OVERLAY keyword or the CLRL (Clear Line) keyword with the ALWROL keyword.

Do not specify the PUTRETAIN keyword at the field level when you also specify the ALWROL keyword. If you do so, the IBM i operating system sends message CPF5014 when your program sends an output operation regardless of your selection of PUTRETAIN.

If you specify the ALWROL keyword with the following keywords, you must specify option indicators for them:

ERRMSG
ERRMSGID
PUTOVR
PUTRETAIN (at the record-level)

Your program cannot at the same time select one of these keywords and send an output operation that attempts to use the ALWROL function (the IBM i operating system sends CPF5014).

The ALWROL keyword cannot be specified with any of the following keywords:

ASSUME
KEEP
SFL
SFLCTL
USRDFN

A warning message appears at file creation time if the ALWROL keyword is specified on a record with the DSPMOD keyword. At run time, the ALWROL keyword is ignored when the display mode changes.

The ALWROL keyword cannot be specified for the record format specified by the PASSRCD keyword.

Option indicators are not valid for this keyword.

Example

The following example shows how to specify the ALWROL keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A    1  R RECORD1                   ALWROL OVERLAY  2
00020A             FLDA          79   I 23  2CHECK(LC)
00030A  44                            3  ERRMSG('Record not found' 44)
     A
```

1
The application program can send an output operation to RECORD1, displaying FLDA on line 23, position 2. In a subsequent output operation, the program can page through RECORD1 (in this case, FLDA) up or down the display or entirely off the display. A normal case is to page up one line. The originally typed data is then displayed on line 22, and a new input field is displayed on line 23. The display station user cannot type into the field on line 22, and the program cannot read this field. The field on line 22 can be pushed up the display by subsequent output operations in this way until it passes the start line of the window (as specified in the program) or line 1 of the display.

2
The OVERLAY keyword prevents paged records from being deleted.

3
The ERRMSG keyword is shown to illustrate how ERRMSG affects the ALWROL function. The program cannot at the same time set option indicator 44 on and send an output operation that requests the ALWROL function. If the program does so, the IBM i operating system sends a notify message CPF5014.

Related reference

SFLROLVAL (Subfile Roll Value) keyword for display files
You use this field-level keyword in the subfile-control record format to specify that the workstation user can type a value in this field. The value tells the IBM i operating system how many records to page up or down when the appropriate paging key is pressed.

SFLRCDNBR (Subfile Record Number) keyword for display files
You use this field-level keyword on the subfile-control record format to specify that the page of the subfile to be displayed is the page that contains the record whose relative record number is in this field.

ASSUME (Assume) keyword for display files

This record-level keyword specifies that the IBM i operating system is to assume that the record is already shown on the display when the display file is opened.

Such a record can also be defined, with the KEEP keyword, in another display file. That other display file can be closed before this file (in which you are specifying ASSUME) is opened.

This keyword has no parameters.

Specify the ASSUME keyword for at least one record format within the display file so that the IBM i operating system does not erase the display when the file is opened. In addition, specify the OVERLAY keyword with the ASSUME keyword to prevent the IBM i operating system from deleting the display when your program sends the first output operation after opening the file.

If you use the ASSUME keyword, at least one field in the record must be able to be displayed. If more than one record with the ASSUME keyword exists, they must occupy unique display lines.

For the IBM i operating system to process the data correctly, your program must specify the record format name containing this keyword.

The ASSUME keyword is not needed if the record format you are defining is in a shared file (SHARE(*YES)) parameter specified on the Create Display File (CRTDSPF), Change Display File (CHGDSPF), or Override with Display File (OVRDSPF) command).

This keyword cannot be specified with any of the following keywords:

ALWROL	SLNO
CLRL	USRDFN
SFL	USRDSPMGT

A warning message is issued at file creation time if the ASSUME keyword is specified on a record with the DSPMOD keyword. At run time, the ASSUME keyword is ignored when the display mode changes.

A file with the ASSUME keyword will be opened to the display size of the file with the KEEP keyword.

Option indicators are not valid for this keyword.

Example

The following example shows how to specify the ASSUME keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00005A          R RECORD                  ASSUME
     A
```

AUTO (Auto) keyword for display files

Under some circumstances, the AUTO keyword is equivalent to the CHECK keyword.

The AUTO keyword is equivalent to the CHECK keyword as follows:

AUTO(RA)
CHECK(ER)

AUTO(RAB)
CHECK(RB)

AUTO(RAZ)
CHECK(RZ)

The format of the keyword is:

```
AUTO(RA [RAB | RAZ])  AUTO(RAB | RAZ)
```

The CHECK keyword is preferred.

Related reference
CHECK (Check) keyword for display files

You use this keyword to perform a number of functions, depending on the parameter values specified.

BLANKS (Blanks) keyword for display files

This field-level keyword, when specified for a numeric, input-capable field, enables your program to distinguish when the field is blank and when the field is zero on the display. In either case, your program recognizes zeros.

The BLANKS keyword sets on the specified response indicator when the field is blank on the display. After an input operation, your program can test this indicator to determine that the field (whose program value is zero) is actually blank on the display. The field can contain all blanks (hex 40) or all nulls (hex 00). It still appears blank to the display station user. If the indicator is off, the field is zero on the display.

This keyword is also valid for character fields, but there is generally no need to specify it for them. Your program can test character fields directly to determine what is on the display.

See "System/36 environment considerations for display files" on page 247 for special considerations when you specify the BLANKS keyword in files that are used in the System/36 environment.

The format of the keyword is:

```
BLANKS(response-indicator ['text'])
```

The response indicator associated with the BLANKS keyword should be unique within the record. That is, the same response indicator should not be used with other keywords, such as CHANGE, DUP, or VLDCMDKEY; with any of the keywords for function keys; or with the BLANKS keyword on other fields in the same record. This is because the IBM i operating system always turns the response indicator off if the field contains non-blank characters on an input operation. The IBM i operating system does this to make sure that when the field appears as all blanks, the response indicator is set on, and that when it does not appear as all blanks, the response indicator is set off.

The optional text is included on the list generated at program compilation time to explain the intended use of the indicator. This text has no function in the file or the program other than as a comment. The single quotation marks are required. If you specify more than 50 characters between the single quotation marks, the text is truncated to 50 characters on the program list.

Option indicators are not valid for this keyword.

Example 1: Specifying the BLANKS keyword

The following example shows how to specify the BLANKS keyword.

Note: Examples 2, 3, and 4 in this topic, show some cases that restrict the BLANKS keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00030A            QTY1           5Y 0B  5  2BLANKS(01 'ON=QTY1 IS ALL BLANKS')
00040A            QTY2           5Y 0B  6  2BLANKS(02 'ON=QTY2 IS ALL BLANKS')
00050A            QTY3           5Y 0B  7  2BLANKS(03 'ON=QTY3 IS ALL BLANKS')
     A
```

Three numeric fields (QTY1, QTY2, and QTY3) are displayed. If the display station user types values into the fields and presses the Enter key, the following situation occurs:

Value as typed into fields	Value as passed to program	Condition of response indicator
100	00100	Off
0	00000	Off
Blanks	00000	On

Note: If the display station user presses a Field Exit key or the Erase Input key, the field appears blank because it contains nulls.

Restricting the BLANKS keyword

In some cases, the BLANKS keyword does not set the specified response indicator on, but rather restricts its function. The following three examples illustrate these cases.

Note: Other cases occur when the field is a character field, but then it is unnecessary to use the BLANKS keyword.

Example 2

In the following example, when an input/output field contains all blanks (hex 40) or all nulls (hex 00) when displayed, and certain keywords affecting the display of the field are also specified, the response indicator is not set on.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A* When OVRATR is specified
00020A          R REC1                          PUTOVR
00030A            FLD1          10  0B  2  2BLANKS(50) OVRATR
00040A  78                                    DSPATR(HI)
00050A* When PUTRETAIN is specified
00060A          R REC2                          PUTRETAIN OVERLAY
00070A            FLD2          10  0B  2  2BLANKS(50)
00080A          R REC3                          OVERLAY
00090A            FLD3          10  0B  2  2BLANKS(50)
     A                                        PUTRETAIN
     A
```

For all record formats in this example, response indicator 50 is set on as expected the first time the field is read by the program (if the field appears blank on the display). However, after a subsequent display, response indicator 50 is set on again *only if the display station user again blanks the field*. If the workstation user does not again blank the field, response indicator 50 is off.

Example 3

Examples 2 and 4 concern cases when the field is first displayed, then deleted.

In the following example, when an input-capable field is overlapped by another field, causing the first field to be deleted, the response indicator is not set on (even though the field in the input buffer still contains all blanks or all nulls).

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A          R REC4                          OVERLAY
00020A  15        FLDA          10  0B  2  2
00030A            FLD4          10  0B  2  5BLANKS(50)
     A
```

In this example, if option indicator 15 is off when REC4 is first displayed, FLD4 is displayed and FLDA is not. When REC4 is read, response indicator 50 is set on if FLD4 is blank. If option indicator 15 is then set on when REC4 is displayed again, FLDA overlaps FLD4 and deletes it. Response indicator 50 is then off when REC4 is read. (This occurs because the IBM i operating system turns it off when displaying the record format and does not turn it back on for a field that is not on the display, even if the field contains blanks or nulls from a previous I/O operation.)

Example 4

In the following example, after initial display, an input/output field is not displayed again on a subsequent input/output operation (even though the field in the input buffer still contains all blanks or all nulls).

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A          R REC5                          ERASEINP OVERLAY
00020A  20        FLD5          10  0B  2  2BLANKS(50)
00030A*
00040A          R REC6                          ERASEINP  OVERLAY  MDTOFF
00050A  20        FLD6          10  0B  2  2BLANKS(50)  DSPATR(MDT)
     A
```

In this example, if option indicator 20 is on when REC5 or REC6 is first displayed, FLD5 or FLD6 is displayed. When REC5 or REC6 is read, response indicator 50 is set on if FLD5 or FLD6 is blank. However, if option indicator 20 is set off on a second display, FLD5 or FLD6 is not displayed.

BLINK (Blink) keyword for display files

You use this record-level keyword to specify that as long as the record being defined is displayed, the cursor flashes.

The blinking is set off by the next output operation of a record that does not have the BLINK keyword specified.

This keyword has no parameters.

Option indicators are valid for this keyword.

Example

The following example shows how to specify the BLINK keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00020A          R MASTER                  BLINK
     A
```

BLKFOLD (Blank Fold) keyword for display files

You use this field-level keyword for named, output-only fields (but not message or program-to-system fields) so that they overflow onto subsequent display lines.

The keyword causes folding to occur at a blank in the data rather than at the end of the display line. It is used to make long text fields easier to read. The default is for the data to be folded at the end of the physical line.

This keyword has no parameters.

When BLKFOLD is used, the field length is not increased. Therefore, it is possible for a portion of the output data to be truncated.

You cannot specify the BLKFOLD keyword on a floating-point field (F in position 35).

Option indicators are not valid for this keyword.

Example

The following example shows how to specify the BLKFOLD keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00030A            FIELD1       638      2  1BLKFOLD
     A
```

CAnn (Command Attention) keyword for display files

You use this file-level or record-level keyword to specify that the function key specified in the keyword (CA01 through CA24) is available for use.

It is to be used as a command attention (CA) key. No input data is transmitted from the device. Response indicators 01 through 99 are valid.

See “System/36 environment considerations for display files” on page 247 for special considerations when you specify the CA*nn* keyword in files that are used in the System/36 environment.

The format of the keyword is:

```
CAnn[(response-indicator ['text'])]
```

If you specify this keyword and the display station user presses the specified function key, the following situation happens:

- All other function key response indicators in the input buffer are set off (hex F0).
- The response indicator, if specified with the CAnn keyword, is set on (hex F1).
- The IBM i data management feedback area is updated.
- Data already in the input buffer remains unchanged except that the response indicator (if specified) is set on.
- Control is returned to your program.

If you specify a response indicator and the key is pressed, the response indicator is set on and returned to your program. (The text information is associated with the indicator and is used by high-level language compilers to help in program documentation.)

If the display station user presses a function key and you have not specified it as either a command function (CF) key or a command attention key, the IBM i operating system displays a message to the display station user indicating that the key is not valid at that time.

You can use combinations of CA and CF keywords within the same display file, but you cannot specify the same key number as both CA and CF keys. For example, CA02 and CF02 are not valid in the same display file.

Note: File level CA and CF keys are extended to the record level. This must be considered when assigning key numbers. For example, if CA02 is specified at file level and CF02 is specified at record level, CF02 is an error.

If you specify a key in the range 1 through 9, you must supply the leading zero in the keyword (for example, CA04).

Option indicators are valid for this keyword.

Validity checking

When the display station user presses a CF key, the data from fields with their MDT set on is placed into the input buffer before validity checking is done. Any errors in the data are then detected, and the appropriate error messages are sent to the display. Because validity checking is not done until after the data is placed in the input buffer, pressing a valid CA key after the CF key can cause incorrect data to be returned to your program. This condition is not a problem as long as your program does not process the input data when the CA key is pressed.

This condition can be prevented in either of two ways:

- Do not allow the use of CA keys. Specify CF keys, which cause validity checking to be done on the data.
- Do not specify any of the following validity checking keywords if CA keys are allowed:

  CHECK(M10)
  CHECK(M11)
  CHECK(VN)
  CHECK(VNE)
  COMP (EQ, NE, LT, NL, GT, NG, LE, GE)
  RANGE
  VALUES

Function keys valid for command attention keys at processing time

As a general rule, the last output operation determines which function keys are valid. However, the following list shows exceptions to this rule:

- When an operation sends no data to the display, the validity of various function keys is not changed. Such operations include:
  - An output operation to a subfile record

  - An update to a subfile record
  - An output operation to a subfile control record that only clears, deletes, or initializes a subfile without displaying the subfile or the subfile control record
- An output operation that displays an error message by selecting ERRMSG (Error Message) or ERRMSGID (Error Message ID) can also select a CA or CF key to be valid while the error message is displayed.
- If MNUCNL (Menu Cancel), MNUBARSW (Menu Bar Switch), or SFLDROP (Subfile Drop) is specified for a subfile, the validity of the CA or CF key specified for the SFLDROP keyword is determined by the last output operation. However, as long as the subfile is displayed, the CA or CF key, when valid, acts only as a Drop key.
- If SFLFOLD (Subfile Fold) is specified for a subfile, the validity of the CA or CF key specified for the SFLFOLD keyword is determined by the last output operation. However, as long as the subfile is displayed, the CA or CF key, when valid, acts only as a Fold key.
- If two subfiles using SFLDROP or SFLFOLD are displayed at one time, the same function key should be specified on both the SFLDROP and SFLFOLD keywords. If they are different, only the key specified for the most recently displayed subfile is in effect. Pressing the function key affects the subfile containing the cursor. If the cursor is not positioned in a subfile, the function key affects the upper subfile.
- If two subfiles using SFLENTER (Subfile Enter) are displayed at the same time, the only CA or CF key in effect as an Enter key is the CA or CF key specified for the SFLENTER keyword on the most recently displayed subfile. The cursor position at the time the Enter key is pressed determines which subfile is affected.

Note: The following keywords function like CA keys: CLEAR, HELP, HOME, and PRINT (with response indicator specified).

Example

The following example shows how to specify the CAnn keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00011A                                  CA01(91 'End of Program')
00012A                                  CA02(92)
00013A                                  CA03
     A
```

Related concepts

RETKEY (Retain Function Keys) and RETCMDKEY (Retain Command Keys) keywords
You use these record-level keywords to indicate that function keys, command function (CF*nn*) keys, or command attention (CA*nn*) keys, which are enabled on a display, should be retained when the record you are defining is displayed.

CFnn (Command Function) keyword for display files

You use this file-level or record-level keyword to specify that the function key specified in the keyword (CF01 through CF24) is available for use.

It is to be used as a command function (CF) key to transmit changed data as opposed to a command attention (CA) key, which does not transmit changed data. Response indicators 01 through 99 are valid.

See “System/36 environment considerations for display files” on page 247 for special considerations when you specify the CF*nn* keyword in files that are used in the System/36 environment.

The format of the keyword is:

```
CFnn[(response-indicator ['text'])]
```

If you specify this keyword, and the display station user presses the specified function key, the following situation happens:

- All other function key response indicators in the input buffer are set off (hex F0).
- The response indicator, if specified with the CFnn keyword, is set on (hex F1).

- The IBM i data management feedback area is updated.
- Data is placed in the input buffer according to data received from the device.
- Control is returned to your program.

If you specify a response indicator and the key is pressed, the response indicator is set on and returned to your program along with the input data. If no response indicator is specified, the input data is returned to your program. (The text information is associated with the indicator and is used by high-level language compilers to help in program documentation.)

If the display station user presses a function key and you have not specified it as either a CF key or a CA key, the IBM i operating system displays a message to the display station user indicating that the key is not valid at that time.

You can use combinations of CF and CA keywords within the same display file, but you cannot specify the same key number as both command attention and command function. For example, CA01 and CF01 are not valid in the same display file.

Note: File level CA and CF keys are extended to the record level. This must be considered when assigning key numbers. For example, if CA02 is specified at file level and CF02 is specified at record level, CF02 is an error.

If you specify a key in the range 1 through 9, you must supply the leading zero in the keyword (for example, CF03).

Option indicators are valid for this keyword.

Function keys valid for command function keys at processing time

As a general rule, the last output operation determines which function keys are valid. The following list shows exceptions to this rule:

- When an operation sends no data to the display, the validity of the various function keys is not changed. Such operations include:
  - An output operation to a subfile record
  - An update to a subfile record
  - An output operation to a subfile control record that only clears, deletes, or initializes a subfile without displaying the subfile or the subfile control record
- An output operation that displays an error message by selecting ERRMSG or ERRMSGID can also select a CA or CF key to be valid while the error message is displayed.
- If SFLDROP is specified for a subfile, the validity of the CA or the CF key specified for the SFLDROP keyword is determined by the last output operation. However, as long as the subfile is displayed, the CA or CF key, when valid, acts only as a Drop key.
- If SFLFOLD is specified for a subfile, the validity of the CA or the CF key specified for the SFLFOLD keyword is determined by the last output operation. However, as long as the subfile is displayed, the CA or CF key, when valid, acts only as a Fold key.
- If two subfiles using SFLDROP or SFLFOLD are displayed at one time, the same function key should be specified on both the SFLDROP and SFLFOLD keywords. If they are different, only the key specified for the most recently displayed subfile is in effect. Pressing the function key affects the subfile containing the cursor. If the cursor is not positioned in a subfile, the function key affects the upper subfile.
- If two subfiles using SFLENTER are displayed at the same time, the only CA or CF key in effect as an Enter key is the CA or CF key specified for the SFLENTER keyword on the most recently displayed subfile. The cursor position at the time the Enter key is pressed determines which subfile is affected.

Note: The ROLLUP and ROLLDOWN keywords function like CF keys.

Example

The following example shows how to specify the CFnn keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A                                  CF01(91 'End of Program')
00020A                                  CF02(92)
00030A                                  CF03
     A
```

Related concepts

RETKEY (Retain Function Keys) and RETCMDKEY (Retain Command Keys) keywords
You use these record-level keywords to indicate that function keys, command function (CF*nn*) keys, or command attention (CA*nn*) keys, which are enabled on a display, should be retained when the record you are defining is displayed.

CHANGE (Change) keyword for display files

You use this record-level or field-level keyword to set on the specified response indicator for an input operation.

This is done under the following conditions:

- The keyword is specified at the record level, and any input-capable field in the record format has its modified data tag (MDT) set on.
- The keyword is specified for an input-capable field, and that field has its changed data tag (MDT) set on.

See “System/36 environment considerations for display files” on page 247 for information about how to specify the CHANGE keyword in files that are used in the System/36 environment.

The format of the keyword is:

```
CHANGE(response-indicator ['text'])
```

The MDT of an input-capable field is set on when the display station user types in the field, or when your program selects the display attribute (DSPATR(MDT)) keyword for the output operation that displays the field. If the MDT is set on using the DSPATR(MDT) keyword, the data in the field might not have changed even though the MDT (and hence the response indicator specified for CHANGE) is set on. Also, note that the MDT is set on even if the workstation user types the same data in the field as was initially displayed (such as typing into a blank field and then clearing the field).

Note: The CHANGE response indicator is not set on when a command attention key (CAnn, Help, Print, Home, or Clear) is pressed.

When the IBM i operating system detects validity checking errors and displays the record again with an error message, any CHANGE keyword response indicators that have been set on by typing into fields remain on until all validity checks succeed and the record is passed to your program.

The optional text is included on the list created at program compilation time to explain the intended use of the indicator. This text has no function in the file or the program other than as a comment. The single quotation marks are required. If you specify more than 50 characters between the single quotation marks, the text is truncated to 50 characters on the program computer printout.

Option indicators are not valid for this keyword.

Example 1

The following example shows how to specify the CHANGE keyword at the field level.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A            FLDX           5   B  8  2CHANGE(67 'FLDX was changed')
     A
00020A            FLDY           3   I  8 30CHANGE(68 'FLDY was entered')
     A
```

Example 2

The following example shows how to specify the CHANGE keyword at the record level.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A          R REC1
00020A                                      CHANGE(88 'A field was changed')
00030A*
00040A            FIELD1        10   B  3  2
00050A            FIELD2         5   B  5  2
00060A            FIELD3         6   B  7  2
00070A            FIELD4         3   I  9  2DFT('ABC')
     A
```

Related concepts

CHANGE record-level keyword

You use this record-level keyword to indicate that on an input operation, the record is to be returned to the application program only if the user has changed the record.

CHCACCEL (Choice Accelerator Text) keyword for display files

You use this field-level keyword on a single-choice selection field in a pull-down record to specify text for the accelerator key.

Note: CHCACCEL only specifies the text that should describe the accelerator key. It does not enable the function key.

The format of the keyword is:

```
CHCACCEL(choice-number accelerator-text)
```

The choice-number parameter specifies the number of the choice on the single selection field that this keyword applies to. Valid values are 1 to 99.

The accelerator-text parameter specifies the text identifying the accelerator key. The parameter can be specified in one of two forms:

- As a character string: 'Accelerator text'
- As a program-to-system field: &field-name

The field specified must exist in the same record as the selection field and must be defined as a character field with usage P. This text is placed 3 spaces to the right of the maximum length of the choice text. The maximum length of the accelerator text is determined by the length of the longest choice text. The combination of the two must not exceed the width of the smallest display size specified for the file.

The CHCACCEL keyword is allowed only on single-choice selection fields (SNGCHCFLD keyword specified on the same field) in pull-down records (PULLDOWN keyword specified at the record level).

Option indicators are not valid for this keyword.

Example

The following example shows how to specify the CHCACCEL keyword:

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A
     A          R PULLEDIT                  CF04 CF06
     A                                      PULLDOWN
     A            F1             2Y 0B  1  2SNGCHCFLD
     A                                      CHOICE(1 '>Undo')
     A                                      CHOICE(2 '>Mark')
     A                                      CHOICE(3 '>Copy')
     A                                      CHCACCEL(1 'F4')
     A                                      CHCACCEL(2 &F6);
     A            F6             2A  P
```

In this example, choice 1 has the accelerator key CF04 and choice 2 has the accelerator key CF06. When the pull-down menu is displayed, the character text F4 appears to the right of the text 'Undo', with

3 spaces in between, and the text in field F6 appears to the right of the text 'Mark', with 3 spaces in between. The longest choice text determines the length of all choice text. The same is true for the ACCEL text. The ACCEL text is then started 3 spaces to the right of the longest choice.

CHCAVAIL (Choice Color/Display Attribute when Available) keyword for display files

You use this field-level keyword to specify the color or display attributes to be used when the system is displaying the available choices in a menu bar, push button, selection field, or subfile single-choice or multiple-choice selection list.

The format of the keyword is:

```
CHCAVAIL([color] [display-attributes])
```

One parameter must be specified.

The color parameter indicates the color of the choice text for a field on a color workstation. The choice text can be specified on the following keywords:

- MNUBARCHC
- CHOICE
- PSHBTNCHC

The choice text can also come from the text displayed for a subfile used as a single choice or multiple choice selection list. The parameter is specified as an expression of the form (*COLOR value).

The valid values for the color parameter are:

Value
Meaning

BLU
Blue

GRN
Green

PNK
Pink

RED
Red

TRQ
Turquoise

YLW
Yellow

WHT
White

If the color parameter is not specified, the default color for the available choices in a menu bar is green. The default color for the available choices in a selection field is green. This parameter is ignored on a monochrome workstation.

The display-attribute parameter indicates the display attributes of the choice text specified on the MNUBARCHC or CHOICE keyword for the field. The parameter is specified as an expression of the form (**DSPATR value1 <value2 <value3...>>*).

The valid values for the display attributes are:

Value
Meaning

BL
Blink

CS
Column separator

HI
High intensity

ND
Nondisplay

RI
Reverse image

UL
Underline

The default display attribute in a menu bar is high intensity. The default display attribute in a selection field is normal (or low) intensity.

Note: Display attributes CS, HI, and BL can cause fields on 5292, 3179, 3197 Models C1 and C2, 3477 Model FC, 3486, 3487 Model HC, and 3488[1] workstations to appear as color fields. Separator lines do not appear when display attributes HI, RI, and UL are used.

The CHCAVAIL keyword is allowed on a field only if the field has one or more PSHBTNCHC, CHOICE, or MNUBARCHC keywords. It is also allowed on a subfile control record if the subfile control record uses either the SFLSNGCHC or SFLMLTCHC keywords.

Option indicators are valid for this keyword.

Example 1

The following example shows how to specify the CHCAVAIL keyword. In the example, the choices in the menu bar, if available, are displayed in yellow on a color display. For a monochrome display, the menu bar is the default color (green) at high intensity.

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
      A
      A          R RECORD                     MNUBAR
      A            F1             2Y 0B  1  2
      A                                        MNUBARCHC(1 PULLFILE 'File        ')
      A                                        MNUBARCHC(2 PULLEDIT 'Edit        ')
      A                                        CHCAVAIL((*COLOR YLW))
      A
```

Example 2

In the following example, the available choices for the selection field are displayed with underlines.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A
     A          R RECORD
     A            F1             2Y 0B  2  5SNGCHCFLD CHECK(ER)
     A                                        CHOICE(1 'Choice number 1')
     A                                        CHOICE(2 'Choice number 2')
     A                                        CHCCTL(1 &CHCCTL1);
     A                                        CHCCTL(2 &CHCCTL2);
     A                                        CHCAVAIL((*DSPATR UL))
     A
```

[1] Dependent on the monitor attached to the display device.

Example 3

In the following example, the single choice selection list is displayed in yellow on a color display. The available choices are also underlined.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A          R SFLREC                    SFL
     A            CTLFLD         1Y 0H      SFLCHCCTL
     A          R SFLCTLRCD                 SFLCTL(SFLREC)
     A                                      SFLSNGCHC
     A                                          :
     A                                          :
     A                                      CHOICE(1 'Choice number 1')
     A                                          :
     A                                          :
     A                                      CHOICE(2 'Choice number 2')
     A                                          :
     A                                          :
     A                                      CHCAVAIL((*DSPATR UL))
     A                                      CHCAVAIL((*COLOR YLW))
```

Related reference

SFLMLTCHC (Subfile Multiple Choice Selection List) keyword for display files
You use this record-level keyword to define a subfile as a multiple-choice selection list. A *multiple-choice selection list* is a scrollable group of items from which the user can select multiple items.

COLOR (Color) keyword for display files
You use this keyword to specify the color of a field on a color display.

CHCCTL (Choice Control) keyword for display files

You use this field-level keyword on a selection field to control the availability of the choices for the field.

The format of the keyword is either

```
CHCCTL(choice-number &control-field [msg-id [msg-lib/]msg-file])
or
CHCCTL(choice-number &control-field [&msg-id [&msg-lib/]&msg-file])
```

The choice-number parameter is required and it specifies the choice to which this keyword applies. Valid values are 1 to 99.

The control-field parameter is required and it specifies the name of a 1-byte numeric hidden field that, on output, contains the control value for the choice. The field must be defined within the same record as the field you are defining, and must be defined as data type Y (numeric) with length 1, decimal positions 0, and usage H. On input for multiple-choice selection fields, the selection field indicates whether the field was selected.

The following table shows the control values for the hidden field, and their meaning on input and output:

Table 3. Control values for hidden fields

Control value	Meaning on output	Meaning on input
0	Available	Unselected
1	Selected	Selected
2	Unavailable (Cannot place cursor on choice unless help for choice is available.)	
3	Unavailable (Placing cursor on choice is allowed.)	

Table 3. Control values for hidden fields (continued)

Control value	Meaning on output	Meaning on input
4	Unavailable (Cannot place cursor on choice even if help for choice is available.)	

Note: The cursor restrictions described only apply to displays that are connected to a controller that supports an enhanced interface for nonprogrammable workstations. If another display is used, the cursor is not restricted.

The message-id and message-file parameters are optional and specify a message to be displayed when the user selects an unavailable choice. If these parameters are not specified, the system issues a default message, CPD919B, when the user selects an unavailable choice. If a field is used for the message-id, that field must exist in the record you are defining and it must be defined as data type A, usage P, and length of 7.

The message-file parameter is a required parameter when the message-id parameter is used. If you do not specify the library parameter, *LIBL is used to search for the message file at program run time. If a field is used for the message library or message file, that field must exist in the record you are defining and it must be defined as data type A, usage P, and length of 10.

When the CHCCTL keyword is specified on a field, a CHOICE or PSHBTNCHC keyword with the same choice number must also be specified for the field.

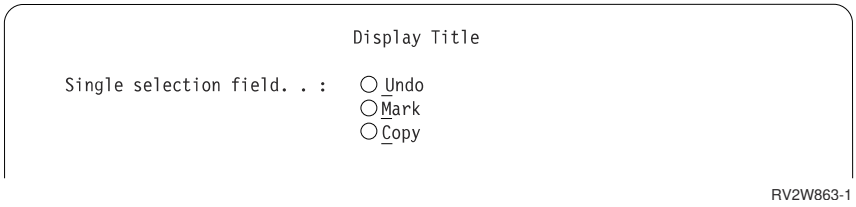
Option indicators are not valid for this keyword.

Example

The following example shows how to specify the CHCCTL keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A                                     :
     A                                     :
     A            F1             2Y 0B  3 35SNGCHCFLD
     A                                       CHOICE(1 '>Undo       ')
     A                                       CHOICE(2 '>Mark       ')
     A                                       CHOICE(3 '>Copy       ')
     A                                       CHCCTL(1 &CTLUNDO MSG1112 QUSER/A)
     A                                       CHCCTL(2 &CTLMARK &MSG &LIB/&MSGF);
     A                                       CHCCTL(3 &CTLCOPY);
     A            CTLUNDO         1Y 0H
     A            CTLMARK         1Y 0H
     A            CTLCOPY         1Y 0H
     A            MSG             7A  P
     A            MSGF           10A  P
     A            LIB            10A  P
     A
```

When using a graphical display station attached to a controller that supports an enhanced interface for nonprogrammable workstations, the selection field looks like this:



CHCSLT (Choice Color/Display Attribute when Selected) keyword for display files

You use this field-level keyword to specify the color or display attributes to be used when the system is displaying a selected choice in a menu bar or selection field.

You can use this keyword to specify the color or display attributes to be used for selected choices in a selection field, if the selection field is in a pull-down menu that has PULLDOWN (*NOSLTIND). You can also use the CHCSLT keyword on a subfile control record, when the subfile is used as a single choice or multiple choice selection list. The selected list item is displayed in the color indicated by the keyword on a color display, or displayed with the attribute indicated by the keyword.

The format of the keyword is:

```
CHCSLT([color] [display-attributes])
```

One parameter must be specified.

The color parameter indicates the color of the choice text specified on the MNUBARCHC or CHOICE keywords for the field on a color workstation. The parameter is specified as an expression of the form (*COLOR value).

The valid values for the color parameter are:

Value	Meaning
BLU	Blue
GRN	Green
PNK	Pink
RED	Red
TRQ	Turquoise
YLW	Yellow
WHT	White

If the color parameter is not specified, the default color for the selected choice in a menu bar is white. The default color for the selected choices in a selection field in a pull-down menu that does not display selection characters is white. The color parameter is ignored on a monochrome display.

The display-attribute parameter indicates the display attributes of the choice text specified on the MNUBARCHC or CHOICE keywords for the field. The parameter is specified as an expression of the form (**DSPATR value1 <value2 <value3...>>*).

The valid values for the display attributes are:

Value	Meaning
BL	Blink
CS	Column separator
HI	High intensity
ND	Nondisplay
RI	Reverse image
UL	Underline

The default display attribute for the selected choice in a menu bar is normal (or low) intensity. The default display attribute for the selected choices in a selection field in a pull-down menu that does not display selection characters is high intensity.

Note: Display attributes CS, HI, and BL can cause fields on 5292, 3179, 3197 Models C1 and C2, 3477 Model FC, 3486, 3487 Models HC, and 3488 workstations to appear as color fields. Separator lines do not appear when display attributes HI, RI, and UL are used.

When this keyword is specified for a field, either the MNUBARCHC keyword or the CHOICE keyword must also be specified on the field. If the CHOICE keyword is specified on the field rather than MNUBARCHC, the record containing this field must have the PULLDOWN keyword specified with the value *NOSLTIND. When CHCSLT is specified for a subfile control record, either the SFLSNGCHC or SFLMLTCHC keyword must also be specified on the subfile record.

Option indicators are valid for this keyword.

Example

The following example shows how to specify the CHCSLT keyword:

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A
     A          R RECORD                      MNUBAR
     A            F1             2Y 0B  1  2
     A                                        MNUBARCHC(1 PULLFILE 'File        ')
     A                                        MNUBARCHC(2 PULLEDIT 'Edit        ')
     A                                        CHCSLT((*COLOR PNK) (*DSPATR RI))
     A
```

In this example, when the choice is selected on a color display, the menu bar is displayed in pink reverse image.

Related reference

SFLMLTCHC (Subfile Multiple Choice Selection List) keyword for display files
You use this record-level keyword to define a subfile as a multiple-choice selection list. A *multiple-choice selection list* is a scrollable group of items from which the user can select multiple items.

COLOR (Color) keyword for display files
You use this keyword to specify the color of a field on a color display.

CHCUNAVAIL (Choice Color/Display Attribute when Unavailable) keyword for display files

You use this field-level keyword to specify the color or display attributes to be used when the system displays the unavailable choices in a selection field or a push button field.

This keyword can also be used to indicate unavailable choices in a subfile single or multiple choice selection list.

The format of the keyword is:

```
CHCUNAVAIL([color] [display-attributes])
```

One parameter must be specified.

The color parameter indicates the color of choice text specified on the CHOICE keywords for the field on a color display station, when the choices are unavailable. It also indicates the color of unavailable entries in a single or multiple choice selection list displayed on a color display.

The valid values for the color parameter are:

Value	Meaning
BLU	Blue
GRN	Green
PNK	Pink

[2] Dependent on the monitor attached to the display device.

Value	Meaning
RED	Red
TRQ	Turquoise
YLW	Yellow
WHT	White

If the color parameter is not specified, the default color for unavailable choices in a selection field is blue. This parameter is ignored on a monochrome display.

The display-attribute parameter indicates the display attributes of the choice text specified on the CHOICE or PSHBTNCHC keyword for the field. The parameter is specified as an expression of the form (**DSPATR value1 <value2 <value3...>>*).

Value	Meaning
BL	Blink
CS	Column separator
HI	High intensity
ND	Nondisplay
RI	Reverse image
UL	Underline

The default display attribute for unavailable choices in a selection field on monochrome display stations is normal (or low) intensity. Also, the first character of an unavailable choice on a monochrome display station is overwritten with an asterisk (*).

Note: Display attributes CS, HI, and BL can cause fields on 5292, 3179, 3197 Models C1 and C2, 3486, 3487 Model HC, and 3488[3] workstations to appear as color fields. Separator lines do not appear when display attributes HI, RI, and UL are used.

When used on a field specification, this keyword is allowed only if there are also one or more CHOICE or PSHBTNCHC keywords. When used on a subfile control record, this keyword is allowed only if the SFLSNGCHC or SFLMLTCHC keyword is also used on the subfile control record.

Option indicators are valid for this keyword.

Example

The following example shows how to specify the CHCUNAVAIL keyword:

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
      A
      A          R RECORD
      A            F1             2Y 0B  2  5SNGCHCFLD CHECK(ER)
      A                                      CHOICE(1 'Choice number 1')
      A                                      CHOICE(2 'Choice number 2')
      A                                      CHCCTL(1 &CHCCTL1);
      A                                      CHCCTL(2 &CHCCTL2);
      A                                      CHCUNAVAIL((*COLOR TRQ))
      A            :
      A            :
      A
```

In this example, the unavailable choices for the selection field are displayed in turquoise on a color display.

[3] Dependent on the monitor attached to the display device.

Related reference

SFLMLTCHC (Subfile Multiple Choice Selection List) keyword for display files
You use this record-level keyword to define a subfile as a multiple-choice selection list. A *multiple-choice selection list* is a scrollable group of items from which the user can select multiple items.

COLOR (Color) keyword for display files
You use this keyword to specify the color of a field on a color display.

CHECK (Check) keyword for display files

You use this keyword to perform a number of functions, depending on the parameter values specified.

Function
Valid parameter values

Validity checking
AB, ME, MF, M10, M10F, M11, M11F, VN, VNE

Keyboard control
ER, FE, LC, RB, RZ

Cursor control
RL, RLTB

The formats of the keyword are:

```
CHECK(validity-checking-code [. . .])
CHECK(keyboard-control-code [. . .])
CHECK(cursor-control-code)
```

The following CHECK keywords are the preferred form of other DDS keywords:

- CHECK(ER) is equivalent to AUTO(RA)
- CHECK(LC) is equivalent to LOWER
- CHECK(RB) is equivalent to AUTO(RAB)
- CHECK(RZ) is equivalent to AUTO(RAZ)

The following CHECK keyword functions can also be specified using the Change Input Default (CHGINPDFT) keyword at the file, record, or field level:

CHECK keyword
CHGINPDFT equivalent

CHECK(FE)
CHGINPDFT(FE)

CHECK(LC)
CHGINPDFT(LC)

CHECK(ME)
CHGINPDFT(ME)

CHECK(MF)
CHGINPDFT(MF)

Option indicators are valid only for CHECK(ER) and CHECK(ME).

Validity checking

Use CHECK at the field level to specify that the IBM i operating system or the device is to check the validity of the data typed into an input-capable (input-only or input/output) field.

CHECK validates the data by applying one or more edit/check algorithms against the data. An error message is displayed if a specified edit/check algorithm is not satisfied.

Note: See “CHKMSGID (Check Message Identifier) keyword for display files” on page 63 for information about defining user-specified messages.

The valid edit/check codes are:

Edit/Check code
Meaning

AB
Allow blanks

Use this code at the file level, record level, or field level to allow all-blank input to satisfy validity checking for an input-capable field when any associated validity check fails. This enables the passing of data to the program when the workstation user has positioned the cursor to the field but left it blank (for instance, by pressing the Erase Input key, the Field Exit key, or the spacebar). For example, FLD1 is an input-capable field with CHECK(M10 ME) in effect. If the workstation user accidentally types in the field, the M10 algorithm must be satisfied. By specifying (CHECK(M10 ME AB)), you enable the display station user to clear the field to satisfy validity checking.

When specified at the file level, this keyword applies for all input-capable fields in the file for which a validity checking keyword is coded. Likewise, when specified at the record level, this keyword applies for all input-capable fields in the record for which a validity checking keyword is coded. At the field level, always specify this keyword with another validity checking keyword (CHECK(M10, M10F, M11, M11F, VN, VNE), CHKMSGID, COMP, RANGE, or VALUES).

CHECK(AB) should not be specified if SFLROLVAL or SFLRCDNBR is also specified for the field.

CHECK(AB) can be used in database files for reference purposes.

When you consider using CHECK(AB) with other validity checking functions, note that processing occurs in the following order:

1. Any of the following order:
   a. The keyboard shift attribute specified in position 35 (such as alphanumeric shift or numeric only) can restrict input typing to certain characters.
   b. If the keyboard shift attribute is numeric shift, the data type (character or numeric) is set by the entry in positions 36 through 37 (decimal positions) and restricts input typing to certain characters.
   c. The CHECK(FE), CHECK(MF), and CHECK(ME) keywords, if specified, restrict input typing.
2. Either of the following order:
   a. If CHECK(AB) is specified, data management passes the input data to the program (blanks for a character field and zeros for a numeric field). No further validity checking is done.
   b. If CHECK(AB) is not specified, data management performs the following validity checking functions before passing the data to the program: CHECK(VN), CHECK(VNE), CHECK(M10), CHECK(M10F), CHECK(M11), CHECK(M11F), COMP(. . .), RANGE(. . .), VALUES(. . .).

You cannot specify the CHECK(AB) keyword on a floating-point field (F in position 35).

Option indicators are not valid for this keyword.

ME
Mandatory enter

This code specifies that at least 1 character of data (a blank is valid) must be typed into the field. Note that when no field currently on the display has been changed, the display station does not enforce mandatory enter. To enforce mandatory enter, specify DSPATR(MDT) for at least one field in each record on the display. For all other fields in the record, CHECK(ME) is then enforced. However, because the device cannot determine if the user has typed data to a field with both DSPATR(MDT) and CHECK(ME), you should also specify DSPATR(ND) so that this field is not displayed.

Option indicators are valid for this keyword.

MF
Mandatory fill

This code specifies that if any part of the field is altered, each position in the field must have a character entered in it. Blanks are considered valid characters.

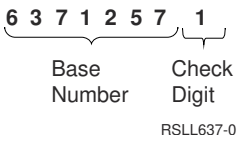
This code cannot be specified with keyboard control codes (RB or RZ) or with the WRDWRAP keyword.

Option indicators are not valid for this keyword.

M10/M10F or M11/M11F (IBM Modulus 10 or Modulus 11 Algorithm)
IBM Modulus 10 self-check algorithm

This code specifies that data typed into the field must satisfy the IBM Modulus 10 (M10 or M10F) or Modulus 11 (M11 or M11F) self-check algorithm. When you specify CHECK(M10) or CHECK(M11), the self-check verifies that the field has a valid Modulus 10 or Modulus 11 number when you press the Enter key or a function key. When you specify CHECK(M10F) or CHECK(M11F), the self-check verifies that the field has a valid Modulus 10 or Modulus 11 number as the user types the data into the field. You cannot specify both the Modulus 10 and the Modulus 11 self-check algorithms for the same field or both formats of the same algorithm for the same field.

A self-check field is composed of two parts: the base number and one check digit. The check digit is the farthest right digit in the field. The base number and the check digit together make up a field in your database (for example, an account number). The following figure is an example of an 8-digit self-check field.



See the Application Display Programming book for information about how to use CHECK(M10), CHECK(M10F), CHECK(M11), and CHECK(M11F).

Notes:

1. The IBM i operating system supports a maximum length of 31 digits for numeric fields.
2. You cannot specify the CHECK(M10), CHECK(M10F), CHECK(M11), and CHECK(M11F) keywords with the COMP(EQ) keyword.
3. You cannot specify the CHECK(M10), CHECK(M10F), CHECK(M11), and CHECK(M11F) keywords on a floating-point field (F in position 35).
4. You cannot specify the CHECK(M10F) or the CHECK(M11F) keyword in a file containing the USRDSPMGT keyword.
5. You cannot specify the CHECK(M10F) or the CHECK(M11F) keyword on a field containing the CHKMSGID or WRDWRAP keyword.

For each position in the base number, there is a Modulus 10 weight factor and a Modulus 11 weight factor. Positions are counted from the farthest right digit (not including the check digit).

The Modulus 10 weight factor is 2 for positions 1, 3, 5, ..., 31. It is 1 for positions 2, 4, 6, ..., 30. The Modulus 11 weight factors are 2, 3, 4, 5, 6, 7, 2, 3, 4, 5, 6, 7,..., 2, 3, 4, 5, 6, 7, 2 for positions 1, 2,..., 31.

To calculate the Modulus 10 self-check digit, follow these steps:

1. Multiply the units position and every alternate position of the base number by 2.
2. Add the digits in the products to the digits in the base number that were not multiplied.
3. Subtract the sum from the next higher number ending in zero.

The difference is the self-check digit.

For example:

```
Base number:                                  6       1       2       4       8
Units position and every alternate position:  6               2               8
Multiply by the weight factor, 2:             x2              x2              x2
```

```
Products:                                        12                4                  16
Digits not multiplied:                                    1                  4
Add the digits of the products and the digits
from the base number that were not used for
multiplication:                                  ( 1 + 2 ) + 4 + ( 1 + 6 ) + 1 + 4 = 19
Next higher number ending in 0:                                                      20
Subtract:                                                                            -19
Self-check digit:                                                                    1
```

To calculate the Modulus 11 self-check digit, follow these steps:

1. Assign a weight factor to each digit position of the base number. These factors are: 2, 3, 4, 5, 6, 7, 2, 3, 4, 5, 6, 7, 2, 3, . . . starting with the units position of the number and progressing toward the high-order digit. For example, the base number 991246351 can be assigned the weight factors as follows:

   ```
   Base number       9 9 1 2 4 6 3 5 1
   Weight factors    4 3 2 7 6 5 4 3 2
   ```

2. Multiply each digit by its weight factor.
3. Add the products.
4. Divide this sum by 11.
5. Subtract the remainder from 11.

The difference is the self-check digit.

For example:

Base number
1 3 7 3 9

Weight factors
x6 x5 x4 x3 x2

Multiply each digit by its weight factor
6 15 28 9 18

Add the products
6 + 15 + 28 + 9 + 18 = 76

Divide the sum by 11
76/11 = 6 plus a remainder of 10

Subtract the remainder from 11
11 - 10 = 1

Self-check digit
1

Note: If the remainder in step 4 is 0, the self-check digit is 0. If the remainder is 1, the base number has no self-check digit; you must make sure that such base numbers are not used in the fields you define as self-check fields.

Option indicators are not valid for CHECK(M10), CHECK(M10F), CHECK(M11), or CHECK(M11F).

VN

Validate name

Use this code to specify that the data typed into the field must be a valid simple name. The first character must be $, #, @, or A through Z. The remaining characters must be alphanumeric ($, #, @, A through Z, 0 through 9, or underscore (_), and must not contain embedded blanks.

When the CHECK(VN) keyword is specified on a field, the field must be character (keyboard shift of A, N, X, W, or I), and must be input-capable (usage of I or B).

CHECK(VN) cannot be specified with any of the following keywords:

CHECK(M10)	CHECK(VNE)
CHECK(M10F)	COMP
CHECK(M11)	RANGE
CHECK(M11F)	VALUES

Option indicators are not valid for this keyword.

VNE

Validate name extended

Use this code to specify that the data typed into the field must be a valid extended name.

When the CHECK(VNE) keyword is specified on a field, the field must be character (keyboard shift of A, N, X, W, or I), input-capable (usage of I or B), and have a maximum length of 255 characters.

If the name is not delimited by double quotation marks:

- The first character must be A through Z, a through z, #, $, or @.
- The remaining characters must be A through Z, a through z, #, $, _, or a period.
- Lowercase letters will be converted to uppercase.

If the name is delimited by double quotation marks:

- Any character is allowed except:

  Hex 00 through Hex 3F
  (device control)

  Hex FF
  (device control)

  Hex 40
  (blank)

  Hex 5C
  (*)

  Hex 6F
  (?)

  Hex 7D
  (')

  Hex 7F
  (")
- Lowercase letters remain lowercase.
- The system removes quotation marks when they are not needed (if the syntax of the name meets the requirements of an unquoted name, and all letters are uppercase).

CHECK(VNE) cannot be specified with any of the following keywords:

CHECK(M10)	CHECK(VN)
CHECK(M10F)	COMP
CHECK(M11)	RANGE
CHECK(M11F)	VALUES

Option indicators are not valid for this keyword.

Keyboard control

When the CHECK keyword is used with a keyboard control code, it controls certain data-entry aspects. The valid keyboard control codes are:

Keyboard control code
Meaning

ER

End of record; equivalent to AUTO(RA)

Use this code so that the workstation user does not need to press the Enter key. Whenever the workstation user keys a character (including a blank) into the last position of the field, the record is sent from the device just the same as if the Enter key had been pressed. If you also specify DSPATR(SP) for the field, the record is sent from the device as soon as the workstation user selects the field. If you use this function, it should be on the last field typed in by the user for this record.

Option indicators are valid for this keyword.

FE

Field exit check

This code specifies that the workstation user cannot advance to the next input field without pressing one of the field exit keys. The cursor remains under the low-order character position of the field until a valid field exit key has been pressed, even though that character has been typed in. If the user presses any other key, an error results.

If you want to specify CHECK(FE) for all the input-capable fields in a record format, specify CHGINPDFT(FE) at the record level. If you want to specify CHECK(FE) for all the input-capable fields in a file, specify CHGINPDFT(FE) at the file level.

Field exit keys include the Field Exit, Field+, Field-, and cursor movement keys. Which keys are valid field exit keys depends on the keyboard style being used.

This code applies only to input fields into which the workstation user can type.

Option indicators are not valid for this keyword.

LC

Lowercase; equivalent to LOWER

Use CHECK(LC) for input-only or input/output fields to permit the workstation user to type lowercase a through z. The way the workstation user types the characters (uppercase or lowercase) is the way the characters appear on the display and are returned to your program.

If you want to specify CHECK(LC) for all the character input-capable fields in a record format, specify CHGINPDFT(LC) at the record level. If you want to specify CHECK(LC) for all the character input-capable fields in a file, specify CHGINPDFT(LC) at the file level.

Your program can display a field that contains both uppercase and lowercase characters.

If you specify this keyword, lowercase a through z remain lowercase. If you do not specify this keyword, lowercase a through z are changed to uppercase.

The CHECK(LC) keyword has no effect on data-entry keyboards. Data-entry keyboards do not support lowercase characters a through z.

Option indicators are not valid for this keyword.

Figure 12 on page 57 shows how to specify the CHECK(LC) keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00100A            NAME          30   I  3  2CHECK(LC)
     A
```

Figure 12. Specifying the CHECK(LC) keyword

RB

Right-aligned with blank fill; equivalent to AUTO(RAB)

This code shifts data typed into the field to the farthest right positions and fills the remaining positions with blanks.

For signed numeric fields, you do not need to specify CHECK(RB). Right-aligned with blank fill is the default. When the value of a signed numeric field is zero, the field appears as all blanks on the display.

The IBM i program converts blanks to zeros when returning numeric fields to your program.

Option indicators are not valid for this keyword.

RZ

Right-aligned with zero fill; equivalent to AUTO(RAZ)

This code shifts data typed into the field to the farthest right positions and fills the remaining positions with zeros.

For signed numeric fields, if you do not specify CHECK(RZ), CHECK(RB) is the default.

Option indicators are not valid with this keyword.

The following list shows programming considerations for CHECK(RB) and CHECK(RZ):

- You can activate right-alignment only by pressing the Field Exit, the Field+, or the Field- key. If you use the cursor movement keys to exit from a right-aligned field, the field is not right-aligned. It is left as is.
- Right-aligned fields longer than 15 character positions slow keyboard entries.
- The Dup key fills a right-aligned field from the cursor location to the end of the field with the duplication character, but the field is not right-aligned.
- You cannot specify the CHECK(RB) or CHECK(RZ) keyword on a field containing the WRDWRAP keyword.

Example 1

The following example shows how to specify the CHECK keyword for right-aligned with blank fill (RB) and for right-aligned with zero fill (RZ).

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A          R RECORD1
00020A* Numeric only
00030A            DATA1          7Y OI  2  2TEXT('No right-adjust')
00040A            DATA2          7Y OI  3  2CHECK(RZ)
00050A* Signed numeric
00060A            DATA3          7S OI  4  2TEXT('CHECK(RB) is the default')
00070A            DATA4          7S OI  5  2CHECK(RZ)
00080A* Character
00090A            DATA5          7   I  6  2TEXT('No right-adjust')
00100A            DATA6          7   I  7  2CHECK(RB)
     A
```

When you specify the CHECK keyword for right-aligned with zero or blank fill, fill the following displays:

Field name	Data typed in	Key pressed	Result on display
Numeric only			
DATA1	1. 1 2 3 _ _ _ _	• Field Exit	• 1 2 3 _ _ _ _
	2. 1 2 3 – _ _ _	• Field Exit	• 1 2 3 – _ _ _
	3. 0 _ _ _ _ _ _	• Field Exit	• 0 _ _ _ _ _ _ _ See note.
	4. _ _ _ _ _ _ _	• Field Exit	• _ _ _ _ _ _ See note.
DATA2	1. 1 2 3 _ _ _ _	• Field Exit	• 0 0 0 0 1 2 3
	2. 1 2 3 – _ _ _	• Field Exit	• 0 0 0 1 2 3 –
	3. 0 _ _ _ _ _ _	• Field Exit	• 0 0 0 0 0 0 0
	4. _ _ _ _ _ _ _	• Field Exit	• 0 0 0 0 0 0 0
Signed Numeric			

Field name	Data typed in	Key pressed	Result on display
DATA3	1. 1 2 3 _ _ _ _ _	• Field Exit	• _ _ _ _ 1 2 3 _
	2. 1 2 3 _ _ _ _ _	• Field-	• _ _ _ _ 1 2 3 –
	3. 0 _ _ _ _ _ _ _	• Field Exit	• _ _ _ _ _ _ 0 _ See note.
	4. _ _ _ _ _ _ _ _	• Field Exit	• _ _ _ _ _ _ _ _ See note.
DATA4	1. 1 2 3 _ _ _ _ _	• Field Exit	• 0 0 0 0 1 2 3 _
	2. 1 2 3 _ _ _ _ _	• Field-	• 0 0 0 0 1 2 3 –
	3. 0 _ _ _ _ _ _ _	• Field Exit	• 0 0 0 0 0 0 0 _
	4. _ _ _ _ _ _ _ _	• Field Exit	• 0 0 0 0 0 0 0 _
	Note: The IBM i operating system converts blanks to zeros when returning numeric fields to your program. Therefore, this field is returned to your program as all zeros.		
Character			
DATA5	1. 1 2 3 _ _ _ _	• Field Exit	• 1 2 3 _ _ _ _
	2. 1 2 3 – _ _ _	• Field Exit	• 1 2 3 – _ _ _
	3. 0 _ _ _ _ _ _	• Field Exit	• 0_ _ _ _ _ _
	4. _ _ _ _ _ _ _	• Field Exit	• _ _ _ _ _ _ _
	5. A B C _ _ _ _	• Field Exit	• A B C _ _ _ _
DATA6	1. 1 2 3 _ _ _ _	• Field Exit	• _ _ _ _ 1 2 3
	2. 1 2 3 –_ _ _	• Field Exit	• _ _ _ 1 2 3 –
	3. 0 _ _ _ _ _ _	• Field Exit	• _ _ _ _ _ _ 0
	4. _ _ _ _ _ _ _	• Field Exit	• _ _ _ _ _ _ _
	5. A B C _ _ _ _	• Field Exit	• _ _ _ _ A B C

Cursor control

When the CHECK keyword is used with a cursor control code, it specifies that the cursor is to move from right to left. This feature is designed for languages where information is read right to left.

The IBM i operating system does not ensure that right-to-left files are opened only for display stations capable of right-to-left cursor movement. Therefore, all workstations in the same system should be configured with the same language capability and with the same right-to-left capability.

The valid cursor control codes that can be specified for cursor control are:

Cursor control code
Meaning

RL
Right-to-left cursor movement within fields

Use the CHECK(RL) keyword at the file, record, or field level to specify that the cursor should move from right to left within input-capable character fields. At the file level, specifying CHECK(RL) makes the cursor move from right to left in all input-capable character fields in the file. At the record level,

specifying CHECK(RL) makes the cursor move from right to left in all input-capable character fields in the record. At the field level, specifying CHECK(RL) makes the cursor move from right to left in only the field with which it is associated.

Example 2:

The following example shows how to specify the CHECK(RL) keyword at the file level.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A                                  CHECK(RL)
00020A          R DSPLY
     A
```

Example 3:

The following example shows how to specify the RL cursor control with edit check.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A             :
     A             :
     A          R RECORD1                  CHECK(RL AB)
     A             :
     A             :
     A          R RECORD2
     A            INPFLD         4   I  4 10CHECK(RL MF)
     A             :
     A             :
     A
```

Note: If you want to specify the RL cursor control code with an edit/check code, you can do so only if the edit/check code is valid at the level you specify. In the example above, CHECK(RL AB) is specified at the record level because AB is valid at that level. CHECK(RL MF) is specified at the field level because MF is valid only at that level.

RLTB

Right-to-left, top-to-bottom cursor movement from field to field

Use the CHECK(RLTB) keyword only at the file level. It specifies the direction the cursor is to advance from input-capable field to input-capable field. CHECK(RLTB) specifies that on exiting from a field, the cursor advances by moving from right to left and from top to bottom of the display until it reaches the next input-capable field. You can specify the RLTB cursor control code with only the edit/check code AB, as the others are not valid at the file level.

Note: Specifying CHECK(RLTB) does not change which input-capable field the cursor is positioned in when the display initially appears.

Example 4:

The following example shows how to specify the CHECK(RLTB) keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A                                  CHECK(RLTB)
00020A          R PROMPT
     A
```

Right-to-left capability restrictions

The right-to-left capability includes the following restrictions:

- The check digit for modulus checking is the farthest right byte in the field.
- Katakana cannot be used with right-to-left support.
- CHECK(RL) and CHECK(RLTB) cannot be specified with user-defined records (having the USRDFN keyword).
- CHECK(RL) applies only to character fields.
- You cannot specify the CHECK(RB) or CHECK(RZ) keyword on a field containing the WRDWRAP keyword.

A warning message appears for the following conditions:

- A right-to-left field that also allows magnetic card reader operator identification data (DSPATR(OID) keyword)
- A right-to-left field that spans more than one line
- A right-to-left field that is also a self-check field (CHECK(M10) or CHECK(M11) keyword)
- A right-to-left field for which CHECK(RZ) or CHECK(RB) is specified

Option indicators are not valid with cursor control codes.

Example 5

The following example shows how to specify the validity-checking CHECK keywords.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00060A          R RECORD1                   CHECK(AB)
00070A            FIELD11       10   B  1  2TEXT('CHECK(AB) not propagated to +
00080A                                        this field')
00090A            FIELD21       10   B  1 22CHECK(VN)
00100A                                       TEXT('CHECK(AB) is propagated to +
00110A                                        this field')
00120A            FIELD31       10   B  1 42CHECK(VNE)
00130A                                       TEXT('CHECK(AB) is propagated to +
00140A                                        this field')
00150A*
00160A          R RECORD2
00170A            FIELD12       10   B  2  2CHECK(VN) CHECK(AB)
00180A            FIELD22       10   B  2 22CHECK(VN AB)
00190A            FIELD32        1   B  2 42CHECK(AB) VALUES('A' 'B' 'C')
00200A            FIELD42       10   B  2 62CHECK(VN)
00210A            FIELD52       10   B  3  2CHECK(VNE)
00220A            FIELD62       10   B  3 22CHECK(VNE AB)
00230A            FIELD72       10   B  4  1CHECK(ME MF)
00240A            FIELD82        8  OB  4 22CHECK(M10)
00250A            FIELD92       10  OB  4 42CHECK(M11)
     A
```

Related reference

AUTO (Auto) keyword for display files
Under some circumstances, the AUTO keyword is equivalent to the CHECK keyword.

LOWER (Lower) keyword for display files
The LOWER keyword is equivalent to the CHECK(LC) keyword.

CHGINPDFT (Change Input Default) keyword for display files

You use this file-level, record-level, or field-level keyword to change one or more input defaults for input-capable fields.

Without parameter values, this keyword removes the underline for input-capable fields (input only or input/output). With parameter values, this keyword applies the specified display attributes or keyboard controls to the affected input-capable fields.

The format of the keyword is:

```
CHGINPDFT[(input-default1 input-default2 . . .)]
```

Valid parameter values for this keyword are:

Parameter value	Equivalent DDS keyword	Meaning
None	DSPATR(UL) specified but not selected	Remove underline
BL	DSPATR(BL)	Blinking field
CS	DSPATR(CS)	Column separators

Parameter value	Equivalent DDS keyword	Meaning
HI	DSPATR(HI)	High intensity
RI	DSPATR(RI)	Reverse image
UL	DSPATR(UL)	Underline
FE	CHECK(FE)	Field exit
LC	CHECK(LC) or LOWER	Lowercase
ME	CHECK(ME)	Mandatory enter
MF	CHECK(MF)	Mandatory fill

Note: If DSPATR(UL) is specified for a field, the CHGINPDFT keyword cannot control underlining for that field.

The above equivalent DDS keywords apply only to output fields. For input and both fields, DSPATR(UL) must also be specified but not selected in addition to the equivalent keywords. This is because DSPATR(UL) is applied to input and both fields by default when CHGINPDFT is not specified.

Two common ways to use this keyword are to allow lowercase data entry for all input-capable fields in a record format or file, and to specify column separators for all input-capable fields in a record format or file.

At the file level, this keyword applies to all input-capable fields in the file. At the record level, this keyword applies to all input-capable fields in the record format. At the field level, this keyword applies only to the fields for which it is specified.

If you specify the CHGINPDFT keyword at more than one level, the lower level keyword overrides the higher level keyword. Thus, specifying CHGINPDFT(BL) at the file level and CHGINPDFT(HI) for a record format causes all input-capable fields in the file except those in that record format to blink. In that record format, all input-capable fields are highlighted.

The CHGINPDFT keyword can be specified with any CHECK or DSPATR keyword. If you specify CHGINPDFT at the file, record, or field level, you can add check codes or display attributes to single fields by specifying CHECK or DSPATR at the field level. For instance, if you specify CHGINPDFT(CS) at the record level and DSPATR(HI) at the field level, the field is displayed with column separators and is highlighted. In addition, the CHECK or DSPATR keyword at the field level controls the check code or display attribute specified with it. For example, if you specify CHGINPDFT(CS) at the record level and DSPATR(CS) with option indicators at the field level, the setting of the option indicators controls the column separators for the field.

If you display a field with UL, RI, and HI in effect, no matter whether specified on the CHGINPDFT keyword, the DSPATR keyword, or a combination of both, the field is not displayed.

When specified at the file or record level, CHGINPDFT(LC) does not apply to numeric fields. If specified for a numeric field, CHGINPDFT(LC) is ignored.

CHGINPDFT(MF) is not allowed with CHECK(RB), CHECK(RZ), AUTO(RAB), AUTO(RAZ), or WRDWRAP keyword.

Option indicators are not valid for this keyword.

Example

The following example shows how to specify the CHGINPDFT keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A          R RECORD1                   CHGINPDFT
00020A            FLD1          10   B  1  2
00030A            FLD2          10      2  2
00040A          R RECORD2                    CHGINPDFT(CS)
00050A            FLD3          10   I  3  2
00060A            FLD4          10   B  4  2
00070A            FLD5          10   B  5  2
```

```
00080A  01                                  DSPATR(CS)
00090A            FLD6          10      6  2
00100A          R RECORD3                   CHGINPDFT(CS)
00110A            FLD7          10   I  7  2
00120A            FLD8          10   I  8  2
00130A  02                                  DSPATR(HI)
     A
```

CHGINPDFT is specified at the record level for RECORD1, RECORD2, and RECORD3:

- For RECORD1, CHGINPDFT removes the underline for FLD1.
- For RECORD2, CHGINPDFT can have the following results:
  - FLD3 and FLD4 have column separators.
  - FLD5 has column separators only when DSPATR(CS) is selected.
  - FLD6 (an output-only field) has no column separators.
- For RECORD3, CHGINPDFT can have the following results:
  - FLD7 and FLD8 have column separators.
  - FLD8 is also highlighted when DSPATR(HI) is selected.

CHKMSGID (Check Message Identifier) keyword for display files

You use this field-level keyword to identify an error message that is issued when a validity check error is detected.

If you do not specify the CHKMSGID keyword, the system supplies a message. You specify the associated validity checking rules on a CHECK(M10), CHECK(M11), CHECK(VN), CHECK(VNE), CMP, COMP, RANGE, or VALUES keyword.

The format of the keyword is:

```
CHKMSGID(message-id [library/]message-file [&message-data-field])
```

The message-ID parameter specifies the message description that contains the text to be displayed on the message line.

The message-file and library parameters identify the message file containing the message descriptions. The library name is optional. If it is not specified, the library list (*LIBL) that is in effect at run time is used to search for the message file.

The message-data-field parameter specifies the name of the field that contains the message replacement text to be displayed on the message line. The format of the message-data-field parameter is &field-name where field-name is the name of the field containing the message replacement text. The field name must exist in the record format, and the field must be defined as a character field (data type A) with usage P.

CHKMSGID is allowed only on fields which also contain a CHECK(M10), CHECK(M11), CHECK(VN), CHECK(VNE), CMP, COMP, RANGE, or VALUES keyword. The field must be input-capable (usage B or I).

Option indicators are not valid for this keyword.

Example

The following example shows how to specify the CHKMSGID keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A                                      MSGLOC(20)
00020A          R RECORD1
00030A            FIELD1        10A  B  4  2CHECK(VN) CHKMSGID(USR1234 +
00040A                                      QGPL/USRMSGS &MSGFLD1);
00050A            MSGFLD1       12A  P
00060A            FIELD2         1A  I  4 20VALUES('A' 'B' 'I')
00070A                                      CHKMSGID(XYZ9999 APPLMSGS)
00080A            FIELD3         3S OB  4 25RANGE(023 199)
     A
```

When RECORD1 is read from the display screen:

- If FIELD1 does not contain a valid name, message USR1234 from the message file USRMSGS in library QGPL with the replacement text specified in MSGFLD1 is displayed on line 20.
- If the data entered into FIELD2 is not the letter A, B, or I, message XYZ9999 from *LIBL/APPLMSGS is displayed on line 20.
- If the data entered into FIELD3 is less than 023 or greater than 199, the system-supplied message CPF5224 (value for the field is not in a valid range) is displayed on line 20 because the CHKMSGID keyword is not specified.

Related reference

VALUES (Values) keyword for display files
You use this field-level keyword to specify a list of values that are valid for the user to type into the field.

CHOICE (Selection Field Choice) keyword for display files

You use this field-level keyword to define a choice for a selection field.

The format of the keyword is:

```
CHOICE(choice-number choice-text [*SPACEB])
```

The choice-number parameter defines an identification number for this choice. This parameter is required. The choice number returns to the application to indicate which choice in the selection field was selected. On non-graphical displays, the choice number is also displayed to the left of the choice text. Valid values for the choice-number are positive integers greater than 0 and less than or equal to 99. Duplicate choice-number values within a selection field are not allowed.

The choice-text parameter defines the text that appears in the selection field for the choice. This parameter is required. The parameter can be specified in one of two forms:

- As a character string: 'Choice text '
- As a program-to-system field: &field-name

The field specified must exist in the same record as the selection field and must be defined as a character field with usage P.

The choice text for all choices within a selection field must fit on the display for the smallest display size specified in the file. Therefore, the maximum length for the choice text depends on the following items:

- The position of the selection field
- The length of the longest choice number that is displayed to the left of the choice
- The length of the choice text itself
- The number of columns in the selection field
- The width of the gutter between columns

If the smallest display size is 24 x 80, the above must be less than or equal to 80. If the smallest display size specified is 27 x 132, this sum must be less than or equal to 132.

Within the choice text, you can specify a mnemonic for the choice by using a greater-than character (>) to indicate the mnemonic character. The character to the right of the > sign is the mnemonic. The mnemonic is used only on a character-based graphical display attached to a controller that supports an enhanced interface for nonprogrammable workstations, where the choices are rendered using radio buttons. The mnemonic is ignored on displays where the field is rendered using numeric selection because the system does not support both numeric and mnemonic selection on a selection field. The following examples show how to specify mnemonics.

Choice text
Appears as

'>File'
File

'F>inish'
F_i_nish

'Save >As...'
Save _A_s...

'X >= 1'
X _=_ 1

In order to specify > as a character in the text, you must specify it twice, just as you must specify the apostrophe character twice in order to get a single apostrophe character in the text. For example:

Choice text
Appears as

'X >>= 1'
X >= 1

'X >>>= 1'
X >_=_ 1

Note: It is not possible to specify the > sign as the mnemonic.

The mnemonic character indicated must be a single-byte character and must not be a blank. Only one mnemonic is allowed in the choice text, and the same mnemonic character cannot be specified for more than one choice.

The *SPACEB parameter is optional and indicates that a blank space (or line) should be inserted before this choice. This parameter is used to specify logical grouping of choices that are numbered consecutively.

For vertical selection fields (selection fields arranged in a single column), if the choice numbers are not consecutive, a blank space is automatically inserted between non-consecutive choices. This does not happen for horizontal selection fields (selection fields arranged in multiple columns).

When the CHOICE keyword is specified on a field, either the SNGCHCFLD or the MLTCHCFLD keyword must also be specified.

Several CHOICE keywords can be specified for one selection field. The maximum number of CHOICE keywords that can be specified depends on the position of the selection field and the display size. All choices must fit on the smallest display size specified for the file.

Option indicators are valid for this keyword. When a CHOICE keyword is turned off, the list of choices is compressed.

Example

The following example shows how to specify the CHOICE keyword:

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A          R RECORD
     A            F1             2Y 0B  1  2SNGCHCFLD
     A  01                                   CHOICE(1 '>Undo       ')
     A                                       CHOICE(2 &MARKTXT);
     A                                       CHOICE(3 '>Copy       ')
     A            MARKTXT        12A  P
     A
```

In this example, three choices are defined for the single-choice selection field F1. The text for choice 2 is contained in field MARKTXT, and the mnemonic for choice 2 must be contained in the text supplied by the application at run time. If indicator 01 is off when the record is written, only choices 2 and 3 are displayed.

CHRID (Character Identifier) keyword for display files

You can use this field-level keyword to specify that a named field be translated if the CHRID parameter value for the display file differs from the CHRID parameter value for the workstation.

This can be important when extended alphabetics (characters such as u with an umlaut or c with a cedilla) are to be displayed or typed in.

This keyword has no parameters.

If the CHRID keyword is not specified for a field and the CHRID value for the display file is not *JOBCCSID, data displayed in that field is displayed in the character set of the device used to type the data. How the data is displayed cannot be predicted and depends on how code points used in the original code page map to the code page used on the device.

The CHRID keyword is not valid on constant fields, numeric fields (fields with decimal positions specified in positions 36 through 37), message fields (M specified in position 38), hidden fields (H specified in position 38), or program-to-system fields (P in Position 38).

The CHRID keyword is ignored if the CHRID value for the display file is *JOBCCSID.

The CHRID keyword cannot be specified with the DUP (Duplication) keyword.

If you specify the CHRID keyword with the DFT keyword on a field, the initial (default) value of the field is not translated, but data entered into that field is translated.

Option indicators are not valid for this keyword, although option indicators can be used to condition the field for which it is specified.

Example

The following example shows how to specify the CHRID keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A          R RECORD1
00020A            TITLE         40      1 20CHRID
     A
```

The field TITLE is a named field. With the CHRID keyword specified, character translation can occur on both output and input, depending on the conditions described in the Application Display Programming book.

CLEAR (Clear) keyword for display files

You use this file-level or record-level keyword to specify that your program is to receive control if the workstation user presses the Clear key, and optionally, that the IBM i operating system is to set on the specified response indicator.

See “System/36 environment considerations for display files” on page 247 for special considerations when you specify the CLEAR keyword in files that are used in the System/36 environment.

The format of the keyword is:

```
CLEAR[(response-indicator ['text'])]
```

The Clear key is processed like a command attention key (no input data is transmitted from the device). The IBM i operating system does not clear the display; your program must perform the required function (such as clearing fields or records from the display).

If you do not specify this keyword and the display station user presses the Clear key, the IBM i program displays a message indicating that it is not valid at that time.

Note: On display stations with the typewriter-like keyboard, the Clear key is activated by pressing CMD, then pressing the Shift key and the left arrow above the Field Exit key. On workstations with the data-entry keyboard, press CMD, then press the Shift key and the farthest right blank key on the top row.

The optional text is included on the printout created at program compilation time to explain the intended use of the indicator. This text has no function in the file or the program other than as a comment. The single quotation marks are required. If you specify more than 50 characters between the single quotation marks, the text is truncated to 50 characters on the program printout.

Option indicators are valid for this keyword.

Example

The following example shows how to specify the CLEAR keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00101A                                  CLEAR(10 'Clear key pressed')
     A
```

CLRL (Clear Line) keyword for display files

You use this record-level keyword to specify that the IBM i operating system is to clear (delete) a specific number of lines before the record is displayed. Only those lines are cleared.

Note: As with OVERLAY, other records remain on the display.

See the Application Display Programming book for information about how to use CLRL in files that are used in the System/36 environment.

The format of the keyword is:

```
CLRL(nn|*END|*NO|*ALL)
```

You can specify the CLRL keyword in one of the following ways:

- Specify nn, where nn is an integer between 1 and 27. The number specified is the number of lines cleared, starting with and including the first line on which the record is to be displayed. If the SLNO (Starting Line Number) keyword is also specified for this record format, the clearing of lines begins with the starting line number in effect for the record format at the time it is displayed.

  Note: When you specify nn, the record must have at least one field defined.
- Specify *END to indicate that all lines starting with and including the first line on which the record is to be displayed are to be cleared. For a 24 x 80 display, lines up to and including line 24 are cleared. For a 27 x 132 display, lines up to and including line 27 are cleared.
- Specify *NO so that no lines on the display are cleared before displaying the record whose format you are defining. The displayed record overlays any data already on the display.
- Specify *ALL so that all of the lines on the display are cleared before displaying the record whose format you are defining. At least one field must be defined in the record format.

When a record format begins with a field in position 1, the beginning attribute byte of the format is in the last position of the previous line. The previous line number is the starting line number in the format. This also applies to a SLNO format with a field defined in the DDS in line 1, position 1.

If the record format for which the CLRL keyword is specified has one or more input-capable fields, any records that are overlaid are no longer recognized by the IBM i operating system. That is, any input-capable fields can no longer be typed into, any input operation written to one of those records results in an error, and they cannot be cleared by selecting the ERASE keyword.

If you specify the CLRL(nn) keyword in a record format without input-capable fields, the input-capable fields in the overlapped records remain input-capable. That is, input-capable fields in the overlaid records remain input-capable, and input operations written to those record formats are still valid. If the ROLLUP or ROLLDOWN keywords are specified on the record containing the CLRL keyword, they are ignored. Records with the CLRL keyword and no input-capable fields are not cleared properly when they are overlaid by other records that have the OVERLAY keyword specified. The lines needed for the overlapping record are cleared and the lines not needed for the overlapping record remain on the display.

You can use the CLRL(*NO) keyword to prevent an overlapped record from being cleared when the overlapping record is written to the display. If you use this keyword, any records being displayed that are to be overlapped are not cleared from the display. The new record overlays them entirely or partially. There is a performance advantage to using CLRL(*NO) if you have a display containing constants and data that is repeatedly sent to the display. Sending constants as a separate format and using the CLRL(*NO) keyword for the format containing the data reduces the time required to send the record format to the display.

If the CLRL keyword is not specified and neither OVERLAY nor PUTOVR (Put with Explicit Override) is specified, the entire display is cleared.

If the CLRL keyword is used and the PUTOVR or PUTRETAIN keyword is in effect, the clearing of any lines might conflict with the PUTOVR or PUTRETAIN function. The PUTOVR or PUTRETAIN keyword requires that the fields being overridden be on the display, while the CLRL(nn) or CLRL(*END) keyword clears those fields first. If a record becomes unavailable for input because of the CLRL(nn) or CLRL(*END) keyword, the input-capable fields remain input-capable if the PUTOVR keyword is in effect. However, the IBM i operating system sends a message if the program attempts to read such a record.

If you specify the CLRL keyword, you should also specify RSTDSP(*YES) on the Create Display File (CRTDSPF) or Change Display File (CHGDSPF) command. Otherwise, data on the display can be lost if the file is suspended.

Preventing overlapped records from being cleared

The CLRL keyword cannot be specified with any of the following keywords:

ASSUME	SFLCTL
KEEP	USRDFN
SFL	

A warning message appears at file creation time if the CLRL keyword is specified on a record with the DSPMOD keyword. At run time, the CLRL keyword is ignored when the display mode changes.

The CLRL keyword cannot be specified for the record format specified by the PASSRCD keyword.

Option indicators are not valid for this keyword.

Example

The following example shows how to specify the CLRL keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A          R RECORD1                   CLRL(5)
00020A            FLD1           5      3  2
00030A            FLD2          10  OB  5  2
00040A            FLD3          10   I  6  2
00050A*
00060A          R RECORD2                   CLRL(*NO)
00070A            FLD1           5  2   2  2
00080A            FLD2           5   H
00090A            FLD3          10   I  4  2
00100A*
00110A          R RECORD3                   CLRL(*END)
00120A            FLD1           5   B  5  2
00130A            FLD2           5   I  8  2
     A
```

Lines 3, 4, 5, 6, and 7 are cleared before RECORD1 is displayed. In RECORD2, no lines are cleared, and when the record is displayed, it will overlay anything already displayed. Lines 5 through 24 are cleared before RECORD3 is displayed.

CMP (Comparison) keyword for display files

This keyword is equivalent to the COMP keyword.

The format of the keyword is:

```
CMP(relational-operator value)
```

The COMP keyword is preferred.

Related reference

COMP (Comparison) keyword for display files
You use this field-level keyword to specify that the IBM i operating system is to compare the data that the workstation user types into an input or input/output field with the specified value.

CNTFLD (Continued-Entry Field) keyword for display files

You use this field-level keyword to define a field as a continued-entry field.

Continued-entry fields are sets of associated entry fields that are treated by the workstation controller as a single field during field-data entry and editing. If the display device is not attached to a controller that supports an enhanced interface for nonprogrammable workstations, each segment of the continued entry field is treated separately when editing is performed on the field.

Figure 13 on page 69 illustrates the use of continued fields to create a rectangular text entry field.

```
Enter Text . . .  ________________________________________
                  ________________________________________
                  ________________________________________
                  ________________________________________
                  ________________________________________
```

Figure 13. Continued-Entry field in rectangular arrangement

The text input format is more appealing to the user than a single input field that wraps across multiple display lines. Even though the last line does not occupy the full width of the column, no other field is allowed in the rectangle. A continued-entry field allows a multiple-row entry field to be defined inside a window.

The format of the keyword is:

```
CNTFLD(width of column)
```

One parameter must be specified.

The width of the column parameter specifies the number of columns to be used for this continued field. This value must fit within the width of the display or window. This value must be less than the length of the field.

The field containing the CNTFLD keyword must be defined as an input-capable field with the data type A. It cannot be defined in a subfile.

The following keywords cannot be specified on a field with the CNTFLD keyword:

- AUTO (RAB, RAZ)
- CHECK(AB, MF, RB, RZ, RLTB)
- CHOICE
- DSPATR(OID SP)
- EDTMSK

The CNTFLD keyword must be defined with at least 2 spaces separating it from other fields.

Option indicators are not valid for this keyword.

Although the maximum number of input fields is 256, the use of the CNTFLD keyword can impact the field count. The segments of CNTFLD fields are treated at run time as separate input fields on some emulators or controllers. Some controllers consider each segment of the field as an actual input field, and each segment counts toward the 256 maximum. However, some emulators count a CNTFLD field as a single field toward the count of 256. But, they might keep track of the segments separately. One emulator is known to have a maximum of 512 segments allowed. Therefore, when an emulator and a controller use the same display file, the emulator might allow more CNTFLD fields than the controller.

See "CNTFLD (Continued-Entry Field) keyword" on page 258 for information about using this keyword with DBCS data.

Example

The following example shows how to specify the CNTFLD keyword:

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A          R RECORD
00020A            F1            90A  B  3  4CNTFLD(30)
```

In this example, a multiple-row entry field is defined. The entry field contains 3 lines and is 30 columns wide.

COLOR (Color) keyword for display files

You use this keyword to specify the color of a field on a color display.

This field-level keyword specifies the color of a field on a color display (3179, 3197 Models C1 and C2, 3477 Model FC, 3486, 3487 Model HC, 3488[4] or 5292 Color Display Stations only). This keyword is ignored if it is selected for a field displayed on monochrome display stations. You can specify one parameter value for the COLOR keyword, but you can specify more than one COLOR keyword on each field.

The format of the keyword is:

```
COLOR(GRN | WHT | RED | TRQ | YLW | PNK | BLU)
```

The valid parameter values are:

Value
Meaning

GRN
Green

WHT
White

RED
Red

TRQ
Turquoise

YLW
Yellow

PNK
Pink

BLU
Blue

[4] Dependent on the monitor attached to the display device.

Because green is the default color of the fields on color display stations, you need to specify COLOR(GRN) only to keep the color of a field green. Specifying DSPATR(HI), DSPATR(CS), or DSPATR(BL) for a field changes the color of the field unless you also specify COLOR(GRN).

Option indicators are valid for this keyword.

When you specify the COLOR keyword more than once for a field, you must specify option indicators with each COLOR keyword. If more than one COLOR keyword is in effect for an output operation, the IBM i operating system uses the first COLOR keyword that is specified in the DDS (see "Example 1" on page 73). You cannot specify the same color more than once for a field.

The number of COLOR keywords you can specify in one display file is limited by the maximum size of an internal storage area of the system called the screen attribute array. The maximum size of the screen attribute array is 32 763 bytes for the entire display file. Each COLOR keyword you specify in the file uses up a significant amount of storage within this array.

If you use many COLOR keywords in a file, particularly with conditioning, you should consider the amount of internal storage these keywords will require. If the 32 763-byte limit is exceeded, message CPF0673 (`Too many COLOR or DSPATR keywords specified in file`) is issued during file creation. To determine the amount of storage required for a particular COLOR keyword, use the following algorithm:

```
(# of conditions for the keyword) x 2 + 2 + 2^9 = # of bytes required
in the screen
attribute array
for the keyword
```

For example, suppose a file contains 8 fields, each field contains 9 COLOR keywords, and each COLOR keyword is optioned using 3 conditions. Using the above algorithm, each COLOR keyword requires 520 bytes in the screen attribute array:

```
3 x 2 + 2 + 2^9 = 520 bytes
```

Because there are nine COLOR keywords per field and eight fields in the file, the total storage required in the screen attribute array is 37 440 bytes (520 x 9 keywords x 8 fields). Because 37 440 is greater than 32 763, message CPF0673 is issued at file-creation time.

Using the COLOR keyword with the DSPATR keyword

In some combinations of COLOR and DSPATR, both keywords have effect. Those combinations are:

COLOR
DSPATR

Any
RI (reverse image)

Any
UL (underline)

RED
BL (blinking field)

RED
BL and RI

RED
BL and UL

RED
RI and UL

GRN
RI and UL

TRQ
RI and UL

PNK
RI and UL

For example, if COLOR(YLW) and DSPATR(RI) are both in effect, the field appears as black characters on a yellow background.

In some combinations of the COLOR and DSPATR keywords, some of the parameter values are ignored. Those combinations are shown in the following table.

COLOR	DSPATR	Effect
Any Any Any	ND (nondisplay) HI (high intensity) CS (column separators)	All colors are ignored HI is ignored CS is ignored[1]
GRN WHT TRQ YLW PNK BLU	BL BL BL BL BL BL	BL is ignored[2] BL is ignored[2] BL is ignored[2] BL is ignored[2] BL is ignored[2] BL is ignored[2]
RED	RI and BL and UL	UL is ignored[3]
YLW BLU WHT	RI and UL RI and UL RI and UL	RI is ignored RI is ignored RI is ignored

[1]Turquoise and yellow fields have column separators even if DSPATR(CS) is not specified. (The column separators appear as small blue dots between characters on color displays. They disappear when the display station user sets the color display station for reduced line spacing.)

[2]The only color that can blink is red.

[3]Underlines are also removed from input-capable fields, which are underlined by default.

For example, if COLOR(YLW) and DSPATR(HI) are both selected for an output operation, the field is yellow but not high intensity.

Using the DSPATR keyword on color displays

When you specify the DSPATR keyword without the COLOR keyword, fields are displayed on color displays with the colors in the following table but without the specified display attributes.

Table 4. DSPATR keyword on color displays

DSPATR(CS) display attribute selected	DSPATR(HI) display attribute selected	DSPATR(BL) display attribute selected	Color produced on the color display station
			Green (normal)
X			Turquoise[1]
	X		White
		X	Red, no blinking
	X	X	Red, with blinking
X	X		Yellow[1]
X		X	Pink

Table 4. DSPATR keyword on color displays (continued)			
DSPATR(CS) display attribute selected	**DSPATR(HI) display attribute selected**	**DSPATR(BL) display attribute selected**	**Color produced on the color display station**
X	X	X	Blue

[1]Turquoise and yellow fields are displayed with column separators (which are always blue) except when the workstation user sets the color display station for reduced line spacing.

For example, if DSPATR(HI) is selected for a field and the COLOR keyword is not specified, the field is white but not highlighted on a color display.

The COLOR keyword is ignored if it is selected for a monochrome display.

Example 1

The following example shows the effects of specifying COLOR and DSPATR for a field.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A          R RECORD                    1
00020A                                    1  2'Column Heading'
00030A                                       DSPATR(HI)
00040A       2  FIELD1         5      3  2
00050A       3  FIELD2         5   I  5  2COLOR(YLW)
00060A       4  FIELD3         5      7  2DSPATR(BL)
00070A       5  FIELD4         5   I  9  2
00080A  42                                   COLOR(YLW)
00090A  43                                   COLOR(TRQ)
00100A  44                                   COLOR(BLU)
     A
```

1

On color displays, the constant field *Column Heading* is white; on monochrome displays, it is highlighted.

2

On all displays, FIELD1 is green.

3

On color displays, FIELD2 is yellow with blue column separators. On all displays, the field is underlined because it is an input-capable field.

4

On color displays, FIELD3 is red and does not blink; on monochrome displays, it blinks.

5

On color displays, FIELD4 can appear in one of the following colors:

- Green, if no indicators are on
- Yellow, if indicator 42 is on (no matter how other indicators are set)
- Turquoise, if indicator 43 is on and indicator 42 is off
- Blue, if indicator 44 is the only indicator on

On monochrome displays, FIELD4 is green. On all displays, FIELD4 is underlined.

Example 2

The following example shows one way of specifying a field for use as an input-capable field on both color and monochrome displays.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A          R RECORD
00020A 2 1      FIELDA         5   B  2  2COLOR(TRQ)
00030A  44                                ERRMSG('Record not found' 44)
     A
```

1
On color displays, FIELDA is turquoise with blue column separators; on monochrome displays, it is green.

2
If option indicator 44 is set on when FIELDA is displayed, the ERRMSG keyword is in effect and has the following effect:

- On color displays, FIELDA is turquoise and its image is reversed. (Because of the COLOR keyword, it is not highlighted.) The error message `Record not found` is displayed on the message line in white.
- On monochrome displays, FIELDA is highlighted and its image is reversed. The error message `Record not found` is also highlighted and is displayed on the message line.

Related reference

CHCAVAIL (Choice Color/Display Attribute when Available) keyword for display files
You use this field-level keyword to specify the color or display attributes to be used when the system is displaying the available choices in a menu bar, push button, selection field, or subfile single-choice or multiple-choice selection list.

CHCSLT (Choice Color/Display Attribute when Selected) keyword for display files
You use this field-level keyword to specify the color or display attributes to be used when the system is displaying a selected choice in a menu bar or selection field.

CHCUNAVAIL (Choice Color/Display Attribute when Unavailable) keyword for display files
You use this field-level keyword to specify the color or display attributes to be used when the system displays the unavailable choices in a selection field or a push button field.

DSPATR (Display Attribute) keyword for display files
You use this field-level keyword to specify one or more display attributes for the field that you are defining.

ENTFLDATR (Entry Field Attribute) keyword for display files
You use this field-level, record-level, or file-level keyword to define that the leading attribute of the field changes to a specified attribute whenever the cursor is located in the field.

ERRMSG (Error Message) and ERRMSGID (Error Message Identifier) keywords for display files
You can use one of these field-level keywords to identify a message that is displayed on the message line and that is associated with that field.

MNUBARSEP (Menu-Bar Separator) keyword for display files
You use this field-level keyword on a menu-bar field to specify the color, display attributes, or the character that is used to form the menu-bar separator line.

WDWBORDER (Window Border) keyword for display files
You use this file-level or record-level keyword to specify the color, display attributes, and characters used to form the border of a window.

COMP (Comparison) keyword for display files

You use this field-level keyword to specify that the IBM i operating system is to compare the data that the workstation user types into an input or input/output field with the specified value.

The relational operator is the criterion for the comparison. If the data typed in this field fails this validity check, the IBM i operating system displays an error message. Note that the IBM i operating system performs this checking only if the field is changed by the workstation user or if its modified data tag (MDT) is set on using DSPATR(MDT).

Note: See “CHKMSGID (Check Message Identifier) keyword for display files” on page 63 for information about defining user-specified error messages.

The format of the keyword is:

```
COMP(relational-operator value)
```

You can specify only one operation for the COMP keyword and only one COMP keyword for a field.

The valid entries for the relational operator are:

Relational operator
Meaning

EQ
Equal

NE
Not equal

LT
Less than

NL
Not less than

GT
Greater than

NG
Not greater than

LE
Less than or equal

GE
Greater than or equal

The specified value must be either numeric or character, depending on the data type (decimal positions entry). Numeric values are expressed by the digits 0 through 9 and a leading sign (+ or -). Character values must be enclosed in single quotation marks.

Note: If the field you are defining is numeric, alignment is based on the decimal positions specified (in positions 36 and 37), and leading and trailing blanks are filled with zeros. If no decimal point is typed in, the decimal point is assumed to be to the right of the last (farthest right) digit. For example, for a numeric field with a length of 5 (specified in position 34) and 2 decimal positions (specified in position 37), 1.2 is interpreted as 001.20, and 100 is interpreted as 100.00.

You cannot specify the COMP keyword on a floating-point field (F in position 35).

Option indicators are not valid for this keyword.

Example

The following example shows how to specify the COMP keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A            FIELD2         6  OI 10 10COMP(EQ +021920)
00020A            FIELD1         3   I 11 11COMP(EQ 'ABC')
     A
```

Related reference

CMP (Comparison) keyword for display files
This keyword is equivalent to the COMP keyword.

CSRINPONLY (Cursor Movement to Input-Capable Positions Only) keyword for display files

You use this file-level or record-level keyword to restrict cursor movement to input-capable positions only. This keyword only affects cursor movement caused by using the arrow keys.

This keyword has no parameters.

You should take care when defining help when this keyword is in effect. The user might not be able to position the cursor in the area where help is valid.

See the Application Display Programming book for more information about the CSRINPONLY keyword.

Option indicators are valid for this keyword.

Example

The following example shows how to specify the CSRINPONLY keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A                                     CSRINPONLY
     A          R RECORD1
     A                                       1 10'ONE--:'
     A            FIELD1        10A  I  1 20TEXT('ONE')
     A                                       2 10'TWO--:'
     A            FIELD2        10A  I  2 20TEXT('TWO')
     A                                       3 10'THREE--:'
     A            FIELD3        10A  I  3 20TEXT('THREE')
     A                                       4 10'FOUR--:'
     A            FIELD4        10A  I  4 20TEXT('FOUR')
     A                                       5 10'OUT--:'
     A            FIELD5        10A  O  5 20TEXT('OUT')
     A
```

This example shows RECORD1 is defined with input, output, and constant fields. Because CSRINPONLY is specified, the user will only be able to position the cursor in either FIELD1, FIELD2, FIELD3, or FIELD4. FIELD5 and all other areas of the display are not accessible by the cursor.

CSRLOC (Cursor Location) keyword for display files

You use this record-level keyword to specify the cursor location on an output operation to the record format that you are defining. Your program sends the output operation after setting the cursor location.

The format of the keyword is:

```
CSRLOC(field-name-1 field-name-2)
```

The parameter values on the keyword specify the names of two fields whose contents are the line number (for field-name-1) and the position number (for field-name-2) of the cursor location. Field-name-1 and field-name-2 are 3-byte, zoned decimal, hidden fields. Your program uses these fields to tell the IBM i operating system where to locate the cursor.

The cursor is not positioned to the required location on an output operation that leaves the keyboard locked. The cursor does not move to the required position until your program sends an input or an output operation that unlocks the keyboard. If your program sets the cursor location fields to values outside the range of values valid for the display device, this keyword is ignored.

For any one output operation, the CSRLOC keyword overrides any other cursor location specifications, such as DSPATR(PC) and SFLRCDNBR(CURSOR), that are in effect. This keyword is in effect until your program sends another output operation with DSPATR(PC), CSRLOC, or SFLRCDNBR(CURSOR) in effect or until the record in which this keyword is specified is overlaid (OVERLAY keyword) or erased (ERASE keyword), whichever comes first.

On an input operation, the cursor location can be determined by looking at the I/O feedback area or specifying the appropriate parameter on the RTNCSRLOC keyword. See the Application Display Programming book for more information about the I/O feedback area.

Specify the CSRLOC keyword only once per record format.

The CSRLOC keyword is not valid for the following record formats:

- Subfile record formats (identified by the SFL keyword)
- User-defined record formats (identified by the USRDFN keyword)

Option indicators are valid for this keyword. Display size condition names are not valid.

Example

The following example shows how to specify the CSRLOC keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A          R RECORD1                   CSRLOC(LINNBR POSNBR)
00020A            TITLE         40   B  1  2
00030A            PAGE           5Y OB  1 60
00040A            TEXT        1760   B  2  1
00050A            LINNBR         3  OH
00060A            POSNBR         3  OH
     A
```

The application program sets the contents of LINNBR and POSNBR before issuing an output operation to RECORD1. When the record is displayed, the fields *Title*, *Page*, and *Text* appear on the display. The cursor can be at some location in a field of RECORD1 where the workstation user is to make changes.

DATE (Date) keyword for display files

You use this field-level keyword to display the current date as a constant (output-only) field.

You can specify the location of the field, the DATE keyword, and, optionally, EDTCDE , EDTWRD, COLOR, DSPATR, or TEXT keywords. Positions 17 through 38 must be blank.

The format of the keyword is:

```
DATE([*JOB|*SYS] [*Y|*YY])
```

The *JOB value causes the current job date to be displayed. If you do not specify a parameter, the parameter is set to *JOB as default. The *SYS parameter displays the current system date.

If you specify *Y, 2 digits are used to represent the year in the date format that the job attribute DATFMT designates. If you specify *YY, 4 digits are used to represent the year in the date format that the job attribute DATFMT designates. If you do not specify a parameter, the parameter is set to *Y as default.

The W edit code on the EDTCDE keyword returns a correctly formatted date only if a four digit year (*YY) is requested, and the job attribute DATFMT is YMD.

If you specify EDTCDE(Y) on a field with the DATE keyword, separators are added according to the date format that the DATFMT job attribute designates. For example, using EDTCDE(Y) when the DATFMT job attribute specifies *MDY changes the date from

```
mmddyy
```

to

```
mm/dd/yy.
```

The date separator is retrieved from the job attribute DATSEP at run time, and the job attribute DATFMT determines the order of the month, day, and year. (DATFMT can be MDY, DMY, YMD, or JUL, where M=month, D=day, Y=year, and JUL=Julian. DATSEP can be a slash (/), dash (-), period (.), or comma (,).)

The field length is dependent on the following conditions:

1. The format that the DATFMT job attribute specifies.
2. Whether the date field includes separators. The EDTCDE(Y) keyword controls separators.
3. The number of digits used to represent the year. The *Y and *YY parameters on the DATE keyword control the number of year digits.

Option indicators are not valid for this keyword, although option indicators can be used to condition the field on which it is specified.

Example

The following example shows how to specify the DATE keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A  20                           1 56DATE(*SYS)
00020A  21                           1 56DATE(*Y) EDTCDE(Y)
     A
```

Option indicator 20 causes the system date to be displayed without editing if it is on. Dates with 2 digit years are displayed with editing if option indicator 20 is off and option indicator 21 is on.

DATFMT (Date Format) keyword for display files

You use this field-level keyword to specify the format of a date field. This keyword is valid only for date fields (data type L).

The format of the keyword is:

```
DATFMT(date-format)
```

The date-format parameter specifies date formats. The following table describes the valid date formats and their default separator values.

Format name	Date-format parameter	Date format and separator	Field length	Example
Job default	*JOB			
Month/Day/Year	*MDY	mm/dd/yy	8	06/21/90
Day/Month/Year	*DMY	dd/mm/yy	8	21/06/90
Year/Month/Day	*YMD	yy/mm/dd	8	90/06/21
Julian	*JUL	yy/ddd	6	90/172
International Standards Organization	*ISO	yyyy-mm-dd	10	1990-06-21
IBM USA Standard	*USA	mm/dd/yyyy	10	06/21/1990
IBM European Standard	*EUR	dd.mm.yyyy	10	21.06.1990
Japanese Industrial Standard Christian Era	*JIS	yyyy-mm-dd	10	1990-06-21

If you do not specify the DATFMT keyword, the default is *ISO.

If you specify *JOB, the high level language and the application handle the format as *ISO. On output, the system converts the format to the format specified by the Date Format Job Definition Attribute. On input, the system converts the format to *ISO before it passes control to the application. There are always 10 spaces reserved on the display screen for a Date field with DATFMT(*JOB), even though 8 characters in the case of *MDY, *DMY, and *YMD, or 6 characters in the case of *JUL are displayed.

The format of DFT, DFTVAL, and MAPVAL keyword values must match the format that the DATFMT keyword specifies. If the DATFMT keyword specifies *JOB or the DATFMT keyword is set to *ISO as default, these values must have a format of *ISO.

If you specify the *ISO, *USA, *EUR, or *JIS value, you cannot specify the DAT keyword. These date formats have fixed separators.

The DATFMT keyword overrides the job attribute for a date field. It does not change the system default.

It is the responsibility of the high-level language and the application to format the date field according to the format specified on the DATFMT keyword and use the separators specified on the DATSEP keyword. The system does not format fields on output. The system validates the date field (L data type) on input according to the format the DATFMT keyword specifies and the separator that the DATSEP keyword specifies.

Option indicators are not valid for this keyword, although option indicators can be used to condition the field for which it is specified.

Example

The following example shows how to specify the DATFMT keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A
00020A          R RECORD
00030A            DATFLD1         L  B  5  2DATFMT(*JUL)
00040A            DATFLD2         L  B  5 22DATFMT(*EUR)
00050A            DATFLD3         L  B  5 42DATFMT(*JOB)
     A
```

If the date to be displayed is June 21, 1990, the date format defined in the Job Definition Attributes is *MDY and the date separator defined in the Job Definition Attributes is a slash (/), the following values are displayed when RECORD is written.

```
DATFLD1     90/172
DATFLD2     21.06.1990
DATFLD3     06/21/90
```

DATSEP (Date Separator) keyword for display files

You use this field-level keyword to specify the separator character for a date field. This keyword is valid only for date fields (data type L).

The format of the keyword is:

```
DATSEP(*JOB | 'date-separator')
```

The date separator parameter specifies the separator character that appears between the year, month, and day. Valid values are a slash (/), dash (–), period (.), comma (,) or blank ( ). Single quotation marks must enclose the parameter.

If you specify the *ISO, *USA, *EUR, or *JIS date format value for the DATFMT keyword, you should not specify the DATSEP keyword. These formats have fixed date separators.

If you do not specify the DATSEP keyword and the format that is specified for DATFMT does not have a fixed date separator, DATSEP is set to *JOB as the default.

If you specify *JOB or if DATSEP is set to *JOB as the default, the high-level language and the application handle the separator as a slash (/). On output, the system converts the separator that was specified by the Date Separator Job Definition Attribute. The system converts the separator to a slash (/), as soon as the system receives the separator, before passing control to the application.

The separator for DFT, DFTVAL, and MAPVAL keyword values must match the separator the DATSEP keyword specifies. If the DATSEP keyword specifies *JOB or the DATSEP keyword is set to *JOB as default, these values must have a format of a slash (/).

The DATSEP keyword overrides the job attribute. It does not change the system default.

It is the responsibility of the high-level language and the application to format the date field according to the format specified for the DATFMT keyword and use the separators specified for the DATSEP keyword. The system does format fields on output. The system validates the date field on input according to the format that the DATFMT keyword specifies and the separator that the DATSEP keyword specifies.

Option indicators are not valid for this keyword, although option indicators can be used to condition the field for which it is specified.

Example

The following example shows how to specify the DATSEP keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A
00020A          R RECORD1
00030A            DATFLD2         L  B  5  2DATFMT(*DMY) DATSEP('-')
00040A            DATFLD4         L  B  5 22DATFMT(*JUL) DATSEP(' ')
00050A            DATFLD6         L  B  5 42DATFMT(*JOB) DATSEP(*JOB)
     A
```

If you want to display the date June 21, 1990, the date format defined in the Job Definition Attributes is *MDY and the date separator defined in the Job Definition Attributes is a slash (/), the following values will be displayed when RECORD1 is written.

```
DATFLD2    21-06-90
DATFLD4    90 172
DATFLD6    06/21/90
```

DFT (Default) keyword for display files

You use this field-level keyword to specify the constant value for constant fields (unnamed fields) and to specify a default value for named fields.

The format of the keyword is:

```
DFT('value') | 'value'
```

The maximum number of characters you can specify in the literal is set by the size of the display on which the field is to be displayed as follows:

Size of display
Maximum characters

24 x 80
1919

27 x 132
3563

Constant fields

The value of a constant field can be specified as a value enclosed by single quotation marks. (For other ways to specify a constant field, see the DATE, MSGCON, and TIME keywords.) You can omit the DFT keyword itself, as well as the parentheses, to simplify the DDS. Whether you specify the DFT keyword explicitly or implicitly, the IBM i operating system displays the specified value as a constant field on the display. See “Name for display files (positions 19 through 28)” on page 7 for a description of constant fields.

Named fields

For input-only fields, the specified value is displayed each time the field is displayed. The displayed value can then be changed by the workstation user and returned to your program.

For output-only and input/output fields, you must also specify PUTOVR at the record level and OVRDTA at the field level with the DFT keyword. The specified value is displayed only on the first output operation. On subsequent output operations, the program value is displayed.

The DFTVAL, EDTCDE, and EDTWRD keywords cannot be specified with the DFT keyword.

The DFT keyword is not valid on floating point fields.

Option indicators are not valid for this keyword, although option indicators can be used to condition the field (whether constant or named) on which it is specified.

Example 1

The following example shows how to specify the DFT keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00030A            HOTTYP         1   I  7  9DFT('D')
00040A                                     VALUES('D' 'S')
00050A                                  8  9'ON'
00060A        01                       12  1'HOTEL NAME: 'TERRACE INN'
00070A                                     TEXT('Constant field is +
00080A                                     conditioned, not the implicit +
00090A                                     DFT keyword')
00100A        02                       12  1'HOTEL NAME: 'RIVER VIEW INN'
00110A                                     TEXT('Either 'TERRACE INN' or +
00120A                                         'RIVER VIEW INN' could +
00130A                                         appear in line 12, position 1')
     A
```

The constant field *ON*, having no option indicators, is always displayed.

If indicator 01 is on, the following information is displayed:

```
HOTEL NAME:  'TERRACE INN'
```

If indicator 02 is on and indicator 01 is off, the following information is displayed:

```
HOTEL NAME:  'RIVER VIEW INN'
```

Example 2

If you are specifying a constant field for more than one display size, and you are changing the location of the field but not the contents of the field for the different display sizes, then do not repeat the value. The following example shows how to do this.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A                                     DSPSIZ(*DS3 *DS4)
     A                                     :
     A                                     :
00080A                                  22  2'Constant data'
00090A                                  26  2
     A
```

The constant field *Constant data* appears on line 22, position 2, on the 24 x 80 display screen, and it appears on line 26, position 2, on the 27 x 132 display.

Related reference

PUTOVR (Put with Explicit Override) keyword for display files

You use this record-level keyword to enable the override of either display attributes or data contents (or both) of specific fields within a record that is displayed on a workstation device. By using the PUTOVR keyword, you can reduce the amount of data sent to the display device.

DFTVAL (Default Value) keyword for display files

You use this field-level keyword to specify a default value for an output-capable field.

On the first output operation, the specified value is displayed if the option indicator is on or has not been specified. Otherwise, the program value is used. On subsequent output operations, the program value appears.

The format of the keyword is:

```
DFTVAL('value')
```

This keyword is valid on output-only (O) or output-input (B) fields.

You can only use this keyword to initialize named fields. It is not allowed on constant fields.

Since the maximum number of characters on a DDS statement is 5000, this keyword, along with any other keywords specified on the DDS statement, must contain less than 5000 characters.

You cannot use this keyword in a subfile format (SFL keyword).

You cannot specify the DFTVAL keyword on the same field with a DFT, EDTCDE (Edit Code), or EDTWRD (Edit Word) keyword, or on a floating-point field.

Option indicators are valid for this keyword.

Example

The following example shows how to specify the DFTVAL keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A          R RECORD1
     A  50                                 PUTOVR
     A            FIELD1         3A  B 12 01DFTVAL('AAA') OVRDTA
     A            FIELD2         3D 0O 12 05OVRDTA
     A  10                                 DFTVAL('000')
     A            FIELD3         3D 0O 12 09DFTVAL('000') OVRDTA
     A
```

In this example, before displaying the record, the application program assigns ZZZ to FIELD1, 999 to FIELD2, and 456 to FIELD3. On the first output operation, AAA 000 000 displays if indicator 10 is on; AAA 999 000 displays if indicator 10 is off.

The workstation user types XXX into FIELD1. On the second output operation, XXX 999 456 displays if indicator 50 is on; AAA 000 000 displays if indicator 50 is off and indicator 10 was on during the first output operation. AAA 999 000 displays if indicator 50 is off and indicator 10 was off during the first output operation.

DLTCHK (Delete Check) keyword for display files

You use this field-level keyword to specify that the IBM i operating system is to ignore all validity checking and CHKMSGID keywords that are specified for a referenced field.

This keyword is valid only when R is specified in position 29.

This keyword has no parameters.

If you specify *any* new validity checking keywords, DLTCHK is unnecessary. The new validity checking keywords override the referenced validity checking keywords.

Option indicators are not valid for this keyword.

Example

The following example shows how to specify the DLTCHK keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00030A                                 REF(FILE)
00040A          R RECORD
00050A            CODE      R        I  3 20DLTCHK
     A
```

DLTEDT (Delete Edit) keyword for display files

You use this field-level keyword to specify that the IBM i operating system is to ignore the EDTCDE or EDTWRD keyword if either of them is specified for a referenced field. This keyword is valid only when you specify R in position 29.

This keyword has no parameters.

If you specify a new editing keyword, DLTEDT is unnecessary. The new editing keyword overrides the referenced editing keyword.

Option indicators are not valid for this keyword.

Example

The following example shows how to specify the DLTEDT keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00040A                                  REF(FILEA)
00050A          R RECORD
00060A            AMT       R        B  5 20DLTEDT
     A
```

DSPATR (Display Attribute) keyword for display files

You use this field-level keyword to specify one or more display attributes for the field that you are defining.

You can specify the DSPATR keyword more than once for the same field, and you can specify more than one attribute for the same keyword. However, each attribute (for example, UL), can be specified only once per field.

Note: The effects of attributes might not appear on the display, depending on the hardware or software emulator that you are using.

The format for the keyword is

```
DSPATR(attribute-1 [attribute-2 [attribute-3 [...]]])
or
DSPATR(&program-to-system-field);
```

If you specify more than one attribute for the same field, whether in one keyword or in separate keywords, each attribute that is specified (and in effect when the field is displayed) affects the field. For example, if you want a field to be displayed with its image reversed and with high intensity, specify either DSPATR (RI HI), or DSPATR(RI), and DSPATR(HI).

The program-to-system-field parameter is required and specifies that the named field must be defined in the record format, alphanumeric (A in position 35), length of one, and usage P (P in position 38). The program uses this P-field to set the display attribute for the field this DSPATR keyword applies to.

The name P-field is used for multiple fields with the record being defined. One DSPATR P-field is allowed per field. The P-field contains the display attribute and identifies whether the field should be protected. See Valid P-field values.

Valid attributes for the first format of the DSPATR keyword

The following list shows valid attributes for the first format of the DSPATR keyword:

- **For all fields**

  Display attribute
  Meaning

  BL
  Blinking field

  CS
  Column separator

  HI
  High intensity

  ND
  Nondisplay

PC
Position cursor

RI
Reverse image

UL
Underline

- **For Input-Capable fields only**

Display attribute
Meaning

MDT
Set changed data tag when displayed

OID
Operator identification

PR
Protect contents of field from input typing

SP
Select by light pen

Notes:

1. If you specify the UL, HI, and RI attributes on the 5250 display station for the same field, the result is the same as if you had specified ND.
2. If OID is specified, then SP should not be specified. Neither OID nor SP can be optioned unless specified with another display attribute.
3. Display attributes BL, CS, HI, RI, and UL can also be specified at the file, record, or field level as parameter values on the CHGINPDFT keyword.
4. Display attributes CS, HI, and BL can cause fields on the 5292, 3477 Model FC, 3487 Model HC, 3179, 3197 Model C1 and C2, and 3488[5] color display stations to appear as color fields.
5. If you are using an IBM Personal System/2 (PS/2) computer that is emulating a 5250 display station and you are directly changing the EBCDIC screen buffer, you need to set the MDT attribute. See the *IBM Personal Computer Enhanced 5250 Emulation Program Technical Reference* manual for additional information.
6. If you are using a PS/2 computer and VGA monitor, the UL attribute does not work due to hardware specific limitations in the way buffers are used.

Option indicators are valid for this keyword, except when the attributes OID or SP are the only display attributes specified.

Detailed descriptions of each of the attributes follow the coding example and sample display are provided in the following figure.

[5] Dependent on the monitor attached to the display device.

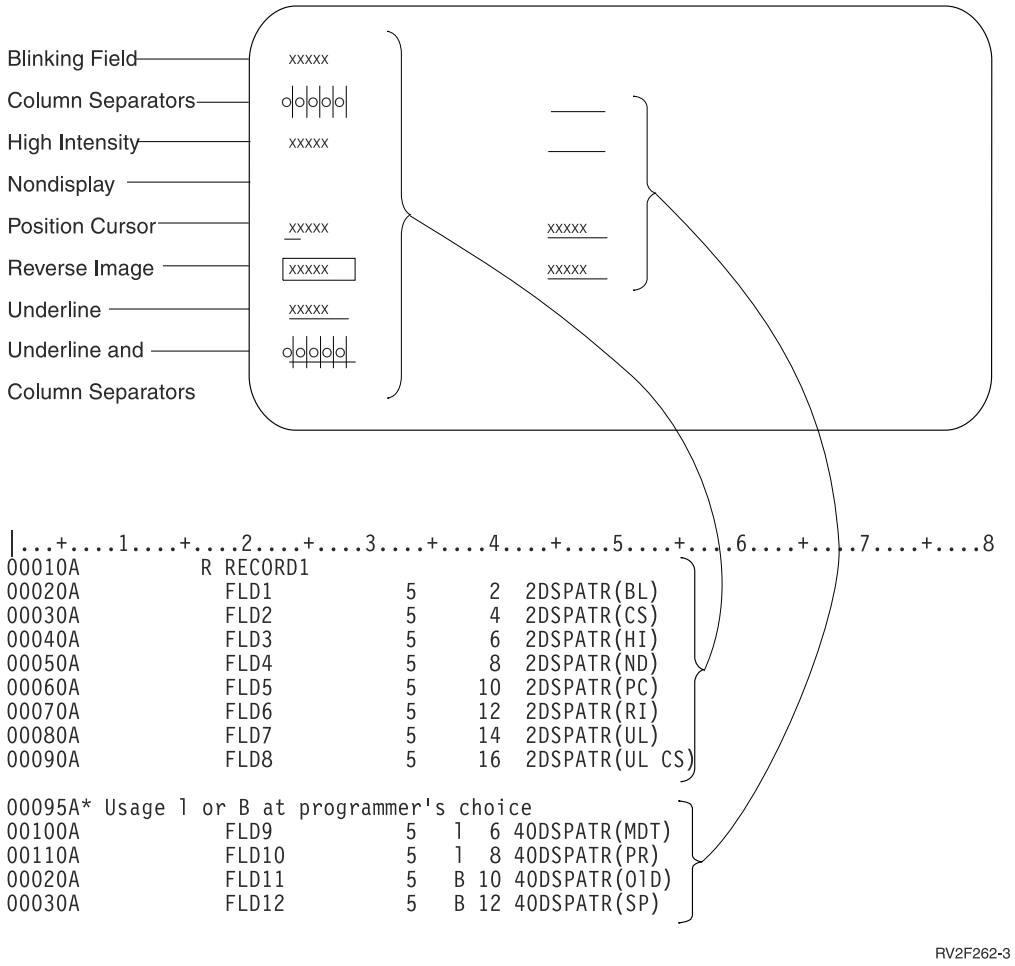


Figure 14. A 5-byte field displayed with various display attributes

Display attributes for all fields

The following list shows attributes for all fields:

BL (Blink)
Use this attribute to specify that the field is to blink when it is displayed.

CS (Column separator)
Use this attribute to specify that each position of the field is to be displayed with a vertical bar at its left and right edge. When specified for a nondisplay field, the separators are displayed even though there are no characters between them. You can use column separators to precisely indicate cursor positioning within a field and to indicate the length of an otherwise blank field.

HI (High intensity)
Use this attribute to specify that the field is to be intensified (highlighted) when it is displayed on the display.

ND (Nondisplay)
Use this attribute to specify that the field is not to be displayed; the display positions for this field appear blank. The attribute can be used for passwords or other security-sensitive data. If the print function (permitted by specifying the PRINT keyword) is performed, nondisplay fields are not printed.

PC (Position cursor)
Use this attribute to position the cursor to the first character position of the field you are defining. You can specify this attribute for several fields, and the cursor will be positioned at the first selected field with this attribute. Note that the fields within a record are ordered in line/position sequence as they appear on the display and not necessarily in the order you specify them.

RI (Reverse image)
Use this attribute to specify that the image of the field is to be reversed from the other portion of the screen when it is displayed. Whether the screen is light-on-dark or dark-on-light depends on the status of the display before the field is displayed. This setting is controlled by the workstation user.

UL (Underline)
Use this attribute to specify that the field is to be underlined when it is displayed. All input-capable fields are underlined by default. Use the CHGINPDFT keyword to prevent the default underlining. (If CHGINPDFT is specified, DSPATR(UL) must be specified to underline an input-capable field.) If DSPATR(UL) is specified with option indicators and the option indicators are not satisfied (DSPATR(UL) is not selected), the field appears without underline.

Display attributes for input-capable fields

The following list shows attributes for input-capable fields:

MDT (Set changed data tag)
Use this attribute to specify that the IBM i operating system is to set on the changed data tag (MDT) for the field you are defining when the field is written to the display. The attribute ensures that the field is sent from the device when the record is read from the display.

Note: The IBM i program saves output data for input/output fields or initialized data for fields with the DFT keyword specified. This causes the saved data to be returned on an input operation if no new (changed) data is entered into the field.

OID (Operator Identification)

Use this attribute to specify that the IBM i operating system is to allow magnetic stripe reader OID data to be entered into this field. If it is to be a nondisplay field also, the DSPATR(ND) attribute must be specified.

A field with the DSPATR(OID) keyword functions like any other input-capable field; data can be entered from either the keyboard or the magnetic stripe reader. The DSPATR(OID) keyword can be specified (but is not required) to indicate that data can be entered using a magnetic stripe reader. You can type into the field unless the keyboard shift Inhibit Keyboard Entry (I) is specified. If both DSPATR(OID) and DSPATR(SP) are specified on the same field, DSPATR(SP) is ignored.

PR (Protect)
Use this attribute to specify that the workstation user cannot type into the input-capable field that you are defining. This attribute is valid for input-capable fields only. Output-only fields and constant fields are protected by definition.

SP (Select by light pen)

Use this attribute to specify that this input-capable field can be selected by a light pen. The workstation user can type in a light pen field unless an I (Inhibit Keyboard Entry) has been specified in position 35 (Data Type/Keyboard Shift) for the field.

When the field is first displayed, the contents of the field are set by your program (input/output field) or in the DDS (input-only field with DFT keyword or character string). If no new data is typed in by the workstation user, this output data is returned to your program on an input operation.

A field that can be selected by a light pen should be at least 3 bytes long. The recommended contents of this field are:

- A switch character, either hex 6F (?) or, if the workstation user selects the field by a light pen, hex 6E (>)
- A blank (hex 40)
- A target character, which can be any character, such as an asterisk (*)
- Another blank
- Additional data to identify the field to the workstation user (1 or more characters)

This attribute is useful only for workstations with a light pen feature for selecting.

Valid P-field values

The DSPATR P-field does not support the following display attributes:

Display attribute
Meaning

MDT
Set changed data tag when displayed

OID
Operator identification

PC
Position cursor

SP
Select by light pen

Valid P-field values (nonprotect)

Hex	Limited color	Full color
20	Normal	Green
21	Reverse image	Green, reverse image
22	High intensity	White
23	High intensity, reverse image	White, reverse image
24	Underscore	Green, underscore
25	Underscore, reverse image	Green, underscore, reverse image
26	Underscore, high intensity	White, underscore
27	Nondisplay	Nondisplay
28	Blink	Red
29	Blink, reverse image	Red, reverse image
2A	Blink, high intensity	Red, high intensity
2B	Blink, high intensity, reverse image	Red, high intensity, reverse image
2C	Blink, underscore	Red, underscore
2D	Blink, underscore, reverse image	Red, underscore, reverse image
2E	Blink, underscore, high intensity	Red, underscore, blink
2F	Nondisplay	Nondisplay
30	Column separator	Turquoise, column separator
31	Reverse image, column separator	Turquoise, column separator, reverse image
32	High intensity, column separator	Yellow, column separator
33	High intensity, reverse image, column separator	White, reverse image, column separator
34	Underscore, column separator	Turquoise, underscore, column separator
35	Underscore, reverse image, column separator	Turquoise, underscore, reverse image, column separator

Hex	Limited color	Full color
36	Underscore, high intensity, column separator	Yellow, underscore, column separator
37	Nondisplay	Nondisplay
38	Blink, column separator	Pink
39	Blink, reverse image, column separator	Pink, reverse image
3A	Blink, high intensity, column separator	Blue
3B	Blink, high intensity, reverse image, column separator	Blue, reverse image
3C	Blink, underscore, column separator	Pink, underscore
3D	Blink, underscore, reverse image, column separator	Pink, underscore, reverse image
3E	Blink, underscore, high intensity, column separator	Blue, underscore
3F	Nondisplay	Nondisplay

Valid P-field values (protect)

Table 5. P-field values (protect)

Hex	Limited color	Full color
A0	Normal	Green
A1	Reverse image	Green, reverse image
A2	High intensity	White
A3	High intensity, reverse image	White, reverse image
A4	Underscore	Green, underscore
A5	Underscore, reverse image	Green, underscore, reverse image
A6	Underscore, high intensity	White, underscore
A7	Nondisplay	Nondisplay
A8	Blink	Red
A9	Blink, reverse image	Red, reverse image
AA	Blink, high intensity	Red, high intensity
AB	Blink, high intensity, reverse image	Red, high intensity, reverse image
AC	Blink, underscore	Red, underscore
AD	Blink, underscore, reverse image	Red, underscore, reverse image
AE	Blink, underscore, high intensity	Red, underscore, blink
AF	Nondisplay	Nondisplay
B0	Column separator	Turquoise, column separator

Table 5. P-field values (protect) (continued)

Hex	Limited color	Full color
B1	Reverse image, column separator	Turquoise, column separator, reverse image
B2	High intensity, column separator	Yellow, column separator
B3	High intensity, reverse image, column separator	White, reverse image, column separator
B4	Underscore, column separator	Turquoise, underscore, column separator
B5	Underscore, reverse image, column separator	Turquoise, underscore, reverse image, column separator
B6	Underscore, high intensity, column separator	Yellow, underscore, column separator
B7	Nondisplay	Nondisplay
B8	Blink, column separator	Pink
B9	Blink, reverse image, column separator	Pink, reverse image
BA	Blink, high intensity, column separator	Blue
BB	Blink, high intensity, reverse image, column separator	Blue, reverse image
BC	Blink, underscore, column separator	Pink, underscore
BD	Blink, underscore, reverse image, column separator	Pink, underscore, reverse image
BE	Blink, underscore, high intensity, column separator	Blue, underscore
BF	Nondisplay	Nondisplay

Example 1

The following example shows how to specify the DSPATR(SP) keyword with an input-only field (showing the recommended data contents as a character string).

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00110A            SPFLD         50I  I  5  4'? * OPTION 1'
00120A                                      DSPATR(SP)
     A
```

No data can be typed into field SPFLD. When the field is selected with the light pen, the data returned in field SPFLD will be: >_*_OPTION_1, where _ represents a blank.

Example 2

The following example shows that when the workstation user selects a field with the light pen, both the MDT bit and the first character of that field are changed. When the field is selected, the MDT bit is set on, changing the first character of the field to >. If the same field is selected again, the MDT bit is set off and the first character becomes ?.

By specifying a switch character, your program prevents the first character of data from being changed to > or ? when the field is selected by the light pen. If the MDT bit is on when your program sends an input

operation to the record format, the contents of the field are returned to your program as a user-changed field.

If you use DSPATR(MDT) to set on the MDT of a field that can be selected by the light pen, then you should either omit the MDTOFF keyword from other record formats, or read that field before displaying any record format with MDTOFF in effect.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A          R LIGHTPEN
00020A            FLD1          10   I  5  2'> * $12.50'
00030A                                      DSPATR(SP MDT)
00040A*
00050A          R RCD2                      OVERLAY MDTOFF
00060A            FLD1          10   B 11  2
     A
```

If the program displays LIGHTPEN, then displays RCD2, then reads LIGHTPEN, and the workstation user does not select FLD1 with the light pen, the MDT of FLD1 is turned off by the display of RCD2.

Also, the switch character of FLD1 is returned as ?, even though the field was not selected, and the switch character appears as >. The MDT and the switch character are in opposing states.

Example 3

The following example shows how to specify the DSPATR keyword with P-field usage:

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A          R RECORD
     A            FLD1           5A     2  6DSPATR(&PFLD1)
     A            FLD2           5A     2  6DSPATR(&PFLD2)
     A            PFLD1          1A  P
     A            PFLD2          1A  P
     A
```

Related reference

COLOR (Color) keyword for display files
You use this keyword to specify the color of a field on a color display.

DSPMOD (Display Mode) keyword for display files

You use this record-level keyword to specify which mode (display size) you want to use for the 3180, the 3477, or the 3197 Model D1, D2, W1, or W2 display station.

*DS3 (24 x 80) and *DS4 (27 x 132) are both supported for the 3180, 3477, 3487 Models HA, HC, HG, and HW, 3488, and the 3197 Model D1, D2, W1, or W2 display stations.

The format of the keyword is:

```
DSPMOD(condition-name)
```

This keyword is valid only when both the 24 x 80 and 27 x 132 display sizes are specified on the DSPSIZ keyword. The first of the two display sizes specified on the DSPSIZ keyword is the default display mode. The record is displayed using this mode unless the DSPMOD keyword indicates that the second specified display size should be used.

Note: This keyword is a runtime keyword and not a compile-time keyword.

You can specify the default display size with this keyword only if you do not specify option indicators for this keyword.

The capability to display in the 27 x 132 mode is allowed only on a 3180-2 or a 3197 Model D1, D2, W1, or W2 device attached locally to a 6040 or 6140 controller or remotely to a 5294 or 5394 controller. The DSPMOD keyword is ignored unless these controllers are used.

When a record with DSPMOD causes the mode to be changed, all records currently on the display are deleted. The record with DSPMOD active is then sent to the display. The mode for this record is maintained on the display as long as the DSPMOD keyword is active. Setting DSPMOD off or writing to

another record without DSPMOD causes the display mode to be placed back in the primary display size for the device.

The following keywords are ignored if the display modes have changed:

ALWROL	OVERLAY
ASSUME	PROTECT
CLRL	PUTOVR
ERASEINP/INZINP	PUTRETAIN
ERRMSG	SFLMSG
ERRMSGID	SFLMSGID
KEEP	

When you create a file specifying any of the above keywords and DSPMOD on the same record, a warning message results at create-time. However, a diagnostic message is *not* issued during processing.

This keyword is not valid for user-defined records (USRDFN keyword).

The DSPMOD keyword cannot be specified on a subfile record (SFL keyword). The subfile is be displayed according to the DSPMOD of the corresponding subfile control record.

Option indicators are valid for this keyword. If the option indicator is on at the time of processing, the display mode you have chosen will be used to display the record. However, if the option indicator is off at the time of processing, the default display mode will be used.

Note: Switching display modes is similar to displaying a record without OVERLAY.

Example 1

The following example shows how to specify the DSPMOD keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A                                      DSPSIZ(*DS3 *DS4)
     A          R RECORD1                   DSPMOD(*DSP4)
     A          R RECORD2
     A          R RECORD3
     A  03                                  DSPMOD(*DS4)
     A
```

The DSPMOD keyword gives the following results:

- If you write RECORD1, RECORD1 is displayed in *DS4 mode.
- If you write RECORD2, the display is cleared and RECORD2 is displayed in *DS3 mode.
- If you write RECORD3 with indicator 03 off, RECORD3 is displayed in *DS3 mode. RECORD2 remains on the display.
- If you write RECORD3 with indicator 03 on, the display is cleared and RECORD3 is displayed in *DS4 mode.

Example 2

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A                                      DSPSIZ(24 80 *NORM +
     A                                             27 132 *WIDE)
     A          R RECORD1
     A  03                                  DSPMOD(*WIDE)
     A
```

DSPRL (Display Right to Left) keyword for display files

You use this file-level keyword to specify that the records in the display file are written right to left on the display.

This keyword has no parameters.

Option indicators are not valid for this keyword.

This keyword can only be used on a bidirectional device.

Example

The following example shows how to specify the DSPRL keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A                                      DSPRL
     A          R RECORD
     A            FIELD1       20A      5  5')emaN remotsuC('
     A
```

DSPSIZ (Display Size) keyword for display files

You use this file-level keyword to specify the display size to which your program can open the display file.

The formats of the keyword are:

```
DSPSIZ(*DSw [*DSx])
DSPSIZ(lines positions[condition-name-1][lines positions[condition-name-2]])
```

The DSPSIZ keyword is optional. If you do not specify it for a display file, the display file can be opened only to display devices with a 24 x 80 display size. You can specify this keyword in one of two ways:

- Using IBM-supplied display size condition names. Specify up to two parameter values as *DS3 or *DS4 in any order. At least one parameter value is required. You cannot specify a parameter value twice.
- Specifying lines and positions to permit user-defined display size condition names. Instead of the IBM-supplied display size condition names, specify the display size in lines and positions (only 24 x 80, and 27 x 132 are valid). (See examples 1, 2, and 3, in this topic.)

  Optionally, you can also define a display size condition name other than *DS3 or *DS4. The display size condition name you define must be from 2 to 8 characters long, and the first character must be an asterisk (*). You can specify these user-defined condition names in positions 7 through 16 (conditioning) on subsequent DDS statements at the field level. (See example 2, in this topic.) If you do not specify user-defined display size condition names, you must use IBM-supplied display size condition names to condition the location of fields.

  If you specify more than one parameter value, see Primary and secondary display sizes.

Option indicators are not valid for this keyword.

Primary and secondary display sizes

Whether you use IBM-supplied display-size condition names or specify lines and positions directly, the first display size you specify is the *primary display size*. The second display size, if specified, is the *secondary display size*. Figure 15 on page 92 shows an example of primary and secondary display size specification. This figure shows the keyword specified as DSPSIZ(24 80 27 132). The primary display size is 24 by 80; the secondary display size is 27 by 132.

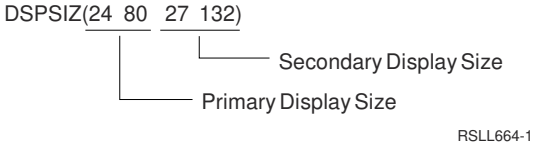


Figure 15. Using DSPSIZ to specify primary and secondary display sizes

When you specify more than one display size for DSPSIZ, you can specify display size condition names in positions 7 through 16 on subsequent DDS statements at the record and field levels. These display size condition names are then used to condition keywords and the locations of fields. When both a primary and secondary display are specified, the display file will be validated for both sizes.

Note: If you specify user-defined display size condition names for DSPSIZ, you cannot use IBM-supplied display size condition names for conditioning.

The capability to display in the 27 x 132 mode is allowed only on a 3180-2, or a 3197 Model D1, D2, W1, or W2 device attached locally to a 6040 or 6041 controller or remotely to a 5294 or 5394 controller. The display size for the 27 x 132 mode will be ignored for DSPSIZ unless these controllers are used.

The following table shows the valid display sizes.

Display sizes	Display device	Meaning
*DS3 or 24 x 80	3179 3180 3196 3197 3476 3486 3487 (Models HA, HC, HG, and HW) 3488 5251 (Models 11 and 12) 5291 5292	24 lines x 80 positions 1920 positions total
*DS4 or 27 x 132	3180 3197 (Models D1, D2, W1, and W2) 3477 (Models FA, FC, FD, and FG) 3487 (Models HA, HC, HG, and HW) 3488 (Use 6040 or 6041 controller locally, or 5294 or 5394 controller remotely for 27 x 132 display capability.)	27 lines x 132 positions 3564 positions total

The display size designated as the primary display size should be the one with which the display file will most often be used. Additional processing is performed when the actual display size is the secondary display size.

The display size condition names let you improve the use of a single display file for any size display. For example, when you are using subfiles, you can specify 24 records per page for a 27 x 132 display and 22 records per page for a 24 x 80 display.

Special cases you might encounter when specifying DSPSIZ

You might encounter the following special cases when specifying DSPSIZ:

- DSPSIZ(*DS3 *DS4). All field locations for display size *DS4 are the same as for display size *DS3.
- All fields of a record can be defined such that none fit on the display size to which the file is opened. In this case, no fields are displayed. The record is handled as it does for a larger display where the fields fit. The record remains active until it is deleted or overlaid. Active records can be read by your program. The input request is sent to the display device, and the workstation user must respond to satisfy the request.
- All fields of a subfile record must fit within the specified subfile page, and the complete page must always fit (vertically) on the display size on which it is displayed at processing time. Specify valid display sizes by conditioning the SFLPAG (Subfile Page) keyword with display size condition names.
- The following records occupy no display space:
  - Records with no fields defined (this is different from none selected)
  - Records with only hidden, message, or program-to-system fields
  - Records that have the CLRL keyword specified and that have no input-capable fields. (These records can remain on the display, but are not recognized by the IBM i operating system for input operations, or they can be cleared through the use of the ERASE keyword.)

  For implementation and programming purposes, these records are assumed to be located at 00 (from line 0 to line 0). On an output operation, any record located at 00 overlays a record at that location. When an overlap occurs, the previous record is disregarded and no longer considered active. The new record at location 00 is active and can be read by your program.

- If two fields in a record format have the same display location (line/position), they are treated as overlapping fields. Overlapping fields are not displayed at operation time. The IBM i operating system checks each field as it is processed to ensure that it does not overlap a previously processed field.

  If a field does overlap, it is treated as an optioned field and not selected. To allow this processing-time checking, data description specifications must ensure all fields within a record are in primary location sequence, even when condition names are specified. For example, assume only one input field is specified for a record format and, according to the field location specification, this field overlaps a preceding output field. The workstation user cannot enter any data because the input field is never displayed.

Note: The primary location sequence as it is seen in the display file must not be changed by specifying a different location sequence for the secondary display size. (A severe error occurs and the file is not created.)

Example 1

The following example shows how to specify primary and secondary display sizes using the DSPSIZ keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A*
00020A*                                          1  2
00030A                                     DSPSIZ(27 132 24 80)
00040A          R RECORDA
00050A            FIELDA        10  0   1  2
00060A            FIELDB        10  0   1 81
00070A            FIELDC        10  0  25  1
     A
```

In this example, the primary display size **1** is 27 x 132 and the secondary display size **2** is 24 x 80. FIELDB is beyond position 80 and FIELDC is beyond line 24, so the data description processor gives them a location of *NOLOC in the expanded source printout for secondary display size 24 x 80.

If the data description processor assigns *NOLOC to an input-capable field, that field is processed at run time to the point of setting up the input buffer data to be returned in the user's input buffer. The field itself is not displayed. The workstation user cannot enter into or change these fields. No processing of any kind is done for output-only fields.

Figure 16 on page 95 shows a compiler listing for the above example.

```
5728SD1 R01 M00 880311              DATA DESCRIPTION              OGPL/DSPSIZ1          01/28/88 14:16:46       PAGE    1
FILE NAME . . .
```

Example 3

The following example shows how to reposition a field when the file is opened to different display sizes.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A                                  DSPSIZ(24 80 27 132)
00020A          R RECORDA
00030A            FIELD1        10  0  23  2
00040A
00050A
00060A  *DS4                          26  2
     A
```

In this example, FIELD1 has valid locations on both display sizes. It appears on the next to the last line on each display size.

Example 4

The following example shows that if you do not specify a display size condition name, the display location of a field can still be display size dependent as the result of the *plus* feature of DDS.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A                                  DSPSIZ(*DS4 *DS3)
00020A          R RECORD1
00030A            FIELD1        21      2 70
00040A            FIELD2        10       +10
     A
```

In this example, a line and a position for each field is calculated for each display size specified on the DSPSIZ keyword. If the plus value extends the field location beyond position 80, the field location is dependent on the display size. Figure 17 on page 97 is a compiler listing for the above example.

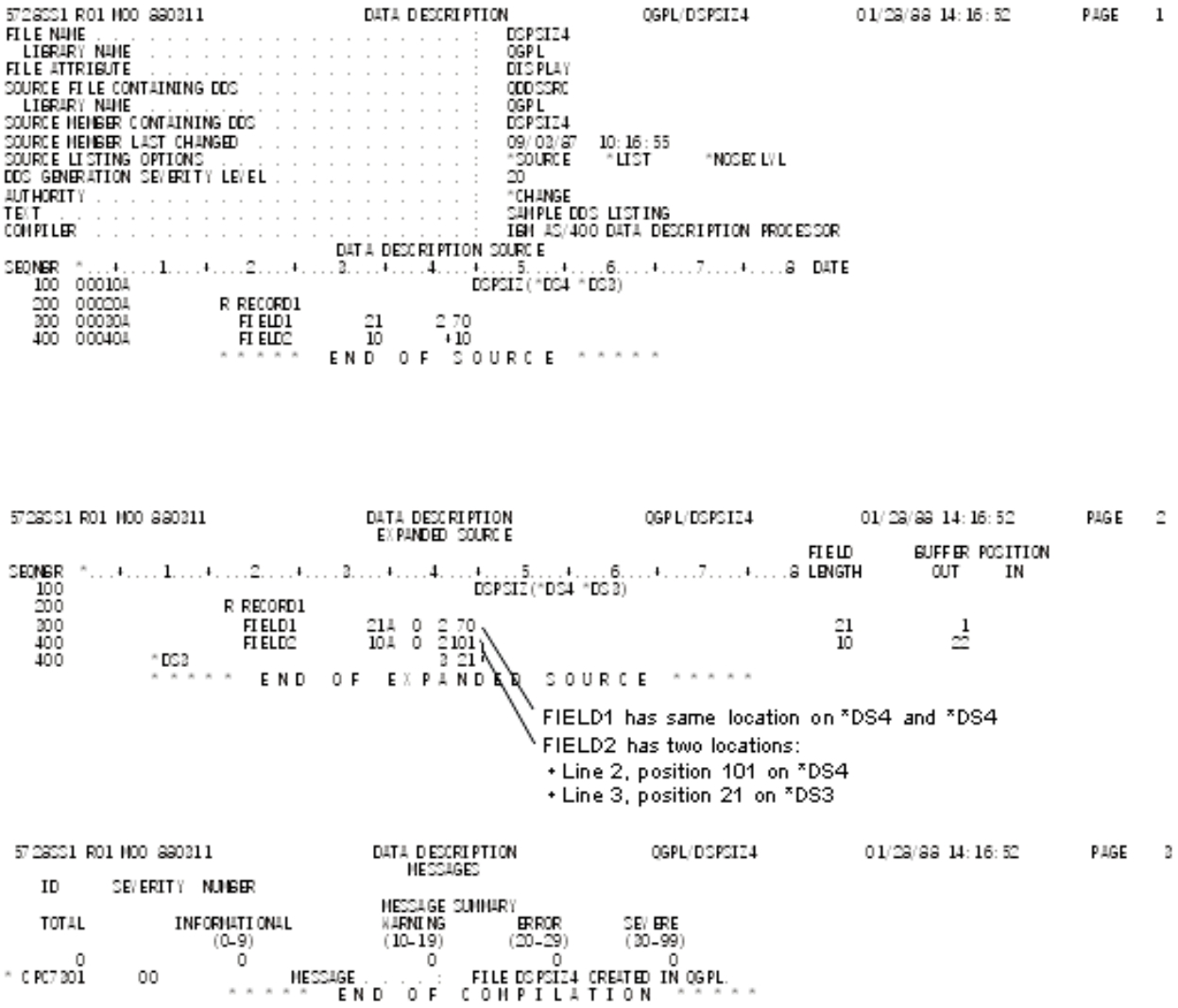


Figure 17. Compiler list

Related concepts

Location for display files (positions 39 through 44)
You use these positions to specify the exact location on the display where each field begins.

Condition for display files (positions 7 through 16)
Positions 7 through 16 are a multiple-field area in which you can specify option indicators.

DUP (Duplication) keyword for display files

You use this field-level keyword to activate the Dup key on the display station keyboard.

Press the Dup key when the cursor is in this input-capable field. This indicates that data for this field is to be duplicated from the record sent in the previous input operation. The actual duplication is the responsibility of your program.

See “System/36 environment considerations for display files” on page 247 for special considerations when specifying the DUP keyword in files that are used in the System/36 environment.

The format of the keyword is:

```
DUP[(response-indicator ['text'])]
```

You cannot specify the DUP keyword on a floating-point field (F in position 35).

You should specify a response indicator for a numeric field defined with this keyword. Hex 1Cs for numeric fields will not be returned to your buffer, but hex F0s will be returned for the remaining field positions.

Option indicators are valid for this keyword.

Restrictions on validity checking

Validity checking keywords (CHECK, COMP, RANGE, and VALUES) can be specified with the DUP keyword. However, they have no effect if the Dup key has been pressed.

If another field in the record format fails validity checking, the IBM i operating system tries to read the display again. The DUP response indicator is still returned to your program in the on condition, even if the workstation user does not type over the DUP characters or presses the Dup key again.

Example

The following example shows how to specify the DUP keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A          R RECORD1
00020A            FLDA           5   I  3  2
00030A  15                                  DUP(16 'FLDA was duped')
00040A            FLDAH          5  2H
     A
```

FLDA is an input-capable character and hexadecimal field for which the workstation user can press the Dup key. In your program, you must test the data in FLDA to ensure that it includes only the digits 0 through 9. In RPG III, you can use the TESTN operation code. In COBOL, you can use a numeric class test. In BASIC, you can specify an ON CONV statement earlier in the program than the statement with which you assign FLDA to FLDAH. In PL/I, you can specify an ON statement with the ERROR condition earlier in the program than the statement with which you assign FLDA to FLDAH.

Field FLDAH is a hidden numeric field with the same length as FLDA. When the application program reads RECORD1 and finds response indicator 16 set off, the program moves FLDA to FLDAH and uses FLDAH (which is a numeric field). When the application program reads RECORD1 and finds response indicator 16 set on, the Dup key was pressed and FLDA contains hex 1Cs. The program uses FLDAH without changing its contents (which are the contents from the previous input received from the display device).

Programming for the Dup key

The system follows this procedure of duplicating entire fields.

About this task

When you press the Dup key, the IBM i operating system handles the field as follows:

- If the field is a character field, the data displayed in the field is returned to your program as is. A hex 1C is placed at the cursor position and in the remaining field positions to the right. (Hex 1C appears as an over-scored asterisk on the display.) The response indicator, if specified, is set on.
- If the field is a numeric field and you specify a response indicator, a hex F0 is placed at the cursor position and in the remaining field positions. The response indicator is set on and returned to your program. If a response indicator is not specified, hex 1Cs are returned to your program.

In your program, you can duplicate entire fields (either character or numeric) with the following procedure:

Procedure

1. Specify two fields for each input-capable field on the display in DDS.

a) Specify one field as an input-capable field. For this field, specify DUP with a response indicator.

   You might want to specify DUP with an option indicator that is off on the first display of the field. This prevents the workstation user from using the Dup key when the field is first displayed.

b) Specify the other field as a hidden field (H in position 38).

2. On the first output operation, set off an option indicator for DUP.

   This prevents the workstation user from using the Dup key.

3. On the first input operation, move the input-capable field to the hidden field.

   This saves the typed value for later use.

4. On each subsequent output operation, set the option indicator on for the DUP keyword.

   This allows the workstation user to use the Dup key.

5. On each subsequent input operation, test the response indicator specified with DUP. If the response indicator is off, the input data should be moved to the hidden field. If the response indicator is on, you can use the existing value in the hidden field.

   Note: When using the DUP keyword in a subfile, an update operation should be performed after steps "3" on page 99 and "5" on page 99 to store the value of the hidden field into the subfile. This will be returned on the next read of that subfile record.

6. Repeat steps "4" on page 99 and "5" on page 99 for subsequent data entry using the Dup key.

Results

You can also duplicate character fields one character at a time by saving them in arrays, then moving the array one character at a time and checking for the DUP key indication of hex 1C.

You can achieve duplication of numeric fields one digit at a time by defining the field as character and eventually moving it to your numeric field after the hex 1Cs have been removed. You can test whether the Dup key has been pressed:

- For numeric fields, a response indicator is required.
- For character fields, a response indicator is optional.

The field will contain hex 1C at the cursor position and in the remaining positions if the Dup key has been pressed.

EDTCDE (Edit Code) keyword for display files

You use this field-level keyword to edit output-capable numeric fields.

The format of the keyword is:

```
EDTCDE(edit-code [* |floating-currency-symbol])
```

Editing includes the following changes to the appearance of displayed fields, depending on which edit code is specified:

- Leading zeros are suppressed.
- The field can be punctuated with commas and periods to show decimal position and to group digits by threes.
- Negative values can be displayed with a minus sign or CR to the right.
- Zero values can be displayed as zero or blanks.
- Asterisks can be displayed to the left of significant digits to provide asterisk protection.
- A currency symbol (corresponding to the system value QCURSYM) can be displayed immediately to the left of the significant digit that is farthest to the left (called *floating-currency symbol*). For fixed-currency symbols, use the EDTWRD keyword.
- The field can be further edited using a user-defined edit code.

EDTCDE covers most editing requirements. Use EDTWRD when the EDTCDE keyword is not sufficient.

You cannot specify both EDTCDE and EDTWRD for the same field. If a field previously defined in a database file has EDTCDE specified, you need not specify EDTCDE for that field in the display file. You can specify R in position 29 to refer to the previously defined field. The editing specified for the referenced field is included in the display file. However, if you also specify length, data type, or decimal positions for a display file field, editing specified for the referenced field is not included in the display file, and you must specify editing again in the display file.

The DFT and DFTVAL keywords cannot be specified with the EDTCDE keyword.

Option indicators are not valid for this keyword.

The rules for specifying edit codes and edit words are the same in all types of files. You can specify two kinds of edit codes: IBM i edit codes and user-defined edit codes.

The EDTCDE keyword is valid only for fields with Y or blank in position 35 (Data Type/Keyboard Shift). The use of this keyword changes the default used for position 35 to a Y.

IBM i edit codes

The IBM i edit codes are:

1 through 4
A through D
J through Q
W through Z

Note: The IBM i hardware operates with a preferred sign of F, which is equivalent to using edit code X. Edit code X sets the default of a blank keyboard shift (position 35) to numeric-only (attribute Y). The display length of the field is determined by the keyboard shift, not by edit code X (the default numeric-only Y attribute might add 1 position to the field for decimals). If the DATE or TIME keyword is specified with edit code X, the separator character is not displayed.

Optionally specifying asterisk fill or floating currency symbol

You can optionally specify asterisk fill or floating currency symbol with edit codes 1 through 4, A through D, and J through Q.

When you specify asterisk fill, an asterisk (*) is printed for each zero that is suppressed. A complete field of asterisks is printed for a zero balance field.

When you specify floating currency symbol, the symbol appears to the left of the first significant digit. The symbol does not print on a zero balance when an edit code is used that suppresses the zero balance. (The symbol that you specify must match the system value for the currency symbol (QCURSYM). The symbol must match when the file is created. It need not match when the file is used.)

Note: If an edit code is changed after a file is created, the editing specified at the time the file was created is used. The new edit code is not used unless the file is recreated.

The following table summarizes the functions provided by IBM i edit codes.

Table 6. Summary chart for IBM i edit codes

Edit codes	Commas[1] displayed	Decimal points[1] displayed	Sign displayed when negative value	Blank value of QDECFMT system value	I value of QDECFMT system value	J value of QDECFMT system value	Leading Zero suppressed
1	Yes	Yes	No sign	.00 or 0	,00 or 0	0,00 or 0	Yes
2	Yes	Yes	No sign	Blanks	Blanks	Blanks	Yes

Table 6. Summary chart for IBM i edit codes (continued)

Edit codes	Commas[1] displayed	Decimal points[1] displayed	Sign displayed when negative value	Blank value of QDECFMT system value	I value of QDECFMT system value	J value of QDECFMT system value	Leading Zero suppressed
3		Yes	No sign	.00 or 0	,00 or 0	0,00 or 0	Yes
4		Yes	No sign	Blanks	Blanks	Blanks	Yes
A	Yes	Yes	CR	.00 or 0	,00 or 0	0,00 or 0	Yes
B	Yes	Yes	CR	Blanks	Blanks	Blanks	Yes
C		Yes	CR	.00 or 0	,00 or 0	0,00 or 0	Yes
D		Yes	CR	Blanks	Blanks	Blanks	Yes
J	Yes	Yes	-(Minus)	.00 or 0	,00 or 0	0,00 or 0	Yes
K	Yes	Yes	-(Minus)	Blanks	Blanks	Blanks	Yes
L		Yes	-(Minus)	.00 or 0	,00 or 0	0,00 or 0	Yes
M		Yes	-(Minus)	Blanks	Blanks	Blanks	Yes
N	Yes	Yes	-(Minus)	.00 or 0	,00 or 0	0,00 or 0	Yes
O	Yes	Yes	-(Minus)	Blanks	Blanks	Blanks	Yes
P		Yes	-(Minus)	.00 or 0	,00 or 0	0,00 or 0	Yes
Q		Yes	-(Minus)	Blanks	Blanks	Blanks	Yes
W[2]							Yes
Y[3]							Yes
Z[4]							Yes

Notes:

1. The QDECFMT system value determines the decimal point character (period in U.S. usage), the character used to separate groups of three digits (comma in U.S. usage), and the type of zero suppression (depending on comma and period placement).
2. The W edit code suppresses the farthest left zero of a date field that is five digits long. It also suppresses the three farthest left zeros of a field that is six to eight digits long. The W edit code also inserts slashes (/) between the month, day, and year according to the following pattern:
   - nn/nnn
   - nnnn/nn
   - nnnn/nnn
   - nnnn/nn/nn
3. The Y edit code suppresses the farthest left zero of a date field that is three to six digits long or eight digits long. It also suppresses the two farthest left zeros of a field that is seven positions long. The Y edit code also inserts slashes (/) between the month, day, and year according to the following pattern:
   - nn/n
   - nn/nn
   - nn/nn/n
   - nn/nn/nn

- nnn/nn/nn
- nn/nn/nnnn

If the DATE keyword is specified with EDTCDE(Y), the separator character used is the job attribute, DATSEP at run time. The slash (/) is the default DATSEP.

4. The Z edit code removes the sign (plus and minus) from a numeric field. The sign of the units position is changed to a hexadecimal F before the field is written.

User-defined edit codes

Edit codes 5 through 9 are user-defined edit codes. A user-defined edit code can do more editing than an IBM i edit code. For example, you might need to edit numbers that include hyphens (such as telephone numbers) or more than one decimal point. You can use user-defined edit codes for these functions. These edit codes are named QEDIT5, QEDIT6, QEDIT7, QEDIT8, and QEDIT9, and can be referred to in DDS or a high-level language program by number (5, 6, 7, 8, or 9).

A user-defined edit code is an IBM i object and must exist before display file creation. It is created using the Create Edit Description (CRTEDTD) command. When you create a display file in which a user-defined edit code is specified, editing information is extracted from the previously created edit description. Changing a user-defined edit code after display file creation does not affect the display file unless the display file is re-created.

The following table shows edit codes, unedited source data, and edited output. Zero suppression and decimal characters are determined by the system value QDECFMT. The date separator character is determined by the job attribute DATSEP. In this figure, QDECFMT is assumed to equal x (blank), and DATSEP is assumed to equal / (slash).

Table 7. Valid edit codes, source data, and edited output

Edit codes	Positive number with two decimal positions	Positive number with no decimal positions	Negative number with three decimal positions[1]	Negative number with no decimal positions[1]	Zero balance with two decimal positions[1]	Zero balance with no decimal positions[1]
Unedited	1234567	1234567	xxxx.125–	125–	xxxxxx	xxxxxx
1	12,345.67	1,234,567	.125	125	.00	0
2	12,345.67	1,234,567	.125	125		
3	12345.67	1234567	.125	125	.00	0
4	12345.67	1234567	.125	125		
A	12,345.67	1,234,567	.125CR	125CR	.00	0
B	12,345.67	1,234,567	.125CR	125CR		
C	12345.67	1234567	.125CR	125CR	.00	0
D	12345.67	1234567	.125CR	125CR		
J	12,345.67	1,234,567	.125-	125-	.00	0
K	12,345.67	1,234,567	.125-	125-		
L	12345.67	1234567	.125-	125-	.00	0
M	12345.67	1234567	.125-	125-		
N	12,345.67	1,234,567	-.125	-125	.00	0
O	12,345.67	1,234,567	-.125	-125		
P	12345.67	1234567	-.125	-125	.00	0

Table 7. Valid edit codes, source data, and edited output (continued)						
Edit codes	**Positive number with two decimal positions**	**Positive number with no decimal positions**	**Negative number with three decimal positions[1]**	**Negative number with no decimal positions[1]**	**Zero balance with two decimal positions[1]**	**Zero balance with no decimal positions[1]**
Q	12345.67	1234567	-.125	-125		
W[2]	1234/567	1234/567	0/125	0/125	0/000	0/000
Y[3]	123/45/67	123/45/67	0/01/25	0/01/25	0/00/00	0/00/00
Z[4]	1234567	1234567	125	125		

Notes:

1. The x represents a blank.
2. The W edit code suppresses the farthest left zero of a date field that is five digits long. It also suppresses the three farthest left zeros of a field that is six to eight digits long. For more information, see the second footnote in Table 6 on page 100.
3. The Y edit code suppresses the farthest left zero of a date field that is three to six digits long or eight digits long. It also suppresses the two farthest left zeros of a field that is seven positions long. For more information, see the third footnote in Table 6 on page 100.
4. The Z edit code removes the sign (plus or minus) and suppresses leading zeros.

Example

The following example shows how to specify the EDTCDE keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A            PRICE          5  2   1 10EDTCDE(J)
00020A            SALES          7  2   2 10EDTCDE(K $)
00030A            SALARY         8  2   3 10EDTCDE(1 *)
     A
```

The display length for PRICE is 7 because the J edit code is specified, causing the field to contain a decimal point and an ending minus sign. It is edited as:

```
ddd.dd-
```

where d represents a digit.

The display length for SALES is 11 because the K edit code and floating currency symbol are specified. It is edited as:

```
$dd,ddd.dd-
```

The display length for SALARY is 10 because the edit code 1 is specified with asterisk fill. It is edited as:

```
ddd,ddd.dd
```

Related concepts

Numeric only (Y)
You can only type the numbers 0 through 9, plus (+), minus (-), period (.), comma (,), and space ( ) into the field. You can press any key to leave the field.

Related information

System values

EDTMSK (Edit Mask) keyword for display files

You use this field-level keyword to define an edit mask for fields with EDTCDE or EDTWRD keywords.

When a field is displayed with this keyword, user-specified areas of the field are protected. The EDTMSK keyword is ignored when the workstation is not attached to a controller that supports an enhanced data stream.

The format of the keyword is:

```
EDTMSK(edit mask)
```

One parameter must be specified.

The edit mask is made of two characters: an ampersand (&) and a blank ( ). The ampersand represents a protected part of the field. A blank represents an unprotected part of the field. The edit mask must equal the display length of the field (after editing), and the number of unprotected positions must equal the program length of the field. The user must be careful to protect only nonnumeric data because protected data is not returned to the user if the field is changed.

The field containing the EDTMSK keyword must be usage I or usage B. It must also contain the EDTCDE or EDTWRD keywords.

The following keywords cannot be specified on a field with the EDTMSK keyword:

- AUTO (RAB, RAZ)
- CHECK(AB, MF, RB, RZ, RLTB)
- CHOICE
- CNTFLD
- DSPATR(OID SP)

Option indicators are not valid for this keyword.

Although the maximum number of input fields is 256, the use of the EDTMSK keyword can impact the field count. The segments of EDTMSK fields are treated at run time as separate input fields on some emulators or controllers. Some controllers consider each segment of the field as an actual input field, and each segment counts toward the 256 maximum. However, some emulators count a EDTMSK field as a single field toward the count of 256. But, they might keep track of the segments separately. One emulator is known to have a maximum of 512 segments allowed. Therefore, when an emulator and a controller use the same display file, the emulator might allow more EDTMSK fields than the controller.

If the ENTFLDATR keyword is specified with the EDTMSK keyword, you might have unpredictable results.

Example

The following example shows how to specify the EDTMSK keyword:

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A          R RECORD
     A            F1            11  0B  3  4EDTWRD('0(   )   -    ')
     A                                       EDTMSK(' &   &   &    ')
     A            F2             6  0B  4  4EDTCDE(Y)
     A                                       EDTMSK('  &  &  ')
     A
```

In this example, the dash and parentheses in field 1 are protected. Also, the data separators in field 2 are protected.

EDTWRD (Edit Word) keyword for display files

You use this field-level keyword to specify an edit word if you cannot obtain the editing that you want through the EDTCDE keyword.

The format of the keyword is:

```
EDTWRD('edit-word')
```

An edit word specifies the form in which the field values are to be displayed and clarifies the data by inserting characters directly, such as decimal points, commas, floating- and fixed-currency symbol, and credit balance indicators. The edit word can also be used to suppress leading zeros and to provide asterisk fill protection.

If a field previously defined in a database file has EDTWRD specified, you need not specify EDTWRD for that field in the display file. You can specify R in position 29 to refer to the previously defined field. The editing specified for the referenced field is included in the display file. However, if you also specify length, data type, or decimal positions for a display file field, editing specified for the referenced field is not included in the display file, and you must specify editing again in the display file.

Parts of an edit word

An edit word consists of three parts: the body, the status, and the expansion. Figure 18 on page 105 shows the three parts of an edit word.

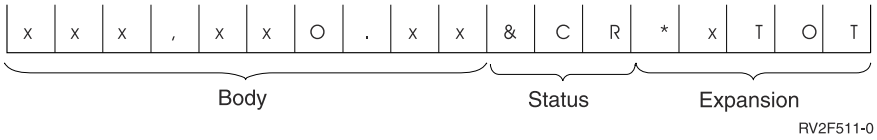


Figure 18. Three parts of an edit word

The body is the space for the digits transferred from the data field to the output record. The body begins at the farthest left position of the edit word. The number of blanks (plus one for a zero or an asterisk) it contains is equal to the number of digits of the data field to be edited. If the zero or asterisk is the first character in the edit word, the number of blanks it contains might equal the number of digits in the data field. The body ends with the farthest right character that can be replaced by a digit.

The status positions display the sign (+ or -) of the data field. The status continues to the right of the body to either a CR (credit) or - (minus) symbol. These two symbols print only when the field is negative. Edit words without the CR or - symbol have no status positions.

The expansion positions are not changed by the edit operation. The expansion starts at the first position to the right of the status (or body, if status is not specified) and ends with the farthest right character of the edit word.

Forming the body of an edit word

The following characters have special meanings when used in the body of an edit word.

Blank
A blank is replaced with the character from the corresponding position of the data field. A blank position is referred to as a digit position.

Ampersand
An ampersand causes a blank in the edited field. The ampersand is not displayed. Note that ampersands specified in the edit word between blanks can result in incorrect data when specified for an input/output field. This is because embedded blanks in a numeric-only field are converted to zeros.

Zero
A zero stops zero suppression. Place it in the farthest right position where zero suppression is to stop. The zero is replaced with the character from the corresponding position of the data field, unless that character is a zero. Any zeros in the data that appear to the right of the stop-zero-suppression character are displayed. The stop-zero-suppression character is considered a digit position; however, when it is the first character, it might not represent the digit position. At least one leading zero is suppressed, unless it is the first character of the EDTWRD. Then it does not count as a digit because the number of blanks equals the number of digits in the field. Each zero that is suppressed is replaced by a blank. An asterisk replaces zeros with asterisks (asterisk protection). Place the asterisk in the

farthest right position where zero suppression is to stop. Each zero that is suppressed is replaced by an asterisk.

Note: If your display file was created before Version 2 Release 1, it is possible that the Edit Word (EDTWRD) keyword will produce different output after a recompile.

Asterisk
An asterisk preceding a zero is interpreted as representing asterisk protection, and in this case, the zero prints as a constant. Any asterisks or zeros to the right of the stop-zero-suppression character are constants.

Currency symbol

If you code a currency symbol immediately to the left of the zero suppression code, a currency symbol is inserted in the position to the left of the first significant digit. It is called the *floating-currency symbol* when used in this manner.

If you code a currency symbol in the farthest left position of the edit word, it is fixed and prints in the same location each time. When used in this manner, it is called the *fixed-currency symbol*.

The currency symbol is not considered a digit replace position. This symbol must correspond to the system value QCURSYM.

Decimals and commas

Decimals and commas are printed in the same relative positions in which they are coded in the edit word unless they are to the left of the first significant digit. In that case, they are blanked out or replaced by an asterisk.

All other characters are printed if they are to the right of significant digits in the edit word. If they are to the left of the high-order significant digits in the edit word, they are blanked out or replaced by asterisks if asterisk protection is being used.

If a constant is to be printed in the left most position, the constant must be preceded by a zero and the field length increased by one.

Forming the status of an edit word

The following characters have special meanings when used in the status of an edit word.

Ampersand
Causes a blank in the edited output field. An ampersand cannot be placed in the edited output field.

CR or minus symbol
If the sign in the edited output field is plus (+), these positions are blanked out. If the sign in the edited output field is minus (-), these positions remain undisturbed.

Forming the expansion of an edit word

The characters in the expansion portion of an edit word are always written. The expansion cannot contain blanks. If a blank is required in the edited output field, specify an ampersand in the body of the edit word.

Specifying a valid edit word

Use the following rules to specify a valid edit word:

- The EDTWRD keyword is valid for numeric only fields (Y specified in position 35).
- You cannot specify both EDTWRD and EDTCDE for the same field.
- The DFT and DFTVAL keywords cannot be specified with the EDTWRD keyword.
- Enclose the edit word in single quotation marks.
- The sum of the blanks and stop-zero-suppression characters (digit positions) in the edit word must equal the length of the field.

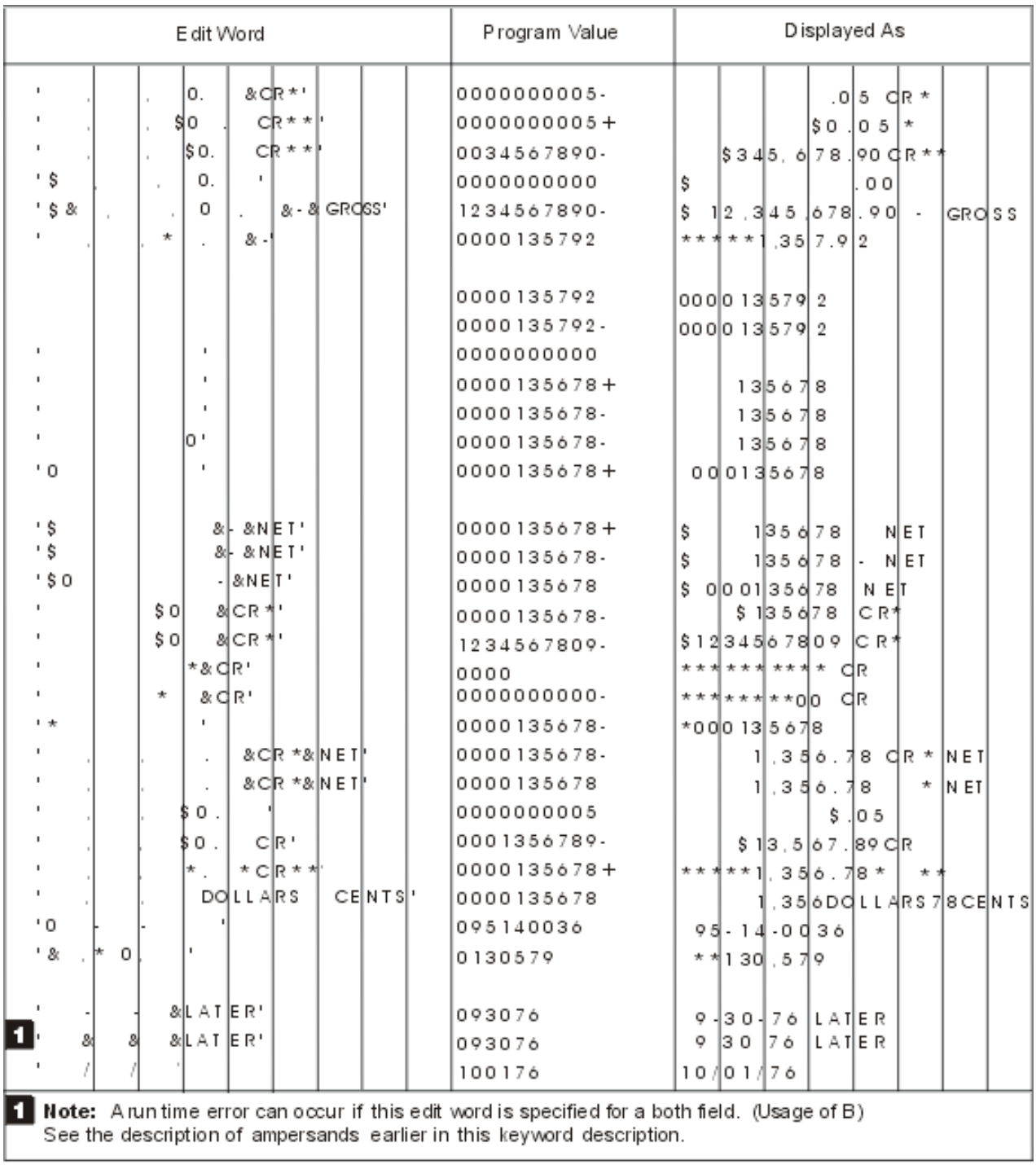
- If the stop-zero-suppression character is the first character in the edit word, the sum of the blanks might equal the length of the field or the length of the field minus one.
- When you use the floating-currency symbol, the currency symbol is not counted as a digit position. For example, if you specify the floating-currency symbol for a field with a length of 7 and 2 decimal positions, the edit word is:

  ```
  EDTWRD('____$0.__)
  ```

  where _ represents a blank.
- If you want to show a negative sign with a negative number, include a sign in the edit word. Use either the minus sign (-) or the letters CR (credit) to the right of the last digit replacement character. These print only if the number is negative.

Option indicators are not valid for this keyword.

Figure 19 on page 108 shows sample edit words with the program value of the field and the display value of the field (as edited).



Edit Word	Program Value	Displayed As
' . . 0. &CR*'	0000000005-	.05 CR*
' . . $0 . CR**'	0000000005+	$0.05 *
' . . $0. CR**'	0034567890-	$345,678.90CR**
'$ . . 0. '	0000000000	$ .00
'$& . . 0 . &-&GROSS'	1234567890-	$ 12,345,678.90 - GROSS
' . . * . &-'	0000135792	*****1,357.92
	0000135792	0000135792
	0000135792-	0000135792
' '	0000000000	
' '	0000135678+	135678
' '	0000135678-	135678
' 0'	0000135678-	135678
'0 '	0000135678+	0001356 78
'$ &- &NET'	0000135678+	$ 135678 NET
'$ &- &NET'	0000135678-	$ 135678 - NET
'$0 -&NET'	0000135678	$ 0001356 78 NET
' $0 &CR*'	0000135678-	$135678 CR*
' $0 &CR*'	1234567809-	$1234567809 CR*
' *&CR'	0000	******** CR
' * &CR'	0000000000-	*******00 CR
'* '	0000135678-	*0001356 78
' . . . &CR*&NET'	0000135678-	1,356.78 CR* NET
' . . . &CR*&NET'	0000135678	1,356.78 * NET
' . . $0. '	0000000005	$.05
' . . $0. CR'	0001356789-	$13,567.89CR
' . . *. *CR**'	0000135678+	*****1,356.78* **
' . . DOLLARS CENTS'	0000135678	1,356DOLLARS78CENTS
'0 - - '	095140036	95-14-0036
'& .* 0 '	0130579	**130,579
' - - &LATER'	093076	9-30-76 LATER
1 ' & & &LATER'	093076	9 30 76 LATER
' / / '	100176	10/01/76

1 **Note:** A run time error can occur if this edit word is specified for a both field. (Usage of B) See the description of ampersands earlier in this keyword description.

Figure 19. Sample edit words

Example

The following example shows how to specify the EDTWRD keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00020A           FIELDA         7  2   5  2EDTWRD('    $0. ')
     A
```

Related concepts

Numeric only (Y)

You can only type the numbers 0 through 9, plus (+), minus (-), period (.), comma (,), and space ( ) into the field. You can press any key to leave the field.

ENTFLDATR (Entry Field Attribute) keyword for display files

You use this field-level, record-level, or file-level keyword to define that the leading attribute of the field changes to a specified attribute whenever the cursor is located in the field.

When defined at both the field- and record-level, the field-level specification is used for the field. The ENTFLDATR keyword is ignored when the workstation is not attached to a controller that supports an enhanced data stream.

The format of the keyword is:

```
ENTFLDATR[([color] [display attribute] [cursor visible])]
```

The parameters are optional for the keyword.

The color parameter specifies the color the field will change to when the cursor enters the field on a color workstation. The parameter is specified as an expression of the form (*COLOR value).

The valid values for the color parameter are:

Value
Meaning

BLU
Blue

GRN
Green

PNK
Pink

RED
Red

TRQ
Turquoise

YLW
Yellow

WHT
White

If the color parameter is not specified, the default is white.

The display-attribute parameter specifies the display attributes of the field when the cursor enters the field. The parameter is specified as an expression of the form (**DSPATR value1 <value2 <value3...>>*).

The valid values for the display attributes are:

Value
Meaning

BL
Blink

CS
Column separator

HI
High intensity

ND
Nondisplay

RI
Reverse image

UL
Underline

The default display attribute is HI.

Note: Display attributes CS, HI, and BL can cause fields on 5292, 3179, 3197 Models C1 and C2, 3487 Models HC, and 3488[6] workstations to appear as color fields. Separator lines will not appear when display attributes HI, RI, and UL are used.

The cursor visible parameter allows the user to specify if the cursor is visible or invisible when it enters the field. *CURSOR means the cursor will stay visible and *NOCURSOR indicates that the cursor will become invisible when it enters the field. *CURSOR is the default. When *NOCURSOR is specified on the ENTFLDATR keyword, the specified field must have an I (inhibit keyword entry) in position 35. If the field does not have data type I, then the default is used for the visible cursor parameter.

The field containing the ENTFLDATR keyword must be an input-capable field. The ENTFLDATR keyword is ignored for the field with DSPATR(PR).

Option indicators are valid for this keyword.

If the ENTFLDATR keyword is specified with the EDTMSK keyword, you might have unpredictable results.

Example

The following example shows how to specify the ENTFLDATR keyword:

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A
     A     R RECORD
     A       F1            10A  B  3  4ENTFLDATR
     A       F2            10A  B 13  4ENTFLDATR((*COLOR RED))
     A       F3            10I  B 16  4ENTFLDATR(*NOCURSOR (*DSPATR HI RI))
```

In this example, the color turns white, the attribute is high intensity, and the cursor is visible for F1. For F2, the color is red, the attribute is high intensity, and the cursor is visible. For F3, the color is white, the attributes are high intensity and reverse image, and the cursor is not visible.

Related reference
COLOR (Color) keyword for display files
You use this keyword to specify the color of a field on a color display.

ERASE (Erase) keyword for display files

You use this record-level keyword with the OVERLAY keyword to specify that the records whose names you supplied as parameter values are to be erased from the display when this record is written.

The format of the keyword is:

```
ERASE(record-name-1 [record-name-2 ...[record-name-20]])
```

The record formats specified as parameter values must exist in the file.

ERASE can be specified more than once. The OVERLAY keyword must be specified whenever the ERASE keyword is specified.

If the ERASE and CLRL keywords are both in effect on an output operation, the records specified in the ERASE keyword are erased despite the CLRL keyword.

A record already on the display that has no input-capable fields and that has the CLRL specified cannot be erased by the ERASE keyword on another record (ERASE has no effect).

If the specified record is not on the display, this function is ignored for that record name.

Option indicators are valid for this keyword.

[6] Dependent on the monitor attached to the display device.

Note: This function requires extra data transmission and should be used only when you do not want to erase all other record formats. You can also specify the OVERLAY keyword with option indicators so that you can select when to erase all other formats.

Example

The following example shows how to specify the ERASE keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00020A          R REC1
     A             :
     A             :
     A          R REC2                       OVERLAY
     A             :
     A             :
     A          R REC4                       OVERLAY
     A                                       ERASE(REC1)
     A
```

ERASEINP (Erase Input) keyword for display files

You use this record-level keyword with the OVERLAY keyword to erase unprotected input-capable fields already on the display. Unprotected input-capable fields are fields for which the DSPATR(PR) keyword is not in effect.

The fields are erased before the record format you are defining is displayed. Input-capable fields in the record format you are defining are not erased.

See the Application Display Programming book for information about how to use ERASEINP in files that are used in the System/36 environment.

The format of the keyword is:

```
ERASEINP[(*MDTON | *ALL)]
```

To erase all input-capable fields already on the display, specify the *ALL parameter. To erase only input-capable fields that have their changed data tags (MDTs) set on, specify the *MDTON parameter. Specifying ERASEINP(*MDTON) or ERASEINP has the same effect as pressing the Erase Input key.

The OVERLAY keyword must be specified whenever ERASEINP is specified.

When the MDTOFF keyword is specified on the same record format as ERASEINP, two conditions can occur:

- ERASEINP(*ALL) implies MDTOFF(*UNPR), unless the MDTOFF(*ALL) keyword is specified.
- If the ERASEINP or ERASEINP(*MDTON) is specified with MDTOFF(*ALL), the end effect is as if ERASEINP(*ALL) and MDTOFF(*ALL) are both specified.

If the ERASEINP and PROTECT keywords are both in effect for an output operation, the IBM i operating system first erases the input-capable fields specified on the ERASEINP parameter value, and then protects all input-capable fields on the display from input typing.

ERASEINP reduces line traffic because record formats already displayed can be reused and do not need to be sent to the display again.

A warning message appears at file creation time if the ERASEINP keyword is specified in a record with the DSPMOD keyword. At run time, the ERASEINP keyword is ignored when the display mode changes.

Option indicators are valid for this keyword.

Example

The following example shows how to specify the ERASEINP keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A*
00020A*
00030A*
00040A          R RECORD1                   OVERLAY
00050A              :                       ERASEINP
00060A              :
00070A              :
00080A*           (All changed input-capable fields on the screen are erased)
00090A              :
00100A              :
00110A              :
00120A          R RECORD2                   OVERLAY
00130A              :                       ERASEINP(*ALL)
00140A              :
00150A              :
00160A*           (All unprotected input-capable fields on the screen are erased
00170A*             whether changed or not)
00180A
00190A
00200A          R RECORD3                   OVERLAY
00210A  32                                  ERASEINP(*MDTON)
00220A N32                                  ERASEINP(*ALL)
00230A            FIELD1         5   I      DSPATR(PR)
00240A              :
00250A              :
00260A              :
00270A*           (FIELD1 is never erased)
     A
```

ERRMSG (Error Message) and ERRMSGID (Error Message Identifier) keywords for display files

You can use one of these field-level keywords to identify a message that is displayed on the message line and that is associated with that field.

A warning message appears at file creation time if either of these keywords is specified on a record with the DSPMOD keyword. At run time, these keywords are ignored when the display mode changes.

Option indicators are valid for these keywords.

ERRMSG keyword

The format of the keyword is:

```
ERRMSG('message-text' [response-indicator])
```

For ERRMSG, the parameters specify the message text and, optionally, a response indicator. The message text is the message to be displayed. (The Help key is not supported. Message help is not displayed when the Help key is pressed.)

If you specify a response indicator, it should be the same as the option indicator used to condition ERRMSG. On the input operation that follows the display of the error message, the IBM i operating system turns off the indicator. If the response and the option indicators are the same, they are both turned off. One exception to this rule is if the response indicator is also specified for another keyword, such as CHANGE, CAnn, or CFnn. In that case, the on/off setting of the response indicator is based on the results of the function provided by the CHANGE or CFnn keyword. When a response indicator is specified, the first 50 characters of the message are also used as indicator text. Separate response indicator text is not valid for the ERRMSG keyword.

ERRMSGID keyword

The format of the keyword is:

```
ERRMSGID(msgid [library-name/]msg-file [response-indicator] [&msg-data])
```

For ERRMSGID, the parameters specify:

- The message identifier for the message to be displayed
- The message file and, optionally, the library
- Optionally, a response indicator
- Optionally, a msg-data field name

The response indicator, if specified, should be the same as the option indicator used to condition the ERRMSGID keyword. On the subsequent input operation, after the display of the error message, the IBM i operating system turns off the indicator. However, if the response indicator is also specified on another keyword, such as CHANGE, CAnn, or CFnn, the on/off setting of the response indicator is based on the results of the function provided by the CHANGE, CAnn, or CFnn keyword.

Note: Indicator text cannot be specified on the ERRMSGID keyword.

The msg-data field, if specified, contains the replacement text for the specified message. The field must exist in the record format and the field must be defined as a character field (data type A) with usage P. For more information about how replacement text works, refer to the Send Program Message (CL) command in the Control language topic.

Priority among selected keywords

You can specify ERRMSG and ERRMSGID more than once for a single field. During program processing, use option indicators to select a particular message to be displayed.

Only one message can be displayed at one time even if messages are in effect for several fields on the same output operation. The field whose message is displayed is the first field for which the program selected a message.

If several keywords are in effect for one field on an output operation, the message to be displayed is the first of the following keywords:

- ERRMSG (If more than one ERRMSG keyword is selected, the first one the program selects is displayed.)
- ERRMSGID (If more than one ERRMSGID keyword is selected, the first one the program selects is displayed.)

A message field is displayed only if no error message keywords are also to be displayed.

For a list of priorities including the SFLMSG and SFLMSGID keywords, see SFLMSG (Subfile Message) and SFLMSGID (Subfile Message Identifier) keywords for display files.

Conditions occurring during message display

The displaying of a message using ERRMSG and ERRMSGID is similar to the displaying of messages by the IBM i operating system when field validation errors are detected.

When a message is displayed because of either the ERRMSG or the ERRMSGID keyword, all fields on the display are kept, including the field the message is associated with. Except for option indicators, data in the output buffer is ignored (that is, any new data from the program is not sent to the display).

The function keys valid following display of a message are:

- Function keys specified at the file level
- Function keys specified for the record format for which a message is displayed, if selected when the message is displayed

When the message is displayed, the following conditions occur:

- For all errors:
  - The message is highlighted.

  - The cursor is blinked and the keyboard locked until the workstation user presses the Reset key.
- For errors associated with input-capable fields:
  - All fields in error are displayed with their images reversed.

    If a field in error has both the underline (UL) display attribute and the highlight (HI) attribute or the underline (UL) attribute and COLOR(BLU, WHT, or YLW) specified, its image is not reversed.
  - The cursor is repositioned to the first displayed field that is in error.
- For errors associated with output-only fields:
  - The display attribute of the field is not changed.
  - The cursor is not positioned to the field (it does not change position).

Note: Some display attributes can cause fields on the IBM Color Display Station to appear as color fields.

Restoration of reversed image fields

Fields are displayed with their images reversed because of system-detected typing errors or because of the ERRMSG or the ERRMSGID keyword. Generally, the IBM i program restores the image on the next I/O operation to the display, typically the next request from your program. The restoration is done before the requested function is performed. The following list shows exceptions where requests from your program do not cause the IBM i operating system to restore reversed image fields:

- An input request with cancel (canceling a read operation with NOWAIT)
- A close request when the KEEP keyword is in effect
- Any request to a subfile record (no data is sent to the device)
- An output operation to a subfile-control record format that does not display the subfile control record or subfile records (for example, clearing, deleting, or initializing the subfile)

Restrictions and notes

- When an ERRMSG or ERRMSGID keyword is in effect, no processing other than the processing for these keywords is performed for the record. If neither keyword is in effect, the record is processed in the normal manner.
- When the RMVWDW keyword is active, error messages are not displayed.
- ERRMSG and ERRMSGID are valid for output-only, input-only, or input/output fields. These two keywords cannot be specified for a constant, hidden, program-to-system, or message field.
- For input or output capable fields, ERRMSG and ERRMSGID are in effect only if the record containing the field for which they are specified is already on the display.
- ERRMSG and ERRMSGID cannot be specified in a subfile record format (SFL keyword specified). To display error messages for a subfile, see SFLMSG (Subfile Message) and SFLMSGID (Subfile Message Identifier) keywords for display files.
- ERRMSG and ERRMSGID are ignored if the variable start line number (SLNO(*VAR) specified) has changed since the last output operation.
- If you specify ERRMSG or ERRMSGID, you should also specify RSTDSP(*YES) on the Create Display File (CRTDSPF) or Change Display File (CHGDSPF) command. Otherwise, data on the display can be lost if the file is suspended.
- On an output operation that causes the display modes to be changed, ERRMSG and ERRMSGID are ignored.

Example

The following example shows how to specify the ERRMSG and ERRMSGID keywords.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A          R CUSMST
00020A              :
```

```
00030A              :
00040A              :
00050A            QTYORD        10A  I  5  3
00060A  61                                  ERRMSG('No stock available' 61)
00070A  62                                  ERRMSG('Partial stock available' +
00080A                                      62)
00090A  63                                  ERRMSGID(MSG2000 CONSOLEMSG 63 +
00100A                                      &RPLTXT);
00110A            RPLTXT        78A  P
     A
```

Related reference

COLOR (Color) keyword for display files
You use this keyword to specify the color of a field on a color display.

SFLMSG (Subfile Message) and SFLMSGID (Subfile Message Identifier) keywords for display files
You use these record-level keywords on the subfile-control record format to identify a message to be displayed on the message line when your program performs an output operation to the subfile-control record format.

ERRSFL (Error Subfile) keyword for display files

You use this file-level keyword to specify that messages should be displayed using a system-supplied error subfile.

Messages that display in the error subfile are system validity check messages and messages associated with the following keywords:

ERRMSG	CHECK(VN)
ERRMSGID	CHECK(VNE)
SFLMSG	COMP
SFLMSGID	RANGE
CHECK(M10)	VALUES
CHECK(M11)	

Validity check messages associated with the following input errors are also displayed in the error subfile:

- Errors in floating point operations
- Decimal position entry errors

This allows you to page through all the error messages issued when a record is written to the display, and all the validity check error messages issued when a record is read from the display. The system displays the error subfile on the message line. If the message line overlaps a record already displayed on the screen, the ERRSFL keyword is ignored.

This keyword has no parameters.

Option indicators are not valid for this keyword.

Example

The following example shows how to specify the ERRSFL keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A                                      MSGLOC(24)
00020A                                      ERRSFL
00030A          R RCD1
00040A            FIELD1        5A   B  2  3
00050A  10                                  ERRMSGID(MSG0001 MSGF1 10 &MDTA);
00060A            FIELD2        5A   B  5  7
00070A                                      ERRMSG('ERROR MSG 1' 11)
00080A            FIELD3        4S   I  7  7RANGE(1000 9999)
00090A                                      CHKMSGID(MSG0002 MSGF1 &MDTA1);
00100A            FIELD4       10A   B  8  7CHECK(VN)
00110A            MDTA         78A   P
```

```
00120A            MDTA1         4A   P
     A
```

In this example, when RCD1 is read from the display, any previous messages in the error subfile are cleared. Then, if FIELD3 does not contain a value in the range 1000 to 9999 and FIELD4 does not contain a valid name, the system places the message MSG0002 and the system message associated with CHECK(VN) in the error subfile and displays the error subfile on line 24 of the display. The user can view the messages by pressing the Page Up and Page Down keys.

When RCD1 is read again from the display, the previous messages in the error subfile are cleared. Then, if FIELD3 and FIELD4 are valid, control returns to the application. If FIELD1 and FIELD2 are not valid, when the application writes RCD1 to the display with indicators 10 and 11 on, the system places the message MSG0001 and the text ERR0R MSG 1 in the error subfile and displays the error subfile on line 24 of the display. The user can view the messages by pressing the Page Up and Page Down keys.

FLDCSRPRG (Cursor Progression Field) keyword for display files

You use this field-level keyword to define the field to which the cursor moves when it exits a field.

The FLDCSRPRG keyword is ignored when the workstation is not attached to a controller that supports an enhanced data stream.

The format of the keyword is:

```
FLDCSRPRG(name of a field)
```

One parameter *must be* specified.

The field containing the FLDCSRPRG keyword is defined as an input-capable field. It cannot be defined in a subfile.

The field name is a name of an input-capable field that is defined inside the same record that this field is in.

Option indicators are not valid for this keyword.

The FLDCSRPRG keyword is not allowed with the SNGCHCFLD or MLTCHCFLD keywords.

Example

The following example shows how to specify the FLDCSRPRG keyword:

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
                R RECORD
                  F1           10A  B   3  4FLDCSRPRG(F3)
                  F2           10A  B  13  4FLDCSRPRG(F1)
                  F3           10A  B  16  4FLDCSRPRG(F2)
```

In this example, the cursor moves from field 1 to field 3, then from field 3 to field 2, and finally from field 2 to field 1.

FLTFIXDEC (Floating-Point to Fixed Decimal) keyword for display files

You use this field-level keyword to display a number in an output-capable (usage B or O) floating-point field in fixed-decimal notation.

This keyword has no parameters.

The floating-point number is first converted to the equivalent number with an exponent of zero. If the resulting number (digits and exponent) will fit in the field defined by the length and decimal positions values, the number is displayed with the exponent suppressed and aligned at the decimal point. If the number will not fit in the field defined by the length and decimal position values, the number is displayed in standard floating point form, n.nnnnnnE+nnn. When FLTFIXDEC is specified, the display length of the field is the DDS length plus 2 (the sign and the decimal point). The minimum length of the field is 6.

When the number is too large or small for the fixed-point form specified by FLTFIXDEC with the total digits and fractional digits specified for the field, a floating point form is displayed that presents the significand as follows. (The significand is the string of digits including the sign and decimal point to the left of the exponent character E.)

- Total significand decimal digits: DDS total digits minus 5
- Fractional significand digits: DDS total digits minus 6

FLTFIXDEC has no effect on the input format of the data. Numbers can be typed into the field in either fixed point or floating point format. When displayed again, however, FLTFIXDEC will be used to determine the display format.

Option indicators are not valid for this keyword.

Example

The following example shows how to specify the FLTFIXDEC keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A          R RECFMT1
     A            FIELD1        10F 3B  1  2FLTFIXDEC
     A                                     FLTPCN(*DOUBLE)
     A
```

The output numbers for the example will be converted as follows:

Output number	Displayed as
-4.99994321000000E-004	'-4.9999E-004'
-5.00010000000000E-004	' -0.001'
-2.69123400000000E-002	' -0.027'
-0.00000000000000E+000	' 0.000'
0.00000000000000E+000	' 0.000'
2.71828182845900E++003	' 2718.282'
3.14159000000000E-052	'3.1416E-052'
9.87654321012345E+006	'9876543.210'
9.99999999960000E+006	'1.0000E+007'

FLTPCN (Floating-Point Precision) Keyword for Display Files

You use this field-level keyword to specify the precision of a floating-point field.

The format of the keyword is:

```
FLTPCN(*SINGLE | *DOUBLE)
```

The valid parameters are *SINGLE (single precision) and *DOUBLE (double precision). This keyword is valid for floating-point fields only (data type F).

A single precision field can be up to 9 digits; a double precision field can be up to 17 digits. If you specify a field length greater than 9 (single precision) or 17 (double precision), an error message appears and the file is not created.

Option indicators are not valid for this keyword.

Example

The following example shows how to specify the FLTPCN keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00090A           FIELDA        17F 4   1  5FLTPCN(*DOUBLE)
     A
```

FIELDA is a floating-point field with double precision.

FRCDTA (Force Data) keyword for display files

You use this record-level keyword to display a record format immediately, without waiting for the next input or input/output operation. When the buffer is partially full, the FRCDTA keyword can be used to clear the buffer.

Note: If this keyword is used after each write statement, performance problems will occur.

This keyword has no parameters.

When this keyword is in effect for a record format, the record format is displayed as if you had specified DFRWRT(*NO) on the Create Display File (CRTDSPF) command or the Change Display File (CHGDSPF) command.

You can use this keyword when DFRWRT(*YES) is in effect for the display file and when your program does several output operations before doing an input operation. With DFRWRT(*YES) specified, none of the record formats is displayed until the input operation. There might be a long delay for the workstation user while the program completes its processing. You can specify FRCDTA for a record format that is displayed first. This record format tells the workstation user that the delay is normal. For a step-by-step description, see the following example.

The FRCDTA keyword can be specified once for each record format.

Option indicators are valid for this keyword.

Example

The following example shows how to specify the FRCDTA keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A* Following record format displays in progress message
00020A          R INPROG                      LOCK
00030A                                        FRCDTA
00040A                                   12 21'Please wait; +
00050A                                        operations in progress'
     A
00060A* Following record format uses upper part of screen
00070A          R RCD1                        OVERLAY
00080A                                     1 34'Sample Title'
00090A            FLD1           8  0   3  2
00100A            FLD2          20      4  2
     A
00110A* Following record format uses middle part of screen
00120A          R RCD2                        OVERLAY
00120A            FLDA           8     11  2
00140A            FLDB          18     12  2
     A
00150A* Following record format uses lower part of screen
00160A          R RCD3                        OVERLAY
00170A            FLDC           8   B 15  2
00180A            FLDD           8   B 16  2
     A
```

Three record formats (RCD1, RCD2, and RCD3) are used to create a single display; each of these record formats uses only a part of the display. Record format INPROG prepares the workstation user for the delay while the other three record formats are prepared. The program does the following operations:

- Displays record format INPROG. With FRCDTA specified, this displays the in-progress notice immediately. The keyboard remains locked because the LOCK keyword is specified.

- Continues processing to prepare the other three record formats (RCD1, RCD2, and RCD3), then displays them. Because they overlap record format INPROG, it is erased.
- Reads record format RCD3. This unlocks the keyboard and the workstation user can respond to the complete display.

GETRETAIN (Get Retain) keyword for display files

You use this record-level keyword with the UNLOCK keyword to specify that the IBM i operating system is not to erase input-capable fields on input operations as described under the UNLOCK keyword.

This keyword has no parameters.

Using GETRETAIN reduces input typing when successive records contain mostly identical input. With GETRETAIN specified, the workstation user needs only to change certain input fields rather than retyping the entire record.

You must specify the UNLOCK keyword without any parameters when using GETRETAIN. The UNLOCK(*MDTOFF) specification provides the same function as GETRETAIN.

Option indicators are not valid for this keyword.

Example

The following example shows how to specify the GETRETAIN keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00101A          R REC1                      GETRETAIN
     A                                      UNLOCK
     A
```

Related reference

UNLOCK (Unlock) keyword for display files
You use this record-level keyword to specify that the IBM i operating system is to unlock the keyboard immediately after issuing an input operation to the record format you are defining.

HELP (Help) keyword for display files

You use this file-level or record-level keyword to enable the Help key.

See “System/36 environment considerations for display files” on page 247 for special considerations when you specify the HELP keyword in files that are used in the System/36 environment.

The format of the keyword is:

```
HELP[(response-indicator ['text'])]
```

If you specify a response indicator, the response indicator is set on and returned to your program. No input data is transmitted from the device. Processing is similar to that of a command attention key.

The optional text is included on the listing created at program compilation time to explain the intended use of the indicator. This text has no function in the file or program other than as a comment. The single quotation marks are required. If you specify more than 50 characters between the single quotation marks, the text is truncated to 50 characters on the program list.

One of the following actions occurs when the Help key is pressed:

- Second level text for a message is displayed if the cursor is located in a message subfile or on a field which specified either the ERRMSGID or SFLMSGID keyword.
- Online help information associated with either the entire display or an area on the display. This function is indicated by H specifications (refer to HLPARA (Help Area) keyword for display files) HLPDOC keyword.
- Control is returned to the user's program. This occurs when there are no H specifications or file-level HLPDOC, HLPPNLGRP, or HLPRCD keywords in the file.

If this keyword is not specified and the Help key is pressed, the IBM i operating system issues an error message indicating that the Help key is not valid at that time.

The HLPRTN keyword allows you to use option indicators to select when online help information is displayed, and when control is returned to the program.

When a response indicator is specified on the HELP keyword, no H specifications or HLPRCD, HLPPNLGRP, HLPDOC, or HLPRTN keywords can be specified in the file.

HELP (with no response indicator) is required if the file contains H specifications or HLPRCD, HLPPNLGRP, HLPDOC, or HLPRTN keywords.

Option indicators are valid for this keyword.

Example

The following example shows how to specify the HELP keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00024A                                      HELP
     A          R RECORD1
     A
```

Related reference

Type of name or specification for display files (position 17)
You can specify a value in this position to identify the type of name in positions 19 through 28.

ALTHELP (Alternative Help Key) keyword for display files
You use this file-level keyword to assign a command attention (CA) key as an alternative Help key.

HLPRTN (Help Return) keyword for display files
You use this file-level or record-level keyword to return control to your program when you press the Help key.

HLPARA (Help Area) keyword for display files
You use this help-specification-level keyword to define a rectangular area on the display.

HLPARA (Help Area) keyword for display files

You use this help-specification-level keyword to define a rectangular area on the display.

If the cursor is located in this area when you press the Help key, the online help information specified for that help (H) specification (on a HLPDOC, HLPPNLGRP, or HLPRCD keyword) is displayed.

The format of the keyword is:

```
HLPARA(top-line left-position bottom-line right-position)
or
HLPARA(*RCD)
or
HLPARA(*NONE)
or
HLPARA(*FLD field-name [choice-number])
or
HLPARA(*CNST help-identifier)
```

Observe the following rules when the line and position values are specified as parameters:

- The line and position values must be within the display size.
- If you do not specify HLPARA for a secondary display size, the HLPARA of the primary display size is used if it is valid for the secondary display size. HLPARA(*NONE) is used if the HLPARA of the primary display size is not valid for the secondary display size.
- The top line must not exceed the bottom line and the left position must not exceed the right position.
- If you specify the SLNO(n) keyword on the record, the top-line and bottom-line values are adjusted, and any errors are diagnosed at creation time. If you specify the SLNO(*VAR) keyword on the record, the top line and bottom line are adjusted at processing time.

The special value *RCD indicates that the help area is the area of the record containing the H specification. This area includes all display positions in every line occupied by the record.

HLPARA(*RCD) is not valid for subfile control (SFLCTL) or user-defined (USRDFN) record formats.

If you specify HLPARA(*RCD) on an H specification, the record format containing the H specification must contain at least one displayable field for the primary display size. Hidden (H in position 38), message (M in position 38), and program-to-system (P in position 38) fields and fields that specify a SFLPGMQ or SFLMSGKEY keyword are not displayable.

The special value *NONE indicates that no help area is associated with the help information defined on this H specification. If the help information is defined using the UIM (HLPPNLGRP keyword), it is not displayed as item-specific help when the Help key is pressed, but might be displayed as extended help. If the help information is defined using DDS (HLPRCD keyword), it is not displayed as primary help when the Help key is pressed, but it might be displayed as secondary help when the Page Up or Page Down key is pressed on another help display. The *NONE value is not useful when the help information is defined in a document (HLPDOC keyword), because this information will never be displayed when the Help key is pressed.

The special value *FLD indicates that the help area is the area of a field. If the field occupies only one line, the help area consists of the first and last characters of the line and all the characters in between.

If the field wraps from one line to another, the help area consists of the entire length of all lines in the field. For example, if a field starts on line 3, position 4 and ends on line 5, position 10, the help area starts in line 3, position 1 and ends in line 5, position 80.

If the field is a choice or a continued-entry field, the help area consists of the rectangular area occupied by the choice or continued-entry field.

The field-name parameter specifies the name of the field for which the help area is defined. The field must exist in the record containing the H specification.

If the choice-number parameter is specified, the help area is the area of the choice within the field specified. When a choice number is specified, the field name must be the name of a menu-bar field or a selection field, and the choice number you specify must also be specified on a MNUBARCHC or CHOICE keyword for that field. Valid values for the choice number are positive integers greater than 0 and less than 100.

The *CNST special value indicates that the help area is the area of a constant field. This area includes the beginning and ending attribute bytes of the field.

The help-id parameter is a number that identifies the constant field for which this help area is defined. The constant field must exist in the record containing the H specification, and it must have the HLPID keyword specified with the same help-identifier.

You must specify at least one HLPARA keyword on an H specification. When you specify multiple HLPARA keywords for each H specification, you must use display size conditioning.

Help areas can overlap when multiple H specifications are specified on a record. When multiple H specifications are specified, the first H specification with both of the following characteristics is used:

- The help area specified on the HLPARA keyword contains the current cursor location.
- The option indicator on the HLPRCD, HLPPNLGRP, or HLPDOC keyword was in effect when the application record was written to the display.

The following rules apply to H specifications:

- An H in position 17 denotes the start of an H specification. The H specification must be located in the DDS after the record-level keywords and before the first field in that record.
- Each H specification must have exactly one HLPRCD, HLPPNLGRP, or HLPDOC keyword, up to one HLPBDY or HLPEXCLD keyword, and at least one HLPARA keyword.
- The end of the H specification is denoted by another H in position 17 or the first field.
- You cannot use H specifications in subfile (SFL keyword) record formats. H specifications are not allowed in subfile control formats associated with message subfiles (SFLMSGRCD keyword).

Option indicators are not valid for this keyword.

Example 1

The following example shows how to specify the HLPARA keyword:

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A                                     HELP
     A                                     HLPRCD(DFTHELP)
     A          R RECORD1
     A          H                          HLPARA(1 5 3 15)
     A                                     HLPDOC(FLDHELP DOC1 FOLDER1)
     A
     A          H                          HLPARA(*RCD)
     A                                     HLPRCD(HELPRCD1)
     A
     A          H                          HLPARA(*NONE)
     A                                     HLPRCD(HELPRCD2)
     A            FIELD1        10A     2  5
     A            FIELD2        40A    10 10
     A
```

In this example, the HLPARA keyword on the first H specification indicates that the area from line 1, position 5 to line 3, position 15 is to be associated with the online help information document DOC1. If the cursor is located in this area when the Help key is pressed, document DOC1 is displayed beginning at label FLDHELP.

The HLPARA keyword on the second H specification indicates that the area occupied by RECORD1 (lines 2 through 10) is to be associated with the online help information record HELPRCD1. If the cursor is located anywhere on lines 2 through 10 (outside the area defined by the first H specification) when the Help key is pressed, record HELPRCD1 is displayed.

The HLPARA keyword on the third H specification indicates that no area is to be associated with the online help information HELPRCD2. HELPRCD2 can only be displayed by pressing the Page Up or Page Down key from a online help information display.

Example 2

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A                                     HELP HLPRCD(DFTHELP)
     A          R RECORD
     A          H                          HLPARA(*FLD F1 1)
     A                                     HLPRCD(UNDOHLP HLPLIB/HLPFILE)
     A          H                          HLPARA(*FLD F1 2)
     A                                     HLPRCD(MARKHLP HLPLIB/HLPFILE)
     A          H                          HLPARA(*FLD F1 3)
     A                                     HLPRCD(COPYHLP HLPLIB/HLPFILE)
     A          H                          HLPARA(*FLD F2)
     A                                     HLPRCD(F2HLP HLPLIB/HLPFILE)
     A          H                          HLPARA(*CNST 1)
     A                                     HLPRCD(TITLEHLP HLPLIB/HLPFILE)
     A                                  1 37'Title' HLPID(1)
     A            F1             2Y 0B 10  2SNGCHCFLD
     A                                     CHOICE(1 'Choice 1')
     A  01                                 CHOICE(2 'Choice 2')
     A                                     CHOICE(3 'Choice 3')
     A            F2            10A  B 10 30
     A
```

The HLPARA keyword on the first three H specifications indicates the areas occupied by Choice 1, Choice 2, and Choice 3, which are to be associated with online help information. If Choice 2 is turned off so that Choice 3 moves up one line, the help area for Choice 3 automatically moves with the choice.

The HLPARA keyword specified on the 4th H specification indicates the area of F2 to be associated with online help information. This area is line 10, from position 29 through position 40.

The HLPARA keyword specified on the 5th H specification indicates the area of the constant title to be associated with online help information. This area is line 1, from position 36 through position 42.

Related reference

HLPRTN (Help Return) keyword for display files
You use this file-level or record-level keyword to return control to your program when you press the Help key.

HELP (Help) keyword for display files
You use this file-level or record-level keyword to enable the Help key.

HLPBDY (Help Boundary) keyword for display files

You use this help-specification-level keyword to limit the online help information that is available when online help information is displayed.

This keyword has no parameters.

If the HLPBDY keyword is not specified, the online help information associated with all active H specifications (accumulated for all records on the display) is accessible to the user. Specifying the HLPBDY keyword partitions the list into sublists by defining help boundaries. Each sublist contains the H specifications specified between help boundaries. The H specification that has the HLPBDY keyword is considered to be before the boundary. If the help information is defined using DDS (HLPRCD keyword), the user has access only to the help information in the sublist containing the H specification selected when the Help key is pressed. If the help information is defined using the UIM (HLPPNLGRP keyword), the sublists determine the extended help. The extended help consists of the file level HLPPNLGRP followed by the sublist containing the H specification selected for item-specific help.

Option indicators are valid for this keyword.

Example

The following example shows how to specify the HLPBDY keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A                                     HELP
     A                                     HLPTCD(DFTHELP)
     A          R RECORD1
     A*
     A* This is H-spec 1
     A*
     A          H                          HLPARA(1 5 3 15)
     A                                     HLPRCD(HELPRCD1)
     A*
     A* This is H-spec 2
     A*
     A          H                          HLPARA(*NONE)
     A                                     HLPRCD(HELPRCD2)
     A                                     HLPBDY
     A*
     A* This is H-spec 3
     A*
     A          H                          HLPARA(4 5 6 15)
     A                                     HLPRCD(HELPRCD3)
     A  90                                 HLPBDY
     A*
     A* This is H-spec 4
     A*
     A          H                          HLPARA(8 5 10 15)
     A                                     HLPRCD(HELPRCD4)
     A                                     HLPBDY
     A            FIELD1        10A     1 10
     A
```

The list of H specifications for RECORD1 is divided into two or three sublists, depending on the condition of indicator 90 at the time the Help key is pressed. If indicator 90 is off, there are two sublists. The first sublist contains H specifications 1 and 2, and the second sublist contains H specifications 3 and 4. If indicator 90 is on, there are three sublists. The first sublist contains H specifications 1 and 2, the second sublist contains H specification 3, and the third sublist contains H specification 4.

HLPCLR (Help Cleared) keyword for display files

You use this record-level keyword to clear the list of active help specifications. When this record is displayed, only the online help information that is defined on the current record format or on the file level is accessible.

This keyword has no parameters.

If the HLPCLR keyword is not specified, the help specifications are accumulated for all records on the display and are active until the record containing the help specification either is cleared from the display or is completely overlapped by another record.

Option indicators are allowed on this keyword.

The record specifying the HLPCLR keyword must contain at least one help specification. Because help specifications are ignored for an override operation with USRDSPMGT specified, the use of HLPCLR with USRDSPMGT and PUTOVR results in no online help being available for display. To maintain the available online help, HELPCLR should be turned off when PUTOVR is in effect.

Example

The following example shows how to specify the HLPCLR keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A          R RECORD1                   USRDFN
     A                                      HLPCLR
     A          H                           HLPARA(1 10 1 30)
     A                                      HLPRCD(RECORDA FILE1)
     A          H                           HLPARA(4 10 4 30)
     A                                      HLPDOC(A B C)
     A
```

When RECORD1 is displayed, only the online help information associated with the two help specifications defined in RECORD1 is available for display.

HLPCMDKEY (Help Command Key) keyword for display files

You use this record-level keyword to return control to your application program after a command attention (CA) or command function (CF) key is pressed on an application help record format.

This keyword is specified on the application help record format. For control to return, the command key must be specified on both the application record format and the application help record format.

This keyword has no parameters.

A CA or CF key must be specified either at the file level or on the help record containing the HLPCMDKEY keyword. If no CA*nn* or CF*nn* keys are specified at the file level or on the help record, a warning message (severity 10) is issued. If all the CA*nn* and CF*nn* keys specified at the file level and on the same help record as the HLPCMDKEY keyword have option indicators, a warning message (severity 10) is issued. When a response indicator is specified on a CA or CF key on the application help record format, a warning message (severity 10) is issued and the response indicator will be ignored.

If you specify this keyword on the help record and the display station user presses one of the command keys that is specified on both the application record and the application help record, the following case happens:

- If the command key is a CAnn key, no input data from the application record format is transmitted to the application program.
- If the command key is a CFnn key, input data from the application record format is transmitted to the application program.
- The command key will be returned to the application program.

The command key must be specified on both the application record format and the application help record format. If the command key is specified only on the application record format, the command key will not be allowed when the application help record format is displayed. If the command key is specified

only on the application help record format, the command key will function as the Enter key and control will not return to the application program.

You cannot specify HLPCMDKEY on subfile (SFL keyword), subfile control (SFLCTL keyword), or user-defined (USRDFN keyword) record formats.

You cannot specify the HLPCMDKEY keyword in a file containing the USRDSPMGT keyword.

Option indicators are not valid for this keyword.

Example

The following example shows how to specify the HLPCMDKEY keyword. The first record is the application record format and the second record is the application help record format.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A          R APPRCD                    CA01
00020A                                      CA03
00030A                                      CF12(12)
00040A                                      CA04
00050A                                      HELP
00060A          H                           HLPRCD(HELPRCD)
00070A                                      HLPARA(1 1 24 80)
00080A                                    8  2'THIS IS THE APPLICATION'
00090A                                    9  2'RECORD FORMAT'
00100A            INPUT1        10   B 12 10
00110A            INPUT2        10   B 13 10
00120A            INPUT3        10   B 14 20
00130A*
00140A          R HELPRCD                   HLPCMDKEY
00150A                                      CA01(11)
00160A                                      CF03
00170A                                      CF05
00180A                                      CF12
00190A                                    5  8'SPECIFY COMPANY NAME'
00200A                                    6  9'SPECIFY STREET'
00210A                                    7 10'SPECIFY CITY, STATE, ZIP'
     A
```

If the user is at the application help display, the following case occurs:

- If the CMD1 key is pressed, control is returned to the application program, but no data from the application record format is transmitted to the application program. Response indicator 11 is not set on because it was specified on the help record format rather than on the application record format.
- If the CMD12 key is pressed, control is returned to the application program and data from the application record format is transmitted to the application program. Response indicator 12 is on.
- If the CMD5 key is pressed, it functions as the Enter key. The same CA or CF keys must be specified on both the application record format and the application help record format for control to be returned to the application program.
- If the CMD3 key is pressed, it functions as the Enter key. The corresponding CA or CF keys must be specified on both the application help record format and the application record format for control to be returned to the program.
- If the CMD4 key is pressed, it results in a message indicating that the CMD4 key is not allowed. The same CA or CF keys must be specified on both the application record format and the application help record format in order for control to be returned to the application program.

HLPDOC (Help Document) keyword for display files

You use this file- or help-specification-level keyword to define the document as help information text for a specific location on the display.

The format of the keyword is:

```
HLPDOC(online-help-information-text-label-name document-name folder-name)
```

The online-help-information-text-label-name parameter corresponds to a label in the online help document and marks the point at which the document's display begins.

The document-name parameter identifies the online document containing the online help information.

The folder-name parameter identifies the folder containing the document specified. Because folders can reside inside other folders, and because any given folder or document name is only unique within its containing folder, you might be required to concatenate several folder names together to identify a document/folder. The folder name you specify on the HLPDOC keyword can be a simple folder name, which follows the same syntax rules as the document name, or you can specify the folder name as a concatenated name.

The document specified on an H specification level HLPDOC keyword is displayed if both of the following conditions are true:

- The cursor is located in the help area (defined by HLPARA) for that H specification.
- The H specification is active. (The option indicator on the H specification level HLPDOC keyword determines if the H specification is active.)

The document specified on a file level HLPDOC keyword is displayed when no help area for the active records contains the current cursor location.

You cannot specify HLPDOC with HLPBDY, HLPPNLGRP, or HLPRTN.

Option indicators are valid for this keyword.

Example

The following example shows how to specify the HLPDOC keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A                                     HELP
     A                                     HLPDOC(START GENERAL.HLP HELP.F1)
     A          R REC1                     OVERLAY
     A          H                          HLPDOC(LBL1 HELP#1 HELP.F1)
     A                                     HLPARA(10 3 12 50)
     A          H                          HLPARA(15 9 17 61)
     A  90                                 HLPDOC(LBL2 HELP#2 HELP/FLD)
     A          H                          HLPARA(15 9 17 61)
     A N90                                 HLPDOC(LBL3 HELP#3 HELP.F1/FLD)
     A
```

The HELP keyword enables the Help key.

The HLPDOC keyword at the file level specifies that the GENERAL.HLP document in the HELP.F1 folder will be displayed starting at the START help label if the Help key is pressed and the cursor is not in a help area defined by a HLPARA keyword at the H specification level.

At the H specification level:

- The first H specification indicates that the HELP#1 document in the HELP.F1 folder will be displayed starting at the LBL1 help label if the Help key is pressed and the cursor is in positions 3 through 50 of lines 10, 11, or 12.
- The second H specification indicates that the HELP#2 document in the HELP/FLD folder will be displayed starting at the LBL2 help label if the REC1 record is put with indicator 90 on, the Help key is pressed, and the cursor is in positions 9 through 61 of lines 15, 16, or 17.
- The third H specification indicates that the HELP#3 document in the HELP.F1/FLD folder will be displayed starting at the LBL3 help label if the REC1 record is put with indicator 90 off, the Help key is pressed, and the cursor is in positions 9 through 61 of lines 15, 16, or 17.

Related reference

DDS naming conventions

HLPEXCLD (Help Excluded) keyword for display files

You use this help-specification-level keyword to indicate that the online help information associated with this help specification is not displayed as extended help, but is available as item-specific help.

This keyword has no parameters.

If you do not specify this keyword, extended help consists of the online help information associated with both the file-level HLPPNLGRP keyword (if any) and the HLPPNLGRP keyword on all active help specifications.

This keyword is allowed only on help-specifications that specify a HLPPNLGRP keyword.

At least one instance of each parameter on the HLPPNLGRP keyword should not have the HLPEXCLD keyword specified. If all help specifications specifying a particular help panel group name are excluded, an error message is issued at run time if the Help key is pressed when the cursor is located in a help area associated with that help panel group name.

Option indicators are valid for this keyword.

Example

The following example shows how to specify the HLPEXCLD keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A          R RECORD1
     A
     A          H                           HLPARA(1   12   8   14)
     A                                      HLPPNLGRP(R1  PNLA)
     A          H                           HLPARA(1   18   8   19)
     A                                      HLPPNLGRP(R2  PNLA)
     A          H                           HLPARA(1   35   8   37)
     A                                      HLPPNLGRP(R1  PNLA)
     A                                      HLPEXCLD
     A          H                           HLPARA(1   49   8   50)
     A                                      HLPPNLGRP(R2  PNLA)
     A                                      HLPEXCLD
     A
```

In this example, the HLPEXCLD keywords prevent help modules R1 and R2 from being displayed twice as extended help.

HLPFULL (Help Full) keyword for display files

You use this file-level keyword to indicate that the help text for the help panel group of the application is displayed using the full screen rather than using windows.

This keyword has no parameters.

If you do not specify this keyword, the online help information is displayed in a window unless the *HLPFULL option is specified for the user profile.

When you specify the HLPFULL keyword, you must specify the HLPPNLGRP keyword either at the file level or at the help specification level.

Option indicators are not valid for this keyword.

Example

The following example shows how to specify the HLPFULL keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A                                      HELP
     A                                      HLPPNLGRP(GENERAL LIBA/PNL1)
     A                                      HLPFULL
     A                                      HLPTITLE('Sample Screen')
     A          R RECOO1H
     A          H                           HLPARA(4  10  4  29)
     A                                      HLPPNLGRP(NAMETAG LIBA/PNL1)
     A                                  1 10'Sample Screen'
```

```
     A            NAME1         20A  B  2 10
     A
```

When the Help key is pressed in cursor location line 4 positions 10 through 29, the help module NAMETAG from LIBA/PNL1 is displayed using the full screen. If the Help key is pressed in any other location, the help module GENERAL from LIBA/PNL1 is displayed using the full screen.

HLPID (Help Identifier) keyword for display files

You use this constant field-level keyword to specify an identifier for the constant in the field-level help. The identifier you specify can be used on the HLPARA keyword to link help text to this constant field.

The format of the keyword is:

```
HLPID(help-identifier)
```

The help-identifier parameter is required and can be only a numeric value from 1 to 999. The value you specify must be unique within the record you are defining.

Option indicators are not valid for this keyword.

Example

The following example shows how to specify the HLPID keyword:

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A          R RECORD                    HELP
     A          H                           HLPARA(*CNST 1)
     A                                      HLPRCD(HLPCNST1 LIB1/FILE1)
     A          H                           HLPARA(*CNST 2)
     A                                      HLPRCD(HLPCNST2 LIB1/FILE1)
     A                                  2  4'Constant field 1' HLPID(1)
     A                                  4  4'Constant field 2' HLPID(2)
     A
```

In this example, if the cursor is located on the text `'Constant field 1'` on the display and the Help key is pressed, record HLPCNST1 in file FILE1, library LIB1, is displayed as help text. If the cursor is located on the text `'Constant field 2'` on the display and the Help key is pressed, record HLPCNST2 in file FILE1, library LIB1 is displayed as help text.

HLPPNLGRP (Help Panel Group) keyword for display files

You use this file-level or help-specification-level keyword to specify the source of the online help information, defined by the user interface manager (UIM), that is to be displayed when the Help key is pressed.

The format of the keyword is:

```
HLPPNLGRP(help-module-name [library-name/]panel-group-name)
```

The help module name is 1 to 32 characters long. Valid values for the first character in the name are alphabetic characters A through Z. Valid values for subsequent characters are alphabetic characters A through Z, numeric characters 0 through 9, slash (/), and underscore (_). If the name contains a slash or underscore character, the name must be enclosed in single quotation marks.

The panel group name specifies the UIM-panel group object that contains the help module. It need not exist when the display file is created. If you do not specify the library name, *LIBL is used to search for the panel group object.

Every help specification must contain a HLPRCD, HLPDOC, or HLPPNLGRP keyword, but a display file cannot contain both HLPPNLGRP and HLPRCD keywords, nor HLPPNLGRP and HLPDOC keywords.

If you specify the HLPPNLGRP keyword at the file level, you must specify the HELP keyword at the file level. If there are no help specifications in the file, you must also specify the HLPTITLE keyword at the file level.

If you specify the HLPPNLGRP keyword at the help specification level, you must specify a HELP and a HLPTITLE keyword either at the file level or on the current record.

Option indicators are valid for this keyword.

Example

The following example shows how to specify the HLPPNLGRP keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A                                     HELP
     A                                     HLPPNLGRP(GENERAL LIBA/PANEL1)
     A                                     HLPTITLE('Sample Screen')
     A                                     HLPSCHIDX(QHSS1)
     A          R REC001
     A          H                          HLPARA(4  10  4  29)
     A                                     HLPPNLGRP(NAMETAG LIBA/PANEL1)
     A          H                          HLPARA(5  10  5 19)
     A                                     HLPPNLGRP(OPTION1TAG PANEL2)
     A                                  1 10'Sample Screen'
     A            NAME1         20A  B  4 10
     A            OPTION1       10A  B  5 10
     A
```

If the cursor is located on line 4 between positions 10 and 29 when the Help key is pressed, help module NAMETAG from UIM panel group PANEL1 in LIBA is displayed. If the cursor is located on line 5 between positions 10 and 19, help module OPTION1TAG from UIM panel group PANEL2 in the library list is displayed. If the cursor is located anywhere else when the Help key is pressed, help module GENERAL in panel group PANEL1 in LIBA is displayed.

HLPRCD (Help Record) keyword for display files

You use this file-level or help-specification-level keyword to specify the record format containing the online help information to be displayed when the Help key is pressed.

The format of the keyword is:

```
HLPRCD(record-format-name [[library-name/]file-name])
```

The record format can exist either in the file being defined or in the file specified on HLPRCD. If you do not specify the file name, the record format must exist in the file being defined.

The file-name parameter identifies the file containing the record format. The current library list (*LIBL) at program run time is used if you do not specify the library name.

The record specified on an H-specification-level HLPRCD keyword is displayed if both of the following conditions are true:

- The cursor is located in the help area (defined by HLPARA) for that H specification.
- The H specification is active. (The option indicator on the H-specification-level HLPRCD keyword determines whether the H specification is active.)

The record specified on a file-level HLPRCD keyword is displayed when no help area for the active records contains the current cursor location.

Option indicators are valid for this keyword.

Example

The following example shows how to specify the HLPRCD keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A                                     HELP
     A                                     HLPRCD(DFTHELP HELPFILE)
     A          R RECORD1
     A          H                          HLPARA(1 1 24 80)
     A  99                                 HLPRCD(ERRHELP)
     A          H                          HLPARA(1 1 1 80)
```

```
     A                                     HLPRCD(HELPRCD1 HELPFILE)
     A            FIELD1        10A     1 10
     A
```

When indicator 99 is set on, the online help information in the record ERRHELP (which must exist in this display file) displays when the Help key is pressed. If indicator 99 is set off and the cursor is located on the first line when the Help key is pressed, the record HELPRCD1 in file HELPFILE will display. Otherwise, the online help information from the record specified on the file-level HLPRCD keyword will display when the Help key is pressed.

HLPRTN (Help Return) keyword for display files

You use this file-level or record-level keyword to return control to your program when you press the Help key.

If HLPRTN is not specified, online help information associated with the current cursor location is displayed.

See “System/36 environment considerations for display files” on page 247 for special considerations when you specify the HLPRTN keyword in files that are used in the System/36 environment.

The format of the keyword is:

```
HLPRTN[(response-indicator ['text'])]
```

If you specify a response indicator, the response indicator is set on and returned to your program. No input data is transmitted from the device. Processing is similar to that of a command attention key.

The optional text is included on the computer printout created at program compilation time to explain the intended use of the indicator. This text has no function in the file or program other than as a comment. The single quotation marks are required. If you specify more than 50 characters between the single quotation marks, the text is truncated to 50 characters on the program computer printout.

HLPRTN at either the file or record level takes priority over any HLPRCD, HLPPNLGRP, or HLPDOC keywords. Any HLPRTN keyword found in the file is processed before any other applicable help keyword.

When you specify HLPRTN, control might or might not return to your program, depending on whether you use an option indicator:

- If you specify HLPRTN without an option indicator, control returns to your program when you press the Help key. A warning message appears at creation time if you specify an unoptioned HLPRTN keyword on a file or record containing H specifications.
- If you specify HLPRTN with an option indicator, control returns to your program if the option indicator is on at the time the record is displayed. The H specifications are used if the option indicator is off.

Option indicators are valid for this keyword.

Example 1

The following example shows how to use the HLPRTN keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A                                     HELP
     A                                     HLPRCD(GENERAL)
     A          R RECORD1
     A  02                                 HLPRTN
     A          H                          HLPARA(1 1 3 10)
     A                                     HLPRCD(HELPREC1)
     A            FIELD1        10A  B  2  2
     A          R RECORD2                  HLPRTN
     A            FIELDA         5A  B 10  7
     A
```

If indicator 02 is on when RECORD1 is written to the display, control will return to the user program when the Help key is pressed. If indicator 02 is off, online help information record HELPREC1 or GENERAL will

be displayed when the Help key is pressed, depending on the position of the cursor. When RECORD2 is displayed, control will return to the user program when the Help key is pressed.

Example 2

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A                                     HELP
     A  01                                 HLPRTN
     A                                     HLPRCD(GENERAL)
     A          R RECORD1
     A          H                          HLPARA(1 1 3 10)
     A                                     HLPRCD(HELPREC1)
     A            FIELD1        10A  B  2  2
     A          R RECORD2
     A            FIELDA         5A  B 10  7
     A
```

If indicator 01 is on, control returns to your program when you press the Help key regardless of the cursor position.

Related reference

HELP (Help) keyword for display files
You use this file-level or record-level keyword to enable the Help key.

HLPARA (Help Area) keyword for display files
You use this help-specification-level keyword to define a rectangular area on the display.

HLPSCHIDX (Help Search Index) keyword for display files

You use this file-level keyword to enable the index search function (F11 on the Help display) and specify the search index object used for the index search.

The format of the keyword is:

```
HLPSCHIDX([library-name/]search-index-object)
```

The search index object, created using the CRTSCHIDX command, contains the data to be made available when the function key is pressed to start the index search function.

If you do not specify a library name, *LIBL is used to search for the search index object. The search index object need not exist when the display file is created.

HLPSCHIDX is valid only when at least one HLPPNLGRP keyword is specified in the file.

HLPSCHIDX keyword cannot be specified with the HLPSHELF keyword.

Option indicators are not valid for this keyword.

Example

The following example shows how to specify the HLPSCHIDX keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A                                     HELP
     A                                     HLPTITLE('Sample Screen')
     A                                     HLPPNLGRP(GENERAL LIBA/PANEL1)
     A                                     HLPSCHIDX(LIBA/SEARCH1)
     A          R REC001H
     A          H                          HLPARA(4  10  4  29)
     A                                     HLPPNLGRP(NAMETAG LIBA/PNL1)
     A                                  1 10'Sample Screen'
     A            NAME1         20A  B  4 10
     A
```

HLPSEQ (Help Sequencing) keyword for display files

You use this record-level keyword to define the sequencing of text records for Page key processing.

The format of the keyword is:

```
HLPSEQ(group-name sequence-number)
```

The group name is a 1- to 10-character name used to associate the primary help format with the secondary help formats in the help file. When a Page Up or Page Down key is pressed on an online help information display, those record formats in the help file that have the same help group name as the record currently displayed as online help information are displayed.

The sequence number is a numeric value (0 to 99) used to order the record formats within the help group. This order determines the sequence in which the record formats are displayed as secondary online help information.

Duplicate numbers within a group are not allowed.

A help record format that does not have the HLPSEQ keyword coded is considered to be the only record in its group.

You cannot specify HLPSEQ on subfile (SFL keyword) or user-defined (USRDFN keyword) record formats.

Option indicators are not valid for this keyword.

Example

The following example shows how to specify the HLPSEQ keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A          R RECORD1                  HLPSEQ(HGROUP1 10)
     A                                5  1'Help text ...'
     A
```

RECORD1 is in the help group HGROUP1 and has a sequence number of 10.

HLPTITLE (Help Title) keyword for display files

You use this file-level or record-level keyword to define the default title of online help information in panel group. The help information is displayed using the full screen.

This should be the name of the display you were on when you pressed the Help key. Use this keyword only on a full-screen help display and when no help title is specified in the help source.

The format of the keyword is:

```
HLPTITLE('text')
```

The text can be up to 55 characters long.

If you specify the HLPTITLE keyword in a file, the file must contain at least one HLPPNLGRP keyword at either the file or help specification level.

If you specify a file-level HLPPNLGRP keyword and no help specifications are defined in the file, the HLPTITLE keyword is required at the file level.

If you do not specify a HLPTITLE keyword at the file level, at least one HLPTITLE keyword is required on every record that contains help specifications. The HLPTITLE keyword is not valid on records that do not contain help specifications.

Option indicators are not valid on a file-level HLPTITLE keyword. Option indicators are allowed on record-level HLPTITLE keywords and must be specified on each HLPTITLE keyword if the record contains multiple HLPTITLE keywords. You can specify a maximum of 15 HLPTITLE keywords on a record if all have option indicators. At run time, the first HLPTITLE keyword in effect is used. If no HLPTITLE keyword is in effect for the record, a message is issued.

Example

The following example shows how to specify the HLPTITLE keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A                                     HELP
     A                                     HLPPNLGRP(GENERAL LIBA/PANEL1)
     A          R REC001
     A  90                                 HLPTITLE('Sample Screen 1')
     A N90                                 HLPTITLE('Sample Screen 2')
     A          H                          HLPARA(4  10  4  29)
     A                                     HLPPNLGRP(NAMETAG LIBA/PANEL1)
     A          H                          HLPARA(6  10  6  19)
     A  10                                 HLPPNLGRP(OPTION2TAG PANEL2)
     A  90                              1 10'Sample Screen 1'
     A N90                              1 10'Sample Screen 2'
     A            NAME1         20A  B  4 10
     A  10        OPTION2       10A  B  6 10
     A
```

Two titles are associated with the record, so the HLPTITLE keyword is specified at the record level, where option indicators are used. The state of option indicator 90 determines which title is displayed on both the application display and the online help information display. Using an indicator and its complement guarantees that one of the two HLPTITLE keywords is in effect.

HOME (Home) keyword for display files

You use this file-level or record-level keyword to specify that you want to recognize and handle the Home key through your program.

See “System/36 environment considerations for display files” on page 247 for special considerations when you specify the HOME keyword in files that are used in the System/36 environment.

The format of the keyword is:

```
HOME[(response-indicator ['text'])]
```

If you press the Home key and the cursor is not already at the home position, the cursor returns to the home position, even if you do not specify the HOME keyword.

If the cursor is already at the home position when the Home key is pressed, the IBM i operating system returns control to your program as it does when a command attention key is pressed (no data is received from the device). In this situation, if you have not specified the HOME keyword, the IBM i operating system sends a message indicating that the key is not valid at that time.

The home position is one of the following positions (in order of priority):

- The cursor position specified by the last output operation
- The first unprotected input field
- Position 1, line 1

The optional text is included on the computer printout created at program compilation time to explain the intended use of the indicator. This text's only function in the file or the program is as a comment. The single quotation marks are required. If you specify more than 50 characters between the single quotation marks, the text is truncated to 50 characters on the program computer printout.

Option indicators are valid for this keyword.

Example

The following example shows how to specify the HOME keyword so that if the cursor is in the home position when the Home key is pressed, control returns to the program with response indicator 95 set on.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00030A                                 HOME(95 'Home key')
     A
```

HTML (Hypertext Markup Language) keyword for display files

You use this field-level keyword on an unnamed constant field to have Hypertext Markup Language (HTML) tags sent along with the 5250 data stream.

If the data stream is sent to a 5250 Workstation Gateway device, the HTML tags will be processed on the HTML browser. If the data stream is not sent to a 5250 Workstation Gateway device, the HTML keyword is ignored.

The format of this keyword is:

```
HTML('value')
```

or

```
HTML(&program-to-system-field);
```

A parameter is required for the HTML keyword. The parameter must be a valid HTML tag enclosed in single quotation marks, or in a program-to-system field. The program-to-system field can be any legal length and must be alphanumeric (A in position 35). The DDS compiler will not check the HTML syntax of the specified parameter. The browser that receives the HTML at run time will check the syntax.

The following keywords are not allowed with the HTML keyword:

COLOR
DATE
DFT
DSPATR
EDTCDE
EDTWRD
HLPID
MSGCON
NOCCSID
OVRATR
PUTRETAIN
SYSNAME
TIME
USER

Option indicators are not valid for this keyword. However, option indicators are allowed on the constant field.

The HTML keyword is not allowed in a field of a subfile record.

Example

The following example shows how to specify the HTML keyword:

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8

     A
     A          R RECORD
     A                                  7 20HTML('<TITLE>')
     A                                  7 20HTML(&TAG);
     A                                  7 21HTML('</TITLE>')
     A            TAG           20A  P
```

HTML is a tag language where the order of the tags determines when they are processed. Row and column have no meaning in an HTML document. In this case, the row and column determine the order in which the HTML tags are sent to the browser. If the constant fields that contain the HTML keyword have the same row and column value, they will be processed in the order that they appear in the DDS source. For information about how to resolve HTML field overlap, see Chapter 6 of the Application Display Programming book.

If the ENHDSP (Enhanced Display) parameter on the CRTDSPF is set to *NO, the HTML keyword will be ignored. This will give users the ability to turn off the HTML keywords without recompiling.

INDARA (Indicator Area) keyword for display files

You use this file-level keyword to remove option and response indicators from the buffer (also called the record area) and to place them in a 99-byte separate indicator area.

This keyword has no parameters.

Specifying INDARA provides the following advantages:

- Simplifies COBOL programming when both option and response indicators are used. If the same indicator is used as a response indicator and as an option indicator, both indicators always have the same value, regardless of the order in which they are specified in the DDS.
- Assists the RPG programmer using program-described WORKSTN files.

If you specify INDARA, you can add, change, or delete option and response indicators in the DDS and re-create the file without re-creating the high-level language program. This is true because the field locations in the buffer have not changed and therefore the level check data has not changed. However, if the program is to take advantage of the new indicators, it still needs to be changed and re-created.

If you specify INDARA, some high-level languages might require that you specify in your program that a separate indicator area is to be used. See the appropriate high-level language manual.

If you specify INDARA, the LOGINP and LOGOUT keywords do not log response or option indicators when your program sends I/O operations. This is because response and option indicators do not appear in the input or output buffers.

Option indicators are not valid for this keyword.

Example

The following example shows how to specify the INDARA keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A                                      INDARA
00020A                                      CF01(01 'End program')
00030A          R PROMPT
00040A            ACTNBR        10   B  2  2
00050A  41                                  ERRMSG('Account number +
00060A                                      not found' 41)
     A
```

INDARA has been specified; option indicator 41 and response indicators 01 and 41 are removed from the buffer for record format PROMPT and placed in the separate indicator area. Only ACTNBR, a named input/output field, remains in the buffer for record format PROMPT.

INDTXT (Indicator Text) keyword for display files

You use this file-level, record-level, or field-level keyword to associate a descriptive text (indicating intent or use) with a specific response or option indicator. You can specify the keyword once for each response and option indicator.

The format for the keyword is:

```
INDTXT(indicator 'indicator-text')
```

If you specify this keyword, indicator-text is a required parameter value. The text is included on the computer printout created at program compilation time to explain the intended use of the indicator. This text's only function in the file or the program is a comment. The single quotation marks are required. If you specify more than 50 characters between the single quotation marks, the text is truncated to 50 characters on the program computer printout.

Option indicators are not valid for this keyword.

Note: The INDTXT keyword by itself does not cause the specified indicator to appear in either the input or the output record area. It merely provides text to be associated with the indicator. If the indicator has not been specified elsewhere (as either an option indicator or a response indicator), then the text is lost without a diagnostic message. When an indicator has been given a text assignment (either by this keyword or the response indicator text), no other text assignment is allowed.

Example

The following example shows how to specify the INDTXT keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A                                      INDTXT(02 'Alternate month')
00020A          R MASTER
00030A            MTH                   2 10
00040A  02        ALTMTH                2 10
     A
```

INDTXT describes the use of option indicator 02. In a compiler computer printout for a high-level language, 'Alternate month' will be printed as a comment with the description of indicator 02.

INVITE (Invite) keyword for display files

You use this file-level or record-level keyword to invite the device for a later read operation.

To send an invite to a specific device, your program sends an output operation to the device with the INVITE keyword in effect. If the record format used has output-capable fields, the data is sent to the device before the device is invited.

This keyword has no parameters.

INVITE must be used if the display file can have multiple acquired devices and your program does read from invited devices operations. This is because a read from invited devices to a multiple device display file only returns a record from one of the devices that was invited. If you want all the acquired devices to be able to return data, an output operation with INVITE in effect must be sent to each device before the read from invited devices. Even if there is only one device acquired on the multiple device file, the device must be invited via INVITE before a read from invited devices.

INVITE also gives you the ability to create a subset of acquired devices that are eligible to respond on a read from invited devices. For example, if ten devices are currently acquired to the file but only three devices were invited, the next read from invited devices operation returns a record from one of the three invited devices. This is true even if the other devices have data available.

INVITE provides some performance improvement. Normally a read request is issued to a device when your program sends an input operation. However, INVITE allows you to request the read when you issue the output operation. After the output operation is complete, your program can do other processing while the device is issuing data and the IBM i operating system is processing the received data. This can be a significant improvement if the device is remote.

For specific instructions on when an invite operation is necessary and how to specify a read from invited devices operation in your program, see the appropriate high-level language manual.

INVITE cannot be specified at both the file and record level and cannot be specified with the subfile keyword (SFL).

Option indicators are valid for this keyword.

Special considerations when using the INVITE keyword

The following list shows special considerations when you use the INVITE keyword:

- An input operation sent to a specific device does not require an invite operation. Input operations with a specified record format name or device are directed to one device. If that device has an invite outstanding at the time of the input operation, the invite operation is deleted after the input operation is completed.

- After an invite operation has been sent to a device, the only valid operations (in addition to a read-operation-from-invited-devices operation) are these:
  - An input operation to a specific device.
  - An output operation with data that tries to cancel the invite operation. If the cancel operation is successful, the data is written. If the INVITE keyword is in effect on the output operation, the device is invited again. If the cancel operation is not successful (because the data has already been received by the system), the output operation fails. Your program must perform an input operation to process the data. The input operation erases the invite operation for that device.
- On a read operation from invited devices operation to a display file, only data from devices with an outstanding invite operation are considered. The input operation waits for data from any of the invited devices. (See the WAITRCD parameter on the Create Display File (CRTDSPF) and Change Display File (CHGDSPF) commands.) If none of the invited devices responds before the wait time ends, a notify message is sent and no data is returned. All invited devices remain invited.
- If more than one device obtained by the display file has an invite operation outstanding, a read-operation-from-invited-devices operation returns the next available record from one of the invited devices. If records are received from more than one device before this input operation, the other records are kept for subsequent input operations.
- When a read operation from invited devices operation to a display file returns a record to your program from an invited device, the invite operation for that device is deleted. Other devices that have an invite outstanding remain invited. The device that sent the record your program read must be invited again if you want to receive data from this device on a later read from invited devices operation.
- If no device was invited or if a device was invited but the job was canceled with the controlled option, a read operation from invited devices operation to a display file results in a notify message and no data is returned to your program. All invited devices remain invited.
- If you want to invite a device but have no data to send it, issue an output operation using a record format containing no output-capable fields with INVITE in effect.
- After the first record is received from an invited display device, the device should not be re-invited until all the record formats on the display with input-capable fields are read by your program. Your program can read those other record formats if you specify the record format name and the device name on the read operation.
- If your display file has the delayed write option (DFRWRT(*YES) parameter on the (Create Display File (CRTDSPF) and Change Display File (CHGDSPF) commands), an output operation with the INVITE keyword in effect causes the delayed output to appear on the display before the device is invited.

Example

The following example shows how to specify the INVITE keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A  01                                INVITE
00020A          R RCD1
00030A            FLD1          10      2  2
00030A            FLD2           5      2 24
     A
```

INVITE is in effect only when option indicator 01 is set on.

INZINP (Initialize Input) keyword for display files

You use this record-level keyword to initialize input/output fields without necessarily sending the initialized data to the display when the PUTOVR and ERASEINP(*ALL) keywords are both in effect.

This keyword has no parameters.

INZINP is particularly useful for applications that perform data entry from remote workstations. It can reduce line traffic between the system unit and the remote workstation.

The following steps describe how this keyword is used:

1. An output operation displays an input/output field for which the OVRDTA or OVRATR keyword is in effect. The system initializes the input save area to the program value of the field. For example, if the program sets NAME1 to the name Bob, the input save area contains the name Bob.
2. An input operation places data from the workstation in an input buffer. If the workstation user did not key into the input-capable field and the DSPATR(MDT) keyword is not in effect for the field, the field does not return data from the workstation. The system retrieves data from the input save area and places it in the input buffer for use by the program. Thus, all input-capable fields have data in the input buffer, either from the workstation or from the input save area.
3. On another output operation, the following cases might occur:
   - If the INZINP and OVRDTA keywords are not in effect, the input save area is unchanged, even if the program changed the field value. For example, if the program changes the field value to the name Tom, the input save area will still have either the value entered by the workstation user or the name Bob, the earlier program value.
   - If INZINP is in effect, the input save area will have the current program value. The current program value is sent to the display for fields for which the OVRDTA keyword is in effect.

     If the OVRDTA keyword is not in effect on an output operation, the program must clear the output buffer for the fields with the OVRDTA keyword specified to ensure that the input save area matches the fields on the display (which are all blanks after the output operation).

Note: If the ERASEINP(*ALL) keyword is in effect for the output operation, input-capable fields are cleared at the display device (on the display), but the input save area is not cleared. For the contents of the input save area, see Table 8 on page 138 and Table 9 on page 138. Fields with the DFT keyword specified are initialized with the value specified for the DFT keyword even if the fields are not selected for display. The values are maintained unless the application program selects the fields for display, then changes their data values. This keyword requires the PUTOVR, OVERLAY, and ERASEINP(*ALL) keywords to be specified at the record level.

The following tables show the effect of the ERASEINP(*ALL) and PUTOVR keywords with and without the INZINP, OVRATR or OVRDTA keywords.

Table 8. INZINP Input/output fields

OVRATR or OVRDTA keyword	INZINP keyword	Contents of input save area
Not specified	Does not apply	Previous contents
Specified but not in effect	Specified but not in effect	Previous contents
Specified but not in effect	Specified and in effect	Program value (not sent to display)
OVRATR specified and in effect	Does not apply	Program value (also sent to display)
OVRATR specified and in effect; OVRDTA not specified or not in effect	Does not apply	Previous contents (not sent to display)

Table 9. INZINP input-only fields

OVRATR keyword	INZINP keyword	Contents of input save area
Not specified	Does not apply	Previous contents
Specified and in effect	Does not apply	Previous contents
Specified but not in effect	Does not apply	Character fields: blanks Numeric fields: zeros

ERASEINP(*ALL) keyword

You can set the input save area to blanks and zeros to match the fields cleared at the workstation by the ERASEINP(*ALL) keyword.

About this task

Complete the following tasks:

Procedure

1. Specify the same option indicators for INZINP as for ERASEINP(*ALL), PUTOVR, and OVERLAY keywords.
2. Specify OVRDTA or OVRATR for all input/output fields.

   Set option indicators off for these keywords if you do not want to send data or attributes to the device. If you enable OVRATR, also enable OVRDTA.
3. Specify the OVRATR keyword for all input-only fields.

   Set option indicators off for this keyword if you do not want to send attributes to the device.
4. Set all input/output fields to blanks (for character fields) or zeros (for numeric fields) before the output operation.

Results

A warning message appears at file creation time if the INZINP keyword is specified on a record with the DSPMOD keyword. At run time, the INZINP keyword is ignored when the display mode changes.

Option indicators are valid for this keyword.

The following example shows how to specify the INZINP keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A          R FMT1
00020A  77                                  PUTOVR OVERLAY ERASEINP(*ALL)
00030A  77                                  INZINP
00040A                                    7  8'CUSTOMER NUMBER'
00050A            CUSNBR         6   I  7 26
00060A N77                                  OVRATR
00070A                                    9 12'CUSTOMER NAME'
00080A            NAME          25   B  9 30
00090A N77                                  OVRATR
00100A                                   10  2'CUSTOMER ADDRESS LINE 1'
00110A            ADDR1         25   B 10 30
00120A N77                                  OVRATR
00130A                                   11  2'CUSTOMER ADDRESS LINE 2'
00140A            ADDR2         25   B 11 30
00150A N77                                  OVRATR
00160A                                   12  9'NEW CREDIT LIMIT'
00170A            LIMIT          4  0I 12 30
00180A N77                                  OVRATR
     A
```

Example details

This example illustrates the following points:

1. For the first output operation, the user program sets off option indicator 77. Therefore, the PUTOVR, ERASEINP, and INZINP keywords are not in effect. This has the following effect:
   - Fields NAME, ADDR1, and ADDR2 are sent to the workstation.
   - The input save area contains:

     CUSNBR
     All blanks

NAME
The program value (which appears on the display)

ADDR1
The program value (which appears on the display)

ADDR2
The program value (which appears on the display)

LIMIT
All zeros

2. For the second output operation, the user program sets on option indicator 77. This has the following effect:
   - All input-capable fields are cleared at the workstation.
   - No fields are sent to the workstation.
   - The input save area contains:

     CUSNBR
     All blanks

     NAME
     The program value (not sent to the display)

     ADDR1
     The program value (not sent to the display)

     ADDR2
     The program value (not sent to the display)

     LIMIT
     All zeros

Note: If fields NAME, ADDR1, and ADDR2 had been set to blanks before this second output operation, the input save area would contain all blanks and zeros.

INZRCD (Initialize Record) keyword for display files

You use this record-level keyword to specify that if this record is not already on the display, it is to be written to the display before an input operation is sent from the program that specifies this record name.

If this record is already on the display, the keyword is ignored. The performance of this implicit output operation is IBM i program-initiated; its only purpose is to format the display when an input operation is performed.

This keyword has no parameters.

This keyword does not apply to output operations.

If the INZRCD keyword is not specified, your program receives an error if it tries to read a record that is not on the display.

When the INZRCD keyword is processed, the following special conditions exist:

- For output-only fields, no user data is available. The field appears on the display as blanks. Any editing specified is ignored. The BLKFOLD keyword does not affect the display.
- For input/output fields, no user data is available. The field appears on the display as blanks. The input save area is initialized in the same way as uninitialized input-only fields (blanks or zeros, depending on the data type).
- Constants and input-only fields appear the same as when displayed using an explicit output operation.
- Hidden fields are returned on an input operation as blanks or zeros.
- Message and program-to-system fields are ignored.
- The LOGOUT keyword is ignored because there is no output buffer to log.

- The ERRMSG and ERRMSGID keywords are ignored because the record format is not already on the display.
- The SFLMSG and SFLMSGID keywords are ignored.
- All other optioned keywords and fields are processed as if they were optioned.

Note: In order for the INZRCD function to be performed, your program must specify a record format name when issuing an input operation that contains this keyword. The record format used for the input operation must specify the INZRCD keyword.

Option indicators are not valid for this keyword.

Example

The following example shows how to specify the INZRCD keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00274A          R REC4                    INZRCD
     A
```

KEEP (Keep) keyword for display files

You use this record-level keyword to keep the display from being deleted when the display file is closed.

The entire display is kept if any of the records on the display have KEEP specified. The default causes the entire display to be deleted when the file is closed.

In addition, the name of the first, uppermost record on the display that has the KEEP attribute is saved by the IBM i operating system for possible use by subsequent programs. The name kept can be used by a subsequent program that does not specify a record name on its first input operation. This keyword enables you to leave data on the display for review after your program ends, or use that data as input for subsequent programs.

This keyword has no parameters.

This keyword cannot be specified with the following keywords:

ALWROL
CLRL
SLNO

A warning message appears at file creation time if the KEEP keyword is specified on a record with the DSPMOD keyword. At run time, the KEEP keyword is ignored when the display mode changes.

Option and response indicators are not valid for this keyword.

Example

The following example shows how to specify the KEEP keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00061A          R REC46                   KEEP
     A
```

LOCK (Lock) keyword for display files

You use this record-level keyword to specify that the keyboard is to remain locked after an output operation.

Normally, the keyboard automatically unlocks after an output operation. The LOCK keyword is used when there are several consecutive output operations that contain input fields.

This keyword has no parameters.

If this keyword is not specified, the workstation user can type data into a field when a subsequent output operation sends data to the display. In this case, the cursor location might be changed and the key entry data lost.

Note: The default on an output operation is to unlock the keyboard. If the keyboard is locked when an input operation is sent, it is automatically unlocked.

This keyword is independent of other keywords that affect the output operation.

Option indicators are valid for this keyword.

Example

The following example shows how to specify the LOCK keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00011A          R REC1                      LOCK
     A
```

LOGINP (Log Input) keyword for display files

You use this record-level keyword to specify that the input buffer for this record format is to be written to the job log each time the IBM i operating system performs an input operation for the record.

The data logged includes the values of input-capable fields, hidden fields, and response indicators specified in the record format you are defining. It also includes output fields if the record format is a subfile record format. (Response indicators are not logged if the INDARA keyword is specified for the file you are defining.) This keyword is used for debugging and other exception conditions. The job log cannot be read by your program.

This keyword has no parameters.

The IBM i operating system ignores LOGINP for either of the following conditions:

- There are no input-capable fields, no hidden fields, and no response indicators in the record format.
- The record format is a subfile record format for a message subfile.

Option indicators are not valid for this keyword.

Example

The following example shows how to specify the LOGINP keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00072A          R REC24                     LOGINP
     A
```

LOGOUT (Log Output) keyword for display files

You use this record-level keyword to specify that the output buffer for this record format is to be written to the job log each time the IBM i operating system performs an output operation for the record.

The data logged includes the values of output-capable fields, hidden fields, and option indicators specified in the record format you are defining. (Option indicators are not logged if the INDARA keyword is specified for the file you are defining.) The LOGOUT keyword is used for debugging and other exception conditions. The job log cannot be read by your program.

This keyword has no parameters.

The IBM i operating system ignores LOGOUT for either of the following conditions.

- There are no output-capable fields, no hidden fields, and no option indicators in the record format.
- The record format is a subfile record format for a message subfile.

Option indicators are valid for this keyword.

Example

The following example shows how to specify the LOGOUT keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00021A          R REC25                    LOGOUT
     A
```

LOWER (Lower) keyword for display files

The LOWER keyword is equivalent to the CHECK(LC) keyword.

The CHECK keyword is preferred.

Related reference

CHECK (Check) keyword for display files
You use this keyword to perform a number of functions, depending on the parameter values specified.

MAPVAL (Map Values) keyword for display files

You use this field-level keyword to map field data to a different value during input and output operations.

This keyword is only valid with the date (L), time (T), or timestamp (Z) data types.

The format of the keyword is:

```
MAPVAL((program-value-1 system-value-1)
      [(program-value-2 system-value-2) ...
       (program-value-100 system-value-100)])]
```

You can specify the program value and the system value either as explicit values or as the *BLANK or *CUR special values.

You must specify an explicit value within single quotation marks that matches the format and separator values for the corresponding field. The explicit value must also be a valid date or time. The following list indicates how the formats and separators are determined:

- If the explicit value is a date (L) value, you must use the format that the DATFMT keyword specifies. If the DATFMT keyword is not specified or if DATFMT specifies *JOB, then you must use the *ISO format. Also, you must use the separator that the DATSEP keyword specifies. If the DATSEP keyword is not specified, or if DATSEP specifies the *JOB separator, a slash (/) must be used for the separator.
- If the explicit value is a time (T) value, you must use the format that the TIMFMT keyword specifies. If the TIMFMT keyword is not specified, you must use the *ISO format. Also, you must use the separator that the TIMSEP keyword specifies. If the TIMSEP keyword is not specified or if TIMSEP specifies the *JOB separator, you must use a colon (:) for the separator.
- If the explicit value is a timestamp (Z) value, you must use the *yy-mm-dd-hh.mm.ss.mmmmmm* format.

The *BLANK special value indicates field data that is composed of all blanks. The *CUR special value indicates that the date, time, or timestamp makes up the current field data, depending on the data type of the field.

During an output operation, the field data is compared to each program-value on the MAPVAL keyword in the order in which the program-values are specified. For the first match that is found, the corresponding system-value replaces the current field data. If a match does not exist, the field data does not change.

During an input operation, the field data is compared to each system-value on the MAPVAL keyword in the order in which the system-values are specified. For the first match that is found, the corresponding program-value replaces the current field data. If a match does not exist, the field data remains the same.

Option indicators are not valid for this keyword, although you can use option indicators to condition the field for which it is specified.

Example

The following example shows how to specify the MAPVAL keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A
00020A          R RECORD
00030A            DATFLD1         L         DATFMT(*MDY) DATSEP('/')
00040A                                      MAPVAL(('01/01/40' *BLANK))
```

On output, if the field data equals '01/01/40', the field data is changed to all blanks. On input, if the field data is blank, the field data is changed to '01/01/40'.

MDTOFF (Modified Data Tag Off) keyword for display files

You use this record-level keyword with the OVERLAY keyword to set off modified data tags (MDTs) for input-capable fields in record formats already on the display.

The format of the keyword is:

```
MDTOFF[(*UNPR | *ALL)]
```

The MDTs are set off when your program sends an output operation to the record format you are defining.

To set off MDTs for unprotected fields only (those without the DSPATR(PR) keyword in effect), specify the *UNPR parameter value (this is also the default if no parameter value is specified). To set off MDTs for all input-capable fields, specify the *ALL parameter value.

Your program can select the DSPATR(MDT) keyword for fields in the same record format for which it selects MDTOFF (any parameter). If so, these fields are displayed with their MDTs set on.

The ERASEINP(*ALL) keyword implies MDTOFF(*UNPR) unless MDTOFF(*ALL) is specified.

If the ERASEINP(*MDTON) keyword is specified with MDTOFF(*ALL), the end effect is as if the ERASEINP(*ALL) keyword and MDTOFF(*ALL) are both specified. This is also true if the ERASEINP keyword is specified with no parameter value.

Option indicators are valid for this keyword.

MDTOFF is not valid for the subfile record format (identified by the SFL keyword). It is valid for all other record formats for which OVERLAY keyword is also specified.

Example

The following example shows how to specify the MDTOFF keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A          R RECORD1                   OVERLAY MDTOFF
00020A            FLD1          6   B  2  2
00030A            FLD2          6   B  3  2
00040A*
00050A          R RECORD2                   OVERLAY MDTOFF(*UNPR)
00060A            FLD21         6   B  4  2
00070A            FLD22         6   B  5  2
00080A*
00090A          R RECORD3                   OVERLAY MDTOFF(*ALL)
00100A            FLD31         6   B  6  2
00110A            FLD32         6   B  7  2
00120A            FLD33         6   B  8  2DSPATR(PR)
     A
```

RECORD1 and RECORD2 have equivalent MDTOFF keyword specifications. When RECORD1 or RECORD2 is displayed, the MDT of each input-capable field already on the display is set off, unless the field has the DSPATR(PR) keyword in effect (FLD33, when displayed, is such a field). When RECORD3 is displayed, the MDTs of each input-capable field already on the display is set off even if the DSPATR(PR) keyword is in effect for the field.

MLTCHCFLD (Multiple-Choice Selection Field) keyword for display files

You use this field-level keyword to define a field as a multiple-choice selection field.

A multiple-choice selection field is a field that contains a fixed number of choices from which a user can select multiple choices. The field appears as a vertical or horizontal list of choices, an input field to the left of each choice, or as a group of check boxes.

If you see an input field, instead of a check box to the left of each choice, the selection character default value is a slash (/). Message CPX5A0C contains the country-designated selection characters. The value can be changed to allow alternate selection characters for a multiple-choice selection field. These characters are the allowed uppercase or lowercase country-designated selection characters. The characters are defined when the display file is created.

The format of the keyword is:

```
MLTCHCFLD[([*RSTCSR | *NORSTCSR]
[*NOSLTIND | *SLTIND]
[[(*NUMCOL nbr-of-cols) | (*NUMROW nbr-of-rows)]
[(*GUTTER  gutter-width)]])]
```

Parameters are optional. If none are specified, the multiple-choice selection field choices will be arranged in a single vertical column. The user will be allowed to move the selection cursor out of this field using the arrow keys. There will be three spaces between choices and selection indicators will be displayed.

The RSTCSR parameter specifies whether the arrow keys should be allowed to move the selection cursor outside of the selection field. *RSTCSR specifies that the arrow keys will not cause the selection cursor to move outside of the selection field. *NORSTCSR specifies that the arrow keys will cause the selection cursor to leave the selection field. The default is *NORSTCSR.

Note: An exception to the restrictions imposed by *RSTCSR happens if the selection field is the only field contained within a pull-down window. In that case, when the selection cursor is within the leftmost or rightmost columns, the left and right arrow keys will close the current pull-down window and open the pull-down window associated with the menu-bar choice to the left or right of the current menu-bar choice.

The *RSTCSR parameter is ignored on displays that are not connected to a controller that supports an enhanced interface for nonprogrammable workstations.

The SLTIND parameter indicates whether selection indicators (such as check boxes) should be displayed. *NOSLTIND specifies that the selection indicators should not be displayed. The default is *SLTIND.

*NUMCOL specifies that this selection field should be displayed in multiple columns with the choices spread across the columns in this manner:

```
choice1   choice2   choice3
choice4   choice5   choice6
choice7   choice8   choice9
```

The nbr-of-cols portion of the parameter specifies how many columns the selection field should contain. Nbr-of-cols must be a positive integer and the entire multiple-choice selection field must be able to fit on the display when placed in the specified number of columns.

*NUMROW specifies that this selection field should be displayed in multiple rows with the choices spread across the columns in this manner:

```
choice1   choice4   choice7
choice2   choice5   choice8
choice3   choice6   choice9
```

The nbr-of-rows portion of the parameter specifies how many rows the selection field should contain. Nbr-of-rows must be a positive integer and the entire multiple-choice selection field must be able to fit on the display when placed in the specified number of rows.

The *GUTTER parameter is optional and specifies the number of spaces to be placed between each column of the multiple-choice selection field. It can only be specified if either *NUMCOL or *NUMROW

has been specified and must immediately follow the (*NUMxxx #) parameter. The gutter-width must be a positive integer of at least 2. If *GUTTER is not specified, the default of gutter-width is set to three spaces.

A field containing the MLTCHCFLD keyword must contain one or more CHOICE and CHCCTL keywords defining the choices for the field.

The field containing the MLTCHCFLD keyword must be defined as an input-capable field with the data type Y and length of two. The position specified for the field is the position of the input field displayed to the left of the first choice or of the uppermost check box. If *NOSLTIND is used on the PULLDOWN keyword and the device is attached to a controller that supports an enhanced interface for nonprogrammable workstations, the position is the first character of the text for the first choice. On input, the field contains the number of the choices selected, or 0 if no choice was selected. On output, the value of the field is ignored.

Provide a minimum of two spaces between the end of the previous field and the position specified for the multiple choice field. This allows an ending attribute for the previous field, and a beginning attribute for the multiple choice field. These attributes cannot overlap.

The following keywords can be specified on a field with the MLTCHCFLD keyword:

ALIAS
AUTO(RA)
BLANKS
CHANGE
CHCAVAIL
CHCUNAVAIL
CHCSLT[1]
CHCCTL
CHECK(ER, FE)[2]
CHOICE
CHGINPDFT
COLOR[3]
FLDCSRPRG
DSPATR(RI UL BL CS HI ND PC)[3]
ERRMSG
ERRMSGID
INDTXT
OVRATA
OVRATR
PUTRETAIN
TEXT

Notes:

1. CHCSLT functions only if the multiple-choice selection field is displayed in a pull-down menu that does not display selection indicators, when PULLDOWN(*NOSLTIND) is specified.
2. Check(FE) applies only to a display attached to a controller that does not support an enhanced interface.
3. If the COLOR or DSPATR keyword is specified for a field with the MLTCHCFLD keyword, it applies only to the input field portion of the selection field on character-based displays.

Option indicators are not valid for this keyword.

Example

The following example shows how to specify the MLTCHCFLD keyword:

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A
     A          R RECORD
     A            F1             2Y 0B  3 35MLTCHCFLD
     A  01                                  CHOICE(1 '>Undo       ')
     A                                      CHOICE(2 &MARKTXT);
     A                                      CHOICE(3 '>Copy       ')
     A                                      CHCCTL(1 &CTLONE1 MSG1111 QUSER/A)
     A                                      CHCCTL(2 &CTLTWO1 &MSG1 &LIB/&MSGF);
     A                                      CHCCTL(3 &CTLTHR1);
     A            CTLONE1         1Y 0H
     A            CTLTWO1         1Y 0H
     A            CTLTHR1         1Y 0H
     A            MSGF           10A  P
     A            LIB            10A  P
     A            MARKTXT        10A  P
     A
```

The CHCCTL keyword is required for each CHOICE keyword used for the MLTCHCFLD.

On input, the hidden field for the CHCCTL keyword indicates whether the choice was selected. 0 indicates the choice was not selected; 1 indicates the choice was selected. On output, the hidden field controls the availability of the choice, and is used to set a default selection of a choice. 0 indicates the choice should be available, 1 indicates the choice should be selected by default, and 2 indicates the choice is unavailable. Other values, such as 0, are truncated.

MNUBAR (Menu Bar) keyword for display files

You use this record-level keyword to define a menu bar.

A menu bar is a horizontal list of choices that is followed by an optional separator line. The choices represent groups of related actions that the application user can select. For example, a group of actions appears in a pull-down menu when the user selects a menu-bar choice. A menu-bar record contains a field with one or more MNUBARCHC keywords that define the menu-bar choices. The separator line is generated by the system.

The format of the keyword is:

```
MNUBAR([*SEPARATOR | *NOSEPARATOR])
```

The parameter is optional and specifies whether a separator line should be placed below the last line of the menu-bar choices. *SEPARATOR indicates that a menu-bar separator line should be placed after the last line of the menu-bar choices. *NOSEPARATOR indicates that a menu-bar separator line should not be displayed. The default is *SEPARATOR.

Note: If *NOSEPARATOR is specified, the MNUBARSEP keyword cannot be specified on this record.

A record with the MNUBAR keyword specified must contain one and only one menu bar field (a field with one or more MNUBARCHC keywords), and cannot contain any displayable fields other than the menu bar field.

The following keywords are allowed on a record containing the MNUBAR keyword:

CAnn	HLPTITLE	MNUCNL
CFnn	HOME	OVERLAY
CLEAR	INDTXT	PAGEDOWN/PAGEUP
CLRL	INVITE	PRINT
CSRLOC	KEEP	PROTECT
DSPMOD	LOCK	ROLLUP/ROLLDOWN
HELP	MNUBARDSP	TEXT
HLPCLR	MNUBARSEP	UNLOCK
HLPCMDKEY	MNUBARSW	VLDCMDKEY
HLPRTN		

Note: These keywords are ignored on the menu-bar record if the menu-bar record is displayed by the system (for example, if MNUBARDSP is specified on a record other than this menu-bar record).

Option indicators are not valid for this keyword.

Example

The following example shows how to specify the MNUBAR keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A          R RECORD1                   MNUBAR
     A            MNUFLD         2Y 0I  1  2
     A                                       MNUBARCHC(1 RCDFILE 'File')
     A                                       MNUBARCHC(2 RCDEDIT 'Edit')
     A                                       MNUBARCHC(3 RCDVIEW 'View')
     A                                       MNUBARCHC(4 RCDOPT 'Options')
     A                                       MNUBARCHC(5 RCDHELP 'Help')
     A                                       :
```

```
     A                                      :
     A
```

In this example, RECORD1 is defined as a menu-bar record. When RECORD1 is displayed, the menu-bar choices defined on the field MNUFLD are displayed as a menu bar. The menu-bar choices are followed by a separator line. On character-based displays, the separator line is made up of blue dashes. On a graphical display attached to a controller that supports an enhanced interface for nonprogrammable workstations, the separator line is a solid line.

MNUBARCHC (Menu-Bar Choice) keyword for display files

You use this field-level keyword to define a choice for a menu-bar field.

A menu-bar choice represents a group of related actions that the application user can select. A group of actions appears in a pull-down menu when the user requests a menu-bar choice.

The format of the keyword is:

```
MNUBARCHC(choice-number pull-down-record choice-text
[&return-field])
```

The choice-number parameter is required and specifies an identification number. The choice number is returned to the application to indicate which choice in the menu bar was selected. Valid values for the choice number are integers 1 to 99. Duplicate values within a single menu-bar field are not allowed.

The pull-down-record parameter is required and specifies the name of the pull-down record that is displayed when the user selects this choice. The record specified must exist within the file and must contain a PULLDOWN keyword.

The choice-text parameter is required and defines the text that appears in the menu bar for the choice. The parameter can be specified in one of two forms:

- As a character string: 'Choice text '
- As a program-to-system field: &field-name

  The field-name specified must exist in the menu bar record and must be defined as a character field with usage P.

The choice text must fit on one line of the display for the smallest display size specified for the file. Because the text for the first menu-bar choice on a line begins at position 3 and a trailing blank is always inserted after the choice text, the maximum length of the choice text is 76 if the smallest display size for the file is 24 x 80 and 128 if the smallest display size for the file is 27 x 132.

When the choice text contained in the character string or the program-to-system field is displayed, trailing blanks in the text are truncated and 3 blank spaces are inserted between choices. However, the number of lines that the menu-bar field occupies on the display is determined by the sum of the lengths of the choice-text parameters, plus 3 blank spaces between each choice. The length of the choice-text is either the length of the character string, excluding trailing blanks, or the length of the program-to-system field. The maximum number of lines that a menu bar field can occupy is 12 lines (this includes the separator line).

Within the choice text, you can specify a mnemonic for the choice by using a greater than character (>) to indicate the mnemonic character. The character to the right of the > is the mnemonic. Examples:

Choice text
Appears as

'>File'
File

'F>inish'
Finish

'Save >As...'
Save As...

'X >= 1'
X = 1

To specify a > as a character in the text, you must specify it twice, just as you must specify the apostrophe character twice in order to get a single apostrophe character in the text.

Choice text
Appears as

'X >>= 1'
X >= 1

'X >>>= 1'
X >= 1

Note: It is not possible to specify the > as the mnemonic character.

The mnemonic character indicated must be a single-byte character and must not be a blank. Only one mnemonic is allowed in the choice text, and the same mnemonic character cannot be specified for more than one choice.

The return-field parameter is optional and specifies whether control is returned to the application because a menu bar choice was selected. This parameter specifies the name of a hidden field in the menu-bar record that contains the number of the choice selected when control is returned to the application. The hidden field is defined as a data type Y (numeric), the length of the field is two, and decimal positions are 0. The presence of a choice number in this field indicates that control has been returned to the application because a menu-bar choice was selected. The next operation of the application updates (if necessary) and writes the pull-down record associated with that choice; that is, the pull-down record specified on the MNUBARCHC keyword for the choice. When a choice number is returned in this field, zero is returned in the field that contains the choice number after pull-down input has been received. Likewise, when pull-down input has been received, zero is returned in this field, and the presence of a choice number in menu-bar field or the choice field in the application record indicates that the application should process the pull-down input.

The menu bar field contained in the MNUBARCHC keywords is defined as an input-capable field with data type Y (numeric). The length of the field is two and decimal positions 0. If the menu bar record is read, the number of the choice selected (if any) is returned in the menu-bar field. The menu-bar field must always be defined as starting in row 1, column 2.

When MNUBARCHC is specified on a field, the MNUBAR keyword is required at the record level.

Multiple MNUBARCHC keywords can be specified for one menu bar field. The number of MNUBARCHC keywords that can be specified is limited only by the lengths of the choice text parameters (excluding trailing blanks in character string choice-text) and the 12 line limit for a MNUBAR. All the choices defined for a menu bar field must fit on the screen, allowing for 3 spaces between each choice.

The following keywords can be specified on a field with the MNUBARCHC keyword:

ALIAS	INDTXT
CHCAVAIL	MNUBARSEP
CHCSLT	TEXT

Option indicators are valid for this keyword.

Examples

The following examples show how to specify the MNUBARCHC keyword.

Example 1

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A                                  DSPSIZ(*DS3 *DS4)
     A          R MENUBAR               MNUBAR
     A            MNUFLD         2Y 0B  1  2
```

```
     A                                      MNUBARCHC(1 PULLFILE +
     A                                         '>File          ')
     A  01                                  MNUBARCHC(2 PULLEDIT +
     A                                         '>Edit          ')
     A                                      MNUBARCHC(3 PULLVIEW +
     A                                         '>View          ')
     A                                      MNUBARCHC(4 PULLOPT +
     A                                         '>Options       ' &RTNFLD);
     A                                      MNUBARCHC(5 PULLHELP +
     A                                         '>Help          ')
     A            RTNFLD         2Y 0H
```

In this example, five menu bar choices (File, Edit, View, Options and Help) are defined in a menu bar. If option indicator 01 is on and the menu bar record is written before the system displays it, the Edit choice is displayed when the system displays the menu bar. If option indicator 01 is off or the menu-bar record is not written before the system displays it, the Edit choice is not displayed. If the Edit choice is not displayed, the list of choices are compressed and the View choice will follow the File choice, with 3 blank spaces in between.

If the File choice is selected, record PULLFILE is displayed as a pull-down menu beneath the File choice. If the Options choice is selected, control is returned to the application. The application can update the PULLOPT record before the system displays it as a pull-down menu.

On displays capable of a single-character underline, the mnemonic for each choice is the first character in the text. If the menu-bar record is read, the menu-bar field MNUFLD contains the number of the choice selected, or 0 if no choice was selected.

The text for each choice has been specified as a character string, with 15 spaces available for the text. However, the trailing blanks are removed before the system calculates how many choices fit on a line. Therefore, the maximum space required for the menu bar is 87 positions (28 for the text within the character strings, plus 3 spaces between each choice). The menu-bar choices occupy one line. The menu-bar separator occupies one more line; therefore the entire menu bar occupies two lines.

Example 2

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A                                      DSPSIZ(*DS3 *DS4)
     A          R MENUBAR                   MNUBAR
     A            MNUFLD         2Y 0B  1  2
     A                                      MNUBARCHC(1 PULLFILE +
     A                                         &FILETXT);
     A  01                                  MNUBARCHC(2 PULLEDIT +
     A                                         &EDITTXT);
     A                                      MNUBARCHC(3 PULLVIEW +
     A                                         &VIEWTXT);
     A                                      MNUBARCHC(4 PULLOPT +
     A                                         &OPTTXT &RTNFLD);
     A                                      MNUBARCHC(5 PULLHELP +
     A                                         &HELPTXT);
     A           FILETXT       15A  P
     A           EDITTXT       15A  P
     A           VIEWTXT       15A  P
     A           OPTTXT        15A  P
     A           HELPTXT       15A  P
     A           RTNFLD         2Y 0H
     A
```

This example is the same as example 1, except that the choice text has been specified using program-to-system fields.

At run time, the choice text to be displayed for each choice is retrieved from the program-to-system fields. Any mnemonics must be specified in the text supplied by the application at run time. As in example 1 when the menu-bar record is read, the menu-bar field MNUFLD contains the number of the choice selected, or 0 if no choice was selected.

As in example 1, the number of spaces available for the text for each choice is 15. The maximum space required for the menu bar is 78 positions (15 possible text positions for each of the 5 choices plus 3 spaces between choices). Because the smallest display size is 24x80 (*DS3), the menu-bar choices occupy 2 lines. The menu-bar separator occupies one more line, so the entire menu bar occupies 3

lines. However, the actual number of lines that is used to display the choices depends on the text that is contained in the program-to-system fields. When the menu bar is displayed, the trailing blanks in the P-fields are truncated, and 3 blanks are inserted between each choice.

Example 3

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A          R MENUBAR                   MNUBAR
     A            MNUFLD         2Y 0B  1  2
     A                                       MNUBARCHC(1 PULLFILE +
     A                                          '>File        ')
     A                                       MNUBARCHC(2 PULLEDIT +
     A                                          '>Edit        ' &RTNFLD);
     A            RTNFLD         2Y 0H
```

In this example, if choice 2 in the menu bar is selected, control is returned to the application and field RTNFLD contains the number 2. Field MNUFLD contains 0, indicating no pull-down input received. The application must read record MENUBAR in order to get the contents of field RTNFLD. The application must then write record PULLEDIT. The system resumes control of the menu-bar interaction. If input is then entered in record PULLEDIT, control is returned to the application, and field MNUFLD contains number 2. Field RTNFLD contains 0, indicating control has been returned because pull-down input was received.

If choice 1 is selected, the system displays pull-down record PULLFILE. If input is entered in PULLFILE, control is returned to the application, and field MNUFLD contains number 1. Field RTNFLD contains 0, indicating control has been returned because pull-down input was received.

MNUBARDSP (Menu-Bar Display) keyword for display files

You use this record-level keyword to display a menu bar.

The MNUBARDSP keyword has two formats: one for records that contain the MNUBAR keyword and one for records that do not contain a MNUBAR keyword.

The format for MNUBARDSP when specified on a record that is not a menu bar record is:

```
MNUBARDSP(menu-bar-record &choice-field [&pull-down-input])
```

The format for MNUBARDSP when specified on a menu bar record is:

```
MNUBARDSP[(&pull-down-input)]
```

The menu-bar-record parameter specifies the menu-bar record that is to be displayed when this record is written. The menu-bar record must exist in the same file as the record you are defining.

The &choice-field parameter specifies the name of a hidden field, which on input contains the number of the choice (if any) selected from the menu bar. The field must exist in the record you are defining and it must be defined as numeric Y in position 35, usage H, length 2, and decimal positions 0.

The &pull-down-input parameter is optional and specifies the name of a hidden field that contains the input from the pull-down menu when the pull-down menu contains only a single-choice selection field. The field must exist in the record you are defining and it must be defined as a length 2, decimal positions 0, and zoned (S in position 35) field with usage H (hidden). On input, this field contains one of the following values:

Value
Meaning

0
No selection made.

n
Choice n in the pull-down menu was selected.

-1
Pull-down record contains something other than the one single-choice selection field. You must read the pull-down record to receive its contents.

Option indicators are valid for the MNUBARDSP keyword, and more than one MNUBARDSP keyword can be specified on the record if all are optioned. If more than one MNUBARDSP keyword is in effect when the record is written, the first one in effect is used.

Example 1

The following example shows how to specify the MNUBARDSP keyword on a record that is not a menu bar.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A
     A          R RECORD1
     A  01                                 MNUBARDSP(MENURCD &MNUCHOICE &INPUT);
     A            FIELD1        10A  B 10 12
     A            FIELD2         5S 0B 14 12
     A            MNUCHOICE      2Y 0H
     A            INPUT          2S 0H
     A          R MENURCD                  MNUBAR
     A            F1             2Y 0B  1  2
     A                                     MNUBARCHC(1 PULLFILE 'File')
     A                                     :
     A                                     :
     A
```

In this example, if option indicator 01 is on when record RECORD1 is written to the display, the system displays the menu bar in record MENURCD. When the menu bar is activated and the pull-down menu is selected, the number of the menu-bar choice is returned in field MNUCHOICE. If the pull-down menu selected contains one single-choice selection field, the choice made for that field is returned in field INPUT. Otherwise, field INPUT contains -1, indicating that the application must read the pull-down record to receive the pull-down input.

Example 2

The following example shows how to specify the MNUBARDSP keyword on a menu-bar record.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A          R MENURCD                  MNUBAR
     A  01                                 MNUBARDSP
     A            F1             2Y 0B  1  2
     A                                     MNUBARCHC(1 PULLFILE 'File')
     A                                     :
     A                                     :
     A
```

If option indicator 01 is on when record MENURCD is written to the display, the system displays the menu bar defined in MENURCD. If the user selects a pull-down menu from the menu bar, the number of the menu-bar choice selected is returned in F1 field.

MNUBARSEP (Menu-Bar Separator) keyword for display files

You use this field-level keyword on a menu-bar field to specify the color, display attributes, or the character that is used to form the menu-bar separator line.

The format of the keyword is:

```
MNUBARSEP([color] [display-attribute] [character])
```

One parameter must be specified.

The color parameter specifies the color of the separator characters on a color workstation. The parameter is specified as an expression of the form (*COLOR value).

The valid values for the color parameter are:

Value
Meaning

BLU
Blue

GRN
Green

PNK
Pink

RED
Red

TRQ
Turquoise

YLW
Yellow

WHT
White

If the color parameter is not specified, the default is blue. This parameter is ignored if it is specified for a menu bar on a monochrome display.

The display-attribute parameter specifies the display attributes of the separator characters. The parameter is specified as an expression of the form (**DSPATR value1 <value2 <value3...>>*).

The valid values for the display attributes are:

Value
Meaning

BL
Blink

CS
Column separator

HI
High intensity

ND
Nondisplay

RI
Reverse image

UL
Underline

The default display attribute for the menu-bar separator is normal (or low) intensity.

Note: Display attributes CS, HI, and BL can cause fields on 5292, 3179, 3197 Model C1 and C2, 3487 Models HC, and 3488[7] workstations to appear as color fields. Display attributes HI, RI, and UL cause a separator line not to be displayed.

The character parameter specifies the character that makes up the separator line. The parameter is specified as an expression of the form (*CHAR 'separator-character'). The separator-character value is one character. If this parameter is not specified, the default separator character is a dash (-) or on a graphical device this shows as a solid line. Although any displayable character can be specified as the separator character, it is recommended that you use invariant characters.

The following figure shows the invariant characters:

Table 10. Character set for system data

Hexadecimal	Character	Description
40		Blank
4B	.	Period

[7] Dependent on the monitor attached to the display device.

Table 10. Character set for system data (continued)

Hexadecimal	Character	Description
4C	<	Less-than sign
4D	(	Left parenthesis
4E	+	Plus sign
50	&	Ampersand
5C	*	Asterisk
5D	)	Right parenthesis
5E	;	Semicolon
60	-	Minus sign
61	/	Slash
6B	,	Comma
6C	%	Percent sign
6D	_	Underline
6E	>	Greater-than sign
6F	?	Question mark
7A	:	Colon
7D	'	Apostrophe
7E	=	Equal sign

Note: In addition, you can use any of the following characters:

- Uppercase alphabetic characters: A through Z
- Numeric characters: 0 through 9

When the MNUBARSEP keyword is specified on a field, the MNUBAR keyword must also be specified on the associated record. The *NOSEPARATOR parameter cannot be used on the MNUBAR keyword if the MNUBARSEP keyword is specified.

Option indicators are valid for this keyword.

If more than one COLOR keyword is specified, the color parameter, display attribute, and separator character are taken from the first keyword that was specified.

Example

The following example shows how to specify the MNUBARSEP keyword:

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A
     A          R MENUBAR                    MNUBAR
     A            MNUFLD         2Y 0B  1  2
     A                                       MNUBARSEP((*COLOR PNK) +
     A                                       (*DSPATR RI) (*CHAR  ' '))
     A                                       MNUBARCHC(1 PULLFILE +
     A                                          'File         ')
     A                                       MNUBARCHC(2 PULLEDIT +
     A                                          'Edit         ')
     A
```

In this example, the menu-bar separator is made up of pink blanks displayed in reverse image.

Related reference

COLOR (Color) keyword for display files
You use this keyword to specify the color of a field on a color display.

MNUBARSW (Menu-Bar Switch Key) keyword for display files

You use this file- or record-level keyword to assign a command attention (CA) key to be the Switch-to-menu-bar key.

If the menu-bar switch key is active and a menu bar is displayed, pressing the CA key will do one of the following tasks:

- If the cursor is located on the application record, pressing the switch key moves the cursor to the first field in the menu bar.
- If the cursor has been moved to the menu bar using the switch key, pressing the switch key again moves the cursor back to the location where the cursor was when the switch key was pressed to move the cursor into the menu bar.
- If the cursor has been moved to the menu bar manually (for example, using the cursor movement keys), pressing the switch key moves the cursor to the first input-capable field in the application record.

The format of the keyword is `MNUBARSW [(CAnn)]`.

The CAnn parameter is optional. If not specified, the default is CA10. Valid values for the CAnn parameter are CA01 through CA24.

Within a record, the CAnn key specified by the MNUBARSW keyword cannot be specified again using another keyword (such as MNUCNL). Because MNUBARSW at the file level extends to all records in the file, this must be considered when assigning a CAnn key.

If the MNUBARSW keyword is specified on the record, the CAnn key or default CA10 key can be used only as a CA key on other records, not as a CF key.

The MNUBARSW keyword is allowed only in a file containing a menu-bar record.

Option indicators are valid for this keyword.

Example

The following example shows how to specify the MNUBARSW keyword:

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A                                     MNUBARSW(CA10)
     A          R MENUBAR                  MNUBAR
     A            MNUFLD         2Y 0B  1  2
     A                                     MNUBARCHC(1 PULLFILE +
     A                                         '>File        ')
     A  01                                 MNUBARCHC(2 PULLEDIT +
     A                                         '>Edit        ')
     A
     A          R PULLEDIT                 PULLDOWN
     A            F1            1D 0B  1   2RANGE(1 3)
     A                                   1  5'1. Undo        '
     A                                   2  4'2. Mark        '
     A                                   3  4'3. Copy        '
     A            :
     A            :
     A          R APPSCR                   MNUBARDSP(MENUBAR &MNUCHOICE);
     A            FIELD1        10A  B 10 12
     A            FIELD2         5S 0B 14 12
     A                                  24  1'F12=Cancel '
     A            MNUCHOICE      2S 0H
     A
```

In this example, CA10 is defined as the Switch-to-menu-bar key for all records in the file. When the cursor is located anywhere except in the menu bar and CA10 is pressed, the cursor is moved to the File choice on the menu bar. If CA10 is pressed again while the cursor is located anywhere in the menu bar, the cursor is moved back to its previous location within the APPSCR record.

MNUCNL (Menu-Cancel Key) keyword for display files

You use this file- or record-level keyword to assign a command attention (CA) key to be the cancel key for menu bars or pull-down menus.

When the MNUCNL keyword is active and a pull-down menu is displayed, pressing the CA key cancels the pull-down menu and returns the cursor to the choice in the menu bar. If no pull-down menu is displayed and the cursor is located in the menu bar, pressing the CA key cancels the menu bar and returns the cursor to the application screen. If no pull-down menu is displayed and the cursor is on the application screen, pressing the CA key returns control to the application.

The format of the keyword is:

```
MNUCNL[(CAnn [response-indicator])]
```

The CAnn parameter is optional. If not specified, the default is CA12. Valid values are CA01 through CA24.

The response-indicator parameter is optional. The parameter is set on when the MNUCNL keyword is active on a record other than a menu bar or pull-down menu, and control is being returned to the application.

Within a record, the CAnn key specified by the MNUCNL keyword cannot be specified again using another keyword (such as MNUBARSW). Because MNUCNL at the file-level extends to all records in the file, this must be considered when assigning a CAnn key.

If the MNUCNL keyword is specified on the record, the CAnn key or default CA12 key can be used only as a CA key on other records, not as a CF key.

The MNUCNL keyword is allowed only in a file containing a menu-bar record.

Option indicators are valid for this keyword.

Example

The following example shows how to specify the MNUCNL keyword:

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A                                      MNUCNL(CA12 12)
     A          R MENUBAR                   MNUBAR
     A            MNUFLD         2Y 0B  1  2
     A                                      MNUBARCHC(1 PULLFILE +
     A                                         '>File        ')
     A  01                                  MNUBARCHC(2 PULLEDIT +
     A                                         '>Edit        ')
     A
     A          R PULLEDIT                  PULLDOWN
     A            F1             1D 0B  1  2RANGE(1 3)
     A                                   1  5'1. Undo        '
     A                                   2  4'2. Mark        '
     A                                   3  4'3. Copy        '
     A            :
     A            :
     A          R APPSCR                    MNUBARDSP(MENUBAR &MNUCHOICE);
     A            FIELD1        10A  B 10 12
     A            FIELD2         5S 0B 14 12
     A                                  24  1'F12=Cancel '
     A            MNUCHOICE      2S 0H
```

In this example, CA12 is defined as the cancel key for all records in the file. If CA12 is pressed while the pull-down menu, PULLEDIT, is displayed, the pull-down menu is canceled. If CA12 is pressed while the cursor is located on the menu bar (no pull-down menus are displayed), the menu bar is canceled and the cursor is moved back to the application record. If CA12 is pressed while the cursor is not located on the menu bar and no pull-down menu is displayed, response indicator 12 is set on and control is returned to the application program.

MOUBTN (Mouse Buttons) keyword for display files

You use this file-level or record-level keyword to associate a Command key or EVENT-ID parameter with one or two pointer-device events.

When a specified pointer device single event is performed and no other function has a higher priority, the keyboard is locked, the cursor is moved to the pointer device cursor location, and the specified Command key or EVENT-ID is returned to the application. If the Command key or EVENT-ID normally results in entry field data validation, the data validation is performed first. If the specified Command key or EVENT-ID normally returns entry field data, inbound entry field data is included.

For pointer device double events, inbound data is not returned until the trailing edge event also occurs. When the leading edge event is detected, a programmable-two-event state is entered, a marker box is drawn around the location of the pointer device cursor (4 blue lines around the character), the pointer device cursor color is changed to white on a color nonprogrammable workstations (NWS) and the trailing edge event is looked for. Keystrokes and host data streams will cancel the programmable-two-event state. Some pointer device events are ignored while waiting for the trailing edge event. When the trailing edge event is received, the marker box is erased, the pointer device cursor color is changed to input inhibited, then keyboard is locked, the text cursor is moved to the location of the pointer device cursor, and the inbound data is returned to the host.

Note: Although it is permitted, it is not be advisable to program some combinations of pointer device events on the same mouse button and shift state. For example, if the right button is programmed, right button double click should not be programmed because it cannot be detected due to the keyboard being locked from the right button pressed event. Using the *QUEUE parameter will allow the application to handle this situation.

The format of this keyword is:

```
MOUBTN(EVENT  [TRAILING-EVENT]  {Command key | EVENT-ID}
[*QUEUE | *NOQUEUE])
```

The EVENT parameter is required and indicates the pointer device event that will be associated with the Command key or EVENT-ID parameter. Valid values for the EVENT parameter are:

Value
Meaning

***ULP**
Unshifted Left button Pressed

***ULR**
Unshifted Left button Released

***ULD**
Unshifted Left button Double click

***UMP**
Unshifted Middle button Pressed

***UMR**
Unshifted Middle button Released

***UMD**
Unshifted Middle button Double click

***URP**
Unshifted Right button Pressed

***URR**
Unshifted Right button Released

***URD**
Unshifted Right button Double click

***SLP**
Shifted Left button Pressed

***SLR**
Shifted Left button Released

***SLD**
Shifted Left button Double click

***SMP**
Shifted Middle button Pressed

***SMR**
Shifted Middle button Released

***SMD**
Shifted Middle button Double click

***SRP**
Shifted Right button Pressed

***SRR**
Shifted Right button Released

***SRD**
Shifted Right button Double click

The TRAILING-EVENT parameter is optional. If specified, this parameter defines the trailing event of a two event pointer device definition. This parameter has the same valid values as the EVENT parameter. A TRAILING-EVENT is be the trailing edge event for multiple leading edge events and have different Command key or EVENT-ID associations for each one. An event is be a trailing edge event and also defined as a single event (with a different Command key or EVENT-ID association).

Note: There are some restrictions to the Event definitions.

- An event cannot be both a single event and a leading edge of a two event sequence.
- A leading edge event can have only one trailing edge event associated with it.

If you use the same event as a single or leading edge event with multiple mouse button definitions, only the first definition is used.

Either the Command key or EVENT-ID parameter must be specified and associates a Command key or EVENT-ID value with the pointer device event indicated by the first (and second, if provided) parameters. Valid values for a Command key are CA01 through CA24, CF01 through CF24, ENTER, ROLLUP, ROLLDOWN, HELP, HOME, PRINT and CLEAR. Valid EVENT-IDs are E00 through E15. EVENT-IDs are similar to CAxx keys in that no input data is transmitted from the device.

The QUEUE parameter is optional and specifies if the single event being defined should be queued by the controller if received while the keyboard is locked. This feature is primarily used to allow a double-click to be defined for a mouse button that also has either the pressing or releasing of the same button defined. If the queueing is not enabled for the double click, the application will probably not know that the double-click has occurred since the keyboard will still be locked from processing the pressing/releasing of the button. The default is *NOQUEUE.

The following keywords cannot be specified when the listed Command key has been used on the MOUBTN keyword:

Command key	Mutually exclusive keyword
CFxx	ALTHELP(CAyy), CAxx where xx=yy.
CAxx	ALTPAGEDWN(CFyy), ALTPAGEUP(CFyy), CFxx where xx=yy.
CF01	ALTHELP with no parameter
CA07	ALTPAGEUP with no parameter

Command key	Mutually exclusive keyword
CA08	ALTPAGEDWN with no parameter

Although not required, it is valid to specify the CA01-CA24, CF01-CF24, ROLLUP, ROLLDOWN, PAGEUP, PAGEDOWN, CLEAR and HLPRTN keywords even though the associated function key is defined as a command key to a single or double mouse event. Associating a Command key to a mouse event will automatically enable the corresponding Command key from the keyboard. If you want to associate a response indicator with the function key, you must use one of the listed keywords to do this. In that case, the response indicator will be set on regardless if the Command key originates from the keyboard or from a mouse event.

Option Indicators are valid for this keyword.

Example

The following example shows how to specify the MOUBTN keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A                                     MOUBTN(*URP CF03)
     A                                     MOUBTN(*SRP CF12)
     A                                     CF12(12 'CANCEL')
     A          R RECORD1
     A                                     MOUBTN(*ULP *UMP ROLLUP)
     A                                     MOUBTN(*UMP *ULP ROLLDOWN)
     A                                   1 10'ONE--:'
     A            FIELD1        10A  I  1 17TEXT('ONE')
     A                                   2 10'TWO--:'
     A            FIELD2        10A  I  2 17TEXT('TWO')
     A
```

In this example, 2 Programmable Mouse Button events have been defined that will be in effect for all records within this file (unless overridden at the record level). These definitions associate the unshifted right mouse button pressed event with the CF03 key and the shifted right mouse button pressed event with the CF12 key. The CF03 key has no response indicator associated with it while the CF12 key has response indicator 12 associated with it.

Within RECORD1, two two-event mouse button events have been defined. The first associates the unshifted left mouse button pressed followed by the unshifted middle mouse button pressed with the ROLLUP key. The second associates the unshifted middle mouse button pressed followed by the unshifted left mouse button pressed with the ROLLDOWN key. These definitions are only valid when RECORD1 is the last record to be written to the display.

MSGALARM (Message Alarm) keyword for display files

You use this file-level or record-level keyword to specify that the system is to sound the audible alarm when this record is displayed with an active ERRMSG, ERRMSGID, SFLMSG, or SFLMSGID keyword, or when a validity checking error is detected. The alarm is of short duration.

This keyword has no parameters.

If you specify the MSGALARM and ALARM keywords on the same record format and both are active, the alarm sounds only once.

Option indicators are valid with this keyword.

Example 1

The following example shows how to specify the MSGALARM keyword at record level.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A          R RCD1
00020A                                     MSGALARM
00030A            FLD01          8A    12 10
00040A  12                                 ERRMSGID(XYZ0123 MSGFILE)
     A
```

When record format RCD1 is on the display and RCD1 is put to the display again and indicator 12 is on, the message XYZ0123 from message file MSGFILE is displayed on the message line and the workstation alarm sounds.

Example 2

The following example shows how to specify the MSGALARM keyword at file level.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A  01                                 MSGALARM
00020A          R RCD1
00030A            FLD01          8A    12 10
00040A  12                                 ERRMSGID(XYZ0123 MSGFILE)
00050A
00060A          R RCD2
00070A            FLD02          8A    12 10
00080A  10                                 ERRMSG('Message text')
     A
```

When record format RCD1 is on the display and RCD1 is put to the display again and indicators 01 and 12 are on, the message XYZ0123 from message file MSGFILE is displayed on the message line, and the workstation alarm sounds.

When record format RCD2 is on the display and RCD2 is put to the display again and indicator 10 is on (but 01 is off), the message text is displayed on the message line and the alarm does NOT sound.

MSGCON (Message Constant) keyword for display files

You use this field-level keyword to indicate that the text for constant fields is contained in a message description.

If the message description does not exist at DDS compilation time, the file will not be created. If you change the message description, you need to create the file again if you want the display file to contain the updated message.

The format of the keyword is:

```
MSGCON(length message-ID [library-name/]message-file-name)
```

The length parameter specifies the maximum length of the message description. The length can be from 1 to 132 bytes. If the message description is less than the length specified, the remaining bytes are padded with blanks (hex 40). If the message description is longer than the length specified, the message description is truncated to the specified length and a warning message appears.

The message-ID parameter specifies the message description that contains the text to use as the value of the constant field.

The message-file-name parameter identifies the message file that contains the message description. The library-name parameter is optional.

The MSGCON keyword must be explicitly specified for the field. The MSGCON keyword cannot be used to initialize a named field.

The DFT and MSGCON keywords are functionally equivalent. If you specify the DFT and MSGCON keywords for the same field, the MSGCON keyword is ignored and the file is not created.

The MSGCON keyword cannot be specified with any of the following keywords:

DATE
DFT
EDTCDE
EDTWRD
TIME

Option indicators are not valid for changing the value of the message line, but they are valid for conditioning the presence or absence of the message on the display.

Example

The following example shows how to specify the MSGCON keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A          R RECORD1
00020A                                  2  1MSGCON(10 MSG0001 MESSAGE/MSGF)
     A
```

MSG0001 in message file MSGF in library MESSAGES contains the message text.

MSGID (Message Identifier) keyword for display files

You use this field-level keyword to allow an application program to identify, at program run time, the message description that contains text for a named field.

The parameter values on this keyword can specify fields that contain the message identifier, message file, and library. After the program sets the fields to the values you want, an output operation causes the message to be retrieved from the message file and displayed. The retrieved message is truncated if it is longer than the MSGID field. The retrieved message is padded with blanks if it is shorter than the MSGID field.

See "System/36 environment considerations for display files" on page 247 for information about how to specify the MSGID keyword in files that are used in the System/36 environment.

The format of the keyword is:

```
MSGID(message-identifier [library-name/]message-file)
or
MSGID(*NONE)
```

The message-identifier parameter describes the message description containing the text for the named field. You can specify this parameter in one of two ways:

- `[msg-prefix] &field-name`

  A message-prefix and a field-name. The field-name must exist in the same record format as the MSGID field. If you specify a prefix, the length of the prefix must be three, and you must define the field-name as a character field of length four (4), and usage H, P, B, or O. If you do not specify a prefix, you must define the field-name as a character field length of seven (7), and usage H, P, B, or O.
- `[msgid-prefix] msg-id`

  You can also use a value or a combination of values to specify the message identifier. If you specify a prefix, the prefix length must be three (3), and the msg-id length must be four (4). If you do not specify a prefix, the msg-id length must be seven (7).

The message-identifier is a required parameter.

The message-file and library parameters identify the message file containing the message description. You can specify the message-file and library parameters in one of the following forms:

- `[library-name/]file-name`
- `[&field1/]&field2`

  where the lengths of field1 and field2 are ten (10).

  The field names must exist in the same record format as the MSGID field, and the fields must be defined as having character lengths of ten (10) and usage H, P, B, or O.
- Combination of field names and constants:
  - `library-name/&field1` (length of field1 is 10)
  - `&field2/file-name` (length of field2 is 10)

The message-file is a required parameter. If you do not specify the library parameter, *LIBL is used to search for the message file at program runtime. The library is an optional parameter.

The *NONE parameter indicates that no message text is to be displayed.

The user or program can override the message file name by using the OVRMSGF command.

You can specify multiple MSGID keywords on a field. When more than one MSGID keyword is specified, option indicators are required on all except the last MSGID keyword on a field. Option indicators are not allowed on the last (or only) MSGID keyword specified on a field. If more than one MSGID keyword is in effect for a field, the first MSGID specified is used.

You can specify multiple MSGID keywords within a record format. You can specify field names used as parameters on more than one MSGID keyword.

You must have use authority to the message file at program runtime.

The MSGID field must be output-capable (usage B or O). The length specified for the field is the length of the message text that is to be displayed. It should be the length of the longest message to be displayed.

The MSGID field itself does not appear in the output buffer, but it does appear in the input buffer if it is defined as input/output (usage B). MSGID fields that are defined as output-only cannot be used in high-level language programs.

The MSGID parameter fields (if any) appear in the output buffer, but they appear in the input buffer only if they are defined as hidden or input/output (usage H or B) fields.

A program-to-system field can be specified as the message-id, file, or library name.

You cannot specify MSGID in a subfile record format (SFL keyword).

The following keywords cannot be specified on a field with the MSGID keyword:

DFT
DFTVAL
FLTFIXDEC
FLTPCN
MSGCON

Example

The following example shows how to specify the MSGID keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A          R RECORD1
     A            MSGFIELD1     40A  B 02 10MSGID(CPD0001 QGPL/USRMSG)
     A  01        MSGFIELD2     10A  O 02 60
     A  25                                  MSGID(&MSGIDNUM &MSGFILENM)
     A                                      MSGID(CPD1234 QGPL/USRMSG)
     A            MSGFIELD3     80A  B 03 01
     A                                      MSGID(USR &MSGNBR +
     A                                      QGPL/&MSGFILENM)
     A            MSGIDNUM       7A  P
     A            MSGFILENM     10A  H
     A            MSGNBR         4A  B 07 01
     A
```

When RECORD1 is displayed:

- MSGFIELD1 contains the first 40 characters of message CPD0001 from message file USRMSG in library QGPL. Because the field is input/output (usage B), the value of the field can be changed by the user.
- If option indicator 01 is off, MSGFIELD2 is not displayed. When option indicators 01 and 25 are on, MSGFIELD2 contains the first 10 characters of the message identified by the fields MSGIDNUM and MSGFILENM. Values for MSGIDNUM (the message identifier) and MSGFILENM (the message file) must be set in the program before the display of RECORD1.

  When option indicator 01 is on but option indicator 25 is off, MSGFIELD2 contains the first 10 characters of message CPD1234 from the message file USRMSG in library QGPL. Because MSGFIELD2 is an output-only field (usage O), it cannot be used in the program.

- MSGFIELD3 contains the first 80 characters of the message identified by the prefix USR, the message number set in field MSGNBR, and the message file set in field MSGFILENM.

Related reference

MSGID keyword

You use this field-level keyword to allow an application program to identify, at program run time, the message description that contains text for a named field.

MSGLOC (Message Location) keyword for display files

You use this file-level keyword to move the message line to the specified line number.

The format of the keyword is:

```
MSGLOC(line-number)
```

The parameter value is required and must be in the range 1 through 28. Any of these numbers are always valid, regardless of the display sizes specified on the DSPSIZ keyword. A diagnostic will be issued when the file is opened if a message location is in the 26 to 28 range for a 24 x 80 display size.

If MSGLOC is not specified, the message line is the last line of the display. The message line is the display location for the following messages:

- Validity check errors
- Invalid function key messages
- Messages defined as parameter values for the ERRMSG and SFLMSG keywords
- Messages identified by the ERRMSGID and SFLMSGID keywords (the entire display is used for message help)
- Message fields
- Operator error codes and their associated messages

Display size condition names must be specified if the message line for the secondary display size is different from the default message line.

If you do not specify the MSGLOC keyword, the following default values are to be established:

24 x 80 display size:
line 25

27 x 132 display size:
line 28

The default of line 25 for the 24 x 80 display size gives the following results:

- If the display is sent to a 5250 device or a 5251 model 12, line 24 is used as the message line.
- If the display is sent to a 3180-2 device or a 3197 model D1, D2, W1, or W2 attached to a local 6040 or 6041 controller, or remotely attached to a 5294 or 5394 controller, line 25 is used as the message line.

If the ERRSFL keyword is specified in the file, you cannot specify a message location value of 25 for the 24 x 80 display size or 28 for the 27 x 132 display size. When the ERRSFL keyword is in the file, but MSGLOC is not specified, the following default values are to be established:

24 x 80 display size:
line 24

27 x 132 display size:
line 27

The MSGLOC keyword specification is in effect continuously from file opening to file closing. It can be temporarily overridden if the file you are defining is suspended while another file is opened to the same workstation device. The message location in effect for the other file is used until the file that you are defining is restored.

Any data on the message line before the message appears is saved and restored after the Reset key is pressed.

Option indicators are not valid for this keyword.

Examples

The following examples show how to specify the MSGLOC keyword.

Example 1

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A                                     MSGLOC(1)
     A
```

In this example, the message line is moved to line 1 for the primary display size. (Without the DSPSIZ keyword, the primary display size is the 24 x 80 display.)

Example 2

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A                                     DSPSIZ(*DS3 *DS4)
     A                                     MSGLOC(1)
     A  *DS4                               MSGLOC(1)
     A
```

The message line is moved to line 1 for both the primary display size 1 and the secondary display size 2.

NOCCSID (No Coded Character Set Identifier) keyword for display files

You use this field-level keyword to specify that CCSID conversion of the field is not done.

This keyword has no parameters.

The Character Data Representation Architecture (CRDA) specifies the '3F'X character as a replacement character. This character is also a field attribute definition for the 5250 data stream specification. Translation converting '3F'X character to '1F'X for output is done for all fields whether *JOBCCSID translation is active or inactive. Use the NOCCSID keyword to prevent translation at the field level.

If the NOCCSID keyword is not specified, conversion of the field continues normally.

Example

The following example shows how to specify the NOCCSID keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A
     A          R RECORD
     A            FIELD1         5A  B  2 10NOCCSID
     A
```

OPENPRT (Open Printer File) keyword for display files

You use this file-level keyword to specify that after the printer file is opened (the first time the user presses the Print key), it is to remain open until the associated display file is closed.

If you do not specify OPENPRT (and the PRINT keyword is specified), the printer file is opened and closed each time a display image is printed.

The printer file should be spooled if more than one job uses the same printer file and device. While the printer file is open in the nonspooled mode, the associated printer is allocated to the program or process using this function.

This keyword has no parameters.

This keyword is valid only if you have specified a file-level PRINT keyword with a printer file parameter. It is not valid with record-level PRINT keywords.

The OPENPRT keyword has no effect unless the PRINT file is specified on the PRINT keyword.

Option indicators are not valid for this keyword.

Example

The following example shows how to specify the OPENPRT keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00031A                                  PRINT(PRTFILE)
00032A                                  OPENPRT
     A
```

Related reference

PRINT (Print) keyword for display files
You use this file-level or record-level keyword to specify that the workstation user can press the Print key to print the current display.

OVERLAY (Overlay) keyword for display files

You use this record-level keyword to specify that the record format that you are defining should appear on the display without the entire display being deleted first.

This keyword has no parameters.

Normally, the entire display is deleted on each output operation. All records on the display with fields that partially or completely overlap fields in this record are deleted before this record is displayed; all others remain on the display and are not changed in any way. A record already on the display is deleted even if fields specified in the record format are not selected for display. For example, assume that the following records are on the display:

REC1 (lines 1 and 2)
REC2 (lines 3 and 4)
REC3 (line 5)
REC4 (line 9)

An output of REC5 (lines 4 and 5) with OVERLAY will leave the display with the following records:

REC1 (lines 1 and 2)
REC5 (lines 4 and 5)
REC4 (line 9)

If the record with the OVERLAY keyword in effect is already on the display and PUTOVR, PUTRETAIN, or CLRL keyword is not specified, it is deleted and rewritten as a new record.

When the beginning attribute character of a record overlaps the ending attribute character of a record already displayed, the attribute characters overlap each other in position 1 of a line. (The last field of the first record displayed ends in the farthest right display position of the preceding line.)

In the above example, however, if the only portion of REC2 on line 4 is the ending attribute character of the last field of REC2 (which occurs when the last displayed character of the last field of REC2 is in the last position of line 3), REC2 remains displayed following the display of REC5 with OVERLAY. The display will have the following records:

REC1 (lines 1 and 2)
REC2 (line3)
REC5 (lines 4 and 5)
REC9 (line 9)

The display is always deleted on the first output operation after the file is opened, except when both ASSUME and OVERLAY are specified.

OVERLAY is assumed by the IBM i operating system for ERRMSG, ERRMSGID, PUTOVR, and CLRL functions.

If OVERLAY is conditioned and not selected, then the ERASE, ERASEINP, MDTOFF, PROTECT, and PUTRETAIN keywords cannot take effect if they are selected, unless the PUTOVR keyword is selected. In such cases, the ERASE, ERASEINP, and MDTOFF keywords can take effect.

If you specify OVERLAY, you should also specify the RSTDSP(*YES) keyword on the CRTDSPF (Create Display File) or CHGDSPF (Change Display File) command. Otherwise, data on the display can be lost if the file is suspended.

To delete any of the records on the display, use the ERASE keyword to specify the names of the record formats to be deleted.

If you also specify the CLRL keyword, processing proceeds according to the CLRL specification, not the OVERLAY specification.

A warning message is sent at file creation time if the OVERLAY keyword is specified on a record with the DSPMOD keyword. At run time, the OVERLAY keyword is ignored when the display mode changes.

Option indicators are valid for this keyword.

Example

The following example shows how to specify the OVERLAY keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00021A          R RECL                      OVERLAY
     A
```

Related reference

WINDOW (Window) keyword for display files
You use this record-level keyword to specify that the record format you are defining will be displayed using a window.

OVRATR (Override Attribute) keyword for display files

You use this field-level or record-level keyword with the PUTOVR keyword to override the existing display attributes of a field or record that is already on the display.

The OVRATR keyword can be used with the OVRDTA keyword on the same field or record.

When OVRATR is specified at both the record and field level, the field level specification is used for that field.

See the Application Display Programming book for information about how to use OVRATR in files that are used in the System/36 environment.

This keyword has no parameters.

The display attributes that can be overridden by the OVRATR keyword are:

CHECK(ER)
CHECK(ME)
DSPATR (all except OID and SP)
DUP

When the OVRDTA keyword is in effect, the display attribute can also be overridden on the same output operation (as if the OVRATR keyword were also in effect).

When the OVRATR keyword is specified at the field level, it is valid only with the following types of fields:

- Input-only
- Output-only

- Input/output
- Constant

When the OVRATR keyword is specified at the record level, it applies to each of the following types of fields:

- Input-only
- Output-only
- Input/output
- Constant

Option indicators are valid for this keyword.

For a discussion and an example of how to use the OVRATR keyword, see "PUTOVR (Put with Explicit Override) keyword for display files" on page 176.

OVRDTA (Override Data) keyword for display files

You use this field-level or record-level keyword with the PUTOVR keyword to override the existing data contents of a field or record that is already on the display.

The OVRDTA keyword can be used with the OVRATR keyword on the same field or record.

When OVRDTA is specified at both the record and field level, the field-level specification will be used for that field.

This keyword has no parameters.

OVRDTA is required if the DFT keyword is specified for output-only or input/output fields.

When OVRDTA is specified at the field level, it is valid only with the following types of fields:

- Output-only
- Input/output
- Message

When the OVRDTA keyword is specified at the record level, it applies to each of the following types of fields:

- Output-only
- Input/output
- Message

Option indicators are valid for this keyword.

For a discussion and an example of how to use the OVRDTA keyword, see "PUTOVR (Put with Explicit Override) keyword for display files" on page 176.

PAGEDOWN/PAGEUP (Page Down/Page Up) keywords for display files

You use these file-level or record-level keywords to specify that your program handles any situation where the workstation user has pressed the Page Down or Page Up keys and the IBM i operating system cannot page through the display.

If this situation occurs and you have not specified this keyword (whichever one is appropriate), the IBM i operating system sends an error message indicating that the key is not valid at that time.

See "System/36 environment considerations for display files" on page 247 for special considerations when you specify the PAGEDOWN/PAGEUP keywords in files that are used in the System/36 environment.

The format for each of these keywords is:

```
PAGEDOWN[(response-indicator ['text'])]
PAGEUP[(response-indicator ['text'])]
```

You can specify a response indicator with these keywords. If you do, and the appropriate Page key is pressed, the IBM i operating system sets on the specified response indicator within the input record and returns control to your program after it processes the input data. If you do not specify a response indicator and the specified Page key is pressed, the IBM i operating system performs normal input record processing.

The optional text is included on the computer printout created at program compilation to explain the intended use of the indicator. This text functions only as a comment in the file or program. The single quotation marks are required. If you specify more than 50 characters between the single quotation marks, the text is truncated to 50 characters on the program computer printout.

These keys cause data to be returned from the display device to your program (similar to command function (CF) and Enter keys).

The ROLLUP keyword cannot be specified with PAGEDOWN. The ROLLDOWN keyword cannot be specified with PAGEUP.

Note: PAGEDOWN is the same as ROLLUP; PAGEUP is the same as ROLLDOWN.

If the operating system is performing the page function for subfiles (SFLSIZ value does not equal SFLPAG value), you do not need to specify these keywords. For a description of what happens when PAGEDOWN and PAGEUP are specified for a subfile, see "SFLROLVAL (Subfile Roll Value) keyword for display files" on page 217.

Option indicators are valid for these keywords.

Example

The following example shows how to specify the PAGEDOWN and PAGEUP keywords.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A N64                                PAGEUP(52 'Page Up')
     A                                    PAGEDOWN(61)
     A
```

Related reference

ALTPAGEDWN/ALTPAGEUP (Alternative Page Down/Alternative Page Up) keyword for display files
You use these file-level keywords to assign command function (CF) keys as alternative Page Down/Page Up keys.

PASSRCD (Passed Record) keyword for display files

You use this file-level keyword to specify the record format to be used by the IBM i operating system when another program passes unformatted data to your program.

The passed data is processed only if your program's first request after file open is an input operation without a record format name. The data must be processed according to this record format.

The format of the keyword is:

```
PASSRCD(record-format-name)
```

The record-format-name is a required parameter value for this keyword and must exist in the file. The following keywords cannot be specified on the record format:

ALWROL
CLRL
SLNO

Option indicators are not valid for this keyword.

Example

The following example shows how to specify the PASSRCD keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A                                  PASSRCD(RECKEEP)
00020A          R RECORD
     A
```

PRINT (Print) keyword for display files

You use this file-level or record-level keyword to specify that the workstation user can press the Print key to print the current display.

See "System/36 environment considerations for display files" on page 247 for special considerations when you specify the PRINT keyword in files that are used in the System/36 environment.

The format of the keyword is:

```
PRINT[(response-indicator ['text']) v (*PGM) v
  ([library-name/]printer-file-name)]
```

The following four examples illustrate the four ways you can specify the PRINT keyword:

`PRINT`	The IBM i operating system spools the output to printer file QSYSPRT unless you specify another printer file for the workstation on the PRTFILE parameter on the CRTDEVDSP or CHGDEVDSP commands. See PRINT keyword without parameter values.
`PRINT(01 'User presses Print key')`	Your program is given control and decides what to do (for example, produce formatted printer output). The response indicator is set on. No data is sent from the device.
`PRINT(*PGM)`	Control returns to your program when the Print key is pressed.
`PRINT(LIB1/PRINTFILE1)`	The IBM i operating system spools the output to the specified printer file (which can be defined through DDS or on the PRTFILE parameter on the CRTDEVDSP or CHGDEVDSP commands). An Override with Printer File (OVRPRTF) command, if in effect before the printer file is opened (when the Print key is pressed), can change the printer device name.

Further considerations of the ways to specify the PRINT keyword are discussed in the following sections.

If you specify the PRINT keyword in any form, the workstation user can print a display containing the message help. In this case, the print operation is performed as if the PRINT keyword were specified with no parameters.

PRINT keyword without parameter values for the local workstation

The IBM i operating system spools the output to the specified printer file (which can be defined through DDS or on the PRTFILE parameter on the CRTDEVDSP or CHGDEVDSP commands).

Nondisplay fields appear as blanks. Duplicated characters entered by pressing the Dup key appear as asterisks (*). Display attributes appear as blanks. If the print function cannot be performed successfully, the IBM i operating system attempts to complete the print function using the printer file specified on the

PRTFILE parameter on the CRTDEVDSP or the CHGDEVDSP command that is used to describe the display device to the system.

For workstation printers attached through the workstation controller, a message indicating that there is a problem is sent to the workstation user requesting the print function. The workstation user can make the printer ready or press the Reset key. To cancel a print request before it is complete, the workstation user can press and hold the Shift key, then press the Print key.

Note: After the current display is printed, the paper is advanced twice the number of lines as in the current display size (48 lines for a 24 x 80 display, and 54 lines for a 27 x 132 display).

PRINT keyword without parameter values for the remote workstation

The IBM i program attempts to print the display image on the associated workstation printer without sending the data through the system. The associated workstation printer is the printer device specified on the PRINTER parameter of the CRTDEVDSP or CHGDEVDSP command that is used to describe the local display device to the system.

If the printer is not ready when the Print key is pressed, no specific message is sent to the user. The workstation requesting the print function remains inoperable until the printer is made ready, or until the print request is canceled (by using the shifted Print key).

Note: After the current display is printed, the paper is advanced the same number of lines as in the current display size (24 lines for a 24 x 80 display, and 27 lines for a 27 x 132 display).

Option indicators are valid for this keyword.

Example: PRINT keyword with no parameter values

The following example shows how to specify the PRINT keyword with no parameter values.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A                                  PRINT
00020A          R RECORD1
     A
```

Example: PRINT keyword with a response indicator or *PGM special value

If you specify the PRINT keyword with a response indicator, the IBM i operating system returns control to your program with the specified response indicator set on. No data is received from the device. The keyboard is locked until your program sends another output operation to the display file. There is no difference in the print function between local and remote workstations. If you specify *PGM, the IBM i operating system returns control to your program. The only difference between these two forms is the response indicator; all other processing is the same.

The optional text for the response indicator form is included on the computer printout created at program compilation time to explain the intended use of the indicator. This text functions only as a comment. The single quotation marks are required. If you specify more than 50 characters between the single quotation marks, the text is truncated to 50 characters on the program computer printout.

The following example shows how to specify the PRINT keyword with a response indicator.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A                                  PRINT(01 'User presses Print key')
00020A          R RECORD1
     A
```

The following example shows how to specify the PRINT keyword with the *PGM special value.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A                                  PRINT(*PGM)
00020A          R RECORD1
     A
```

Example: PRINT keyword with a specified printer file

The IBM i program reads the display buffer and prints the display image using the specified printer file. The printer file that you specify as a parameter value for this keyword can be either an externally described or a program-described file. It also can be either spooled or nonspooled. If you specify an externally described printer file, it must contain a record format with the same name as the file.

The printer file must exist and be authorized to the user of the display when the Print key is pressed. This also applies to the library name if it is specified. If the IBM i operating system is unable to perform the print function on the specified printer file, it attempts to use the printer file specified on the PRTFILE parameter of the CRTDEVDSP or the CHGDEVDSP command. SPOOL(*YES) should be specified on the CRTPRTF or CHGPRTF command to prevent the keyboard from locking.

If you do not specify the library name, the current library list at program run time is used.

The following example shows how to specify that the display is to be directed to printer file, LIB1/PRINTFILE1.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A                                  PRINT(LIB1/PRINTFILE1)
00020A          R RECORD1
     A
```

Related reference

OPENPRT (Open Printer File) keyword for display files
You use this file-level keyword to specify that after the printer file is opened (the first time the user presses the Print key), it is to remain open until the associated display file is closed.

PROTECT (Protect) keyword for display files

You use this record-level keyword with the OVERLAY keyword to specify that when the record you are defining is displayed, all input-capable fields already on the display are to be changed to output-only fields.

This protects them from input typing. This keyword does not affect the record format in which it is specified. The data contents of the affected fields are not changed, but your program cannot read them unless it first displays the record formats again in which the input-capable fields are specified.

To protect a single field from input typing, see the DSPATR(PR) keyword.

This keyword has no parameters.

The OVERLAY keyword must be specified in the record format in which PROTECT is specified. Also, either the OVERLAY keyword or the CLRL keyword must be in effect for PROTECT to be in effect.

You can use PROTECT to protect input-capable fields in other records only on the first output operation for which you have selected PUTOVR. On subsequent output operations, PROTECT is in effect only if the PUTOVR keyword is not in effect.

If the ERASEINP and PROTECT keywords are both in effect for an output operation, the IBM i operating system first erases the input-capable fields specified on the ERASEINP parameter value, then protects all input-capable fields on the display from input typing.

A warning message appears at file creation time if the PROTECT keyword is specified on a record with the DSPMOD keyword. At run time, the PROTECT keyword is ignored when the display mode changes.

Option indicators are valid for this keyword.

Example

The following example shows how to specify the PROTECT keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A          R RECORD1
00020A            FLD1          5   I  5  3
00030A
00040A          R RECORD2                  OVERLAY
```

```
00050A  32                                  PROTECT
00060A            FLDA          10   I  6  3
     A
```

In this example, RECORD1 has an input-capable field that has been displayed and read and that should be left on the display while RECORD2 is displayed and read. To prevent further entries in FLD1 in RECORD1, send an output operation to RECORD2 with PROTECT in effect. When this is done, FLDA is not protected, but FLD1 is protected.

PSHBTNCHC (Push Button Field Choice) keyword for display files

You use this field-level keyword to define a choice for a push button field.

The format of the keyword is:

```
PSHBTNCHC(choice-number choice-text [command-key] [*SPACEB])
```

The choice-number parameter defines an identification number for this choice. This parameter is required. The choice number returns to the application to indicate which choice in the push-button field was selected. Valid values for the choice-number are positive integers greater than 0 and less than or equal to 99. Duplicate choice-number values within a push-button field are not allowed.

The choice-text parameter defines the text that appears in the push-button field for the choice. This parameter is required. The parameter can be specified in one of two forms:

- As a character string: 'Choice text '
- As a program-to-system field: &field-name

The field specified must exist in the same record as the selection field and must be defined as a character field with usage P.

The choice text must fit on one line of the display for the smallest display size specified for the file. The maximum length for the choice text depends on the following conditions:

- Position of the push-button field
- Length of the choice text
- Gutter width between choices
- Number of columns of choices
- Smallest display size
- Window width, if displayed in a window

Within the choice text, you can specify a mnemonic for the choice by using a greater than character (>) to indicate the mnemonic character. The character to the right of the > is the mnemonic. The mnemonic is used only on a character-based graphical display attached to a controller that supports an enhanced interface for nonprogrammable workstations. Examples of specifying mnemonics:

Choice text
Appears as

'F2=>File'
F2=File

'F3=F>inish'
F3=Finish

'>Enter'
Enter

In order to specify > as a character in the text, you must specify it twice, just as you must specify the apostrophe character twice in order to get a single apostrophe character in the text. For example:

Choice text
Appears as

'X >>= 1'
X >= 1

'X >>>= 1'
X >= 1

Note: You cannot specify the > as the mnemonic.

The mnemonic character indicated must be a single-byte character and must not be a blank. Only one mnemonic is allowed in the choice text, and the same mnemonic character should not be specified for more than one choice. If the same mnemonic character is used more than once than the first definition of the mnemonic is used.

The command-key parameter is optional and indicates which function key should be generated when this push-button choice is selected. The following keys can be used as parameters: CA01 to CA24, CF01 to CF24, PRINT, HELP, CLEAR, ENTER, HOME, ROLLUP, and ROLLDOWN. If the command-key specified is not defined at the file level for this record, then the key will be added to this record. If a parameter is not defined then ENTER will be used.

The *SPACEB parameter is optional and indicates that a blank spot where this choice will be located should be inserted before this choice. This parameter is used to specify logical grouping of choices.

When the PSHBTNCHC keyword is specified on a field, the PSHBTNFLD keyword must also be specified.

Several PSHBTNCHC keywords can be specified for one push-button field. The number of PSHBTNCHC keywords that can be specified depends on the position of the push-button field and the display size. More than one choice can occupy one line, and all choices must fit on the smallest display size specified for the file. The maximum number of choices is 99.

Option indicators are valid for this keyword. When a PSHBTNCHC keyword is off, the list of choices is compressed.

Push buttons always behave as if AUTOENT and AUTOSLT are on.

Example

The following example shows how to specify the PSHBTNCHC keyword:

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8

     A
     A          R RECORD
     A            :
     A            :
     A            F1             2Y 0B 24 02PSHBTNFLD
     A  01                                  PSHBTNCHC(1 '>Help' HELP)
     A                                      PSHBTNCHC(2 &F3 CA03)
     A                                      PSHBTNCHC(3 'E>nter')
     A            F3             4A  P
     A
     A
```

In this example, three choices are defined for the push-button field F1. The text for choice 2 is contained in field F3, and the mnemonic for choice 2 must be contained in the text supplied by the application at run time. If indicator 01 is off when the record is written, only choices 2 and 3 are displayed.

PSHBTNFLD (Push Button Field) keyword for display files

You use this field-level keyword to define a field as a push button field.

A push button field is a field that contains a fixed number of push buttons from which a user can select. The field appears as a list of command keys each enclosed with '<>' or as a group of push buttons.

The format of the keyword is:

```
PSHBTNFLD[([*NORSTCSR  v *RSTCSR]
[(*NUMCOL nbr-of-cols) v (*NUMROW nbr-of-rows)]
[(*GUTTER gutter-width)])]
```

The parameters are optional and can be entered in any order. When no parameter is specified, the push button field choices are arranged horizontally. *GUTTER parameter is set to 3 as default and the field will be displayed using as many lines as it takes to display all of the choices. There will be 3 spaces between each choice.

The RSTCSR parameter specifies whether the arrow keys should be allowed to move the selection cursor outside of the field. *RSTCSR specifies that the arrow keys will not cause the selection cursor to move outside of the push-button field. *NORSTCSR specifies that the arrow keys will cause the selection cursor to leave the field. The default is *NORSTCSR.

The *NUMCOL parameter specifies that this field should be displayed in multiple columns with the choices arranged across the columns in this manner:

```
< F1 >    < F2 >    < F3 >
< F4 >    < F5 >    < F6 >
< F7 >    < F8 >    < F9 >
```

Nbr-of-rows specifies how many rows the push-button field should contain. Nbr-of-rows must be a positive integer and the entire single-choice push-button field must be able to fit on the display when placed in the specified number of rows.

The *GUTTER parameter specifies the number of blanks to be placed between each column of the push-button field. Unlike the SNGCHCFLD keyword, it can be specified even if *NUMCOL or *NUMROW have not been specified. The gutter-width must be a positive integer. If *GUTTER is not specified, the default of gutter-width is set to three blanks. The gutter value must be a number greater than one.

For more information about how to support different device configurations, see the Application Display Programming book.

A field containing the PSHBTNFLD keyword must also contain one or more PSHBTNCHC keywords defining the choices for the field.

The field containing the PSHBTNFLD keyword must be defined as an input-capable field with data type Y, length equal to 2, and decimal positions of 0. The position specified for the field is the position of the first push-button choice. For input, the field contains the number of the choice selected, or 0 if no choice was selected. For output, the value of the field is ignored.

The following keywords can be specified on a field with the PSHBTNFLD keyword:

ALIAS	INDTXT
CHANGE	NOCCSID
CHCAVAIL	PSHBTNCHC
CHCUNAVAIL	DSPATR(PC)
CHCCTL	TEXT

Option indicators are not valid for this keyword.

Example

The following example shows how to specify the PSHBTNFLD keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A
     A          R RECORD
     A            :
     A            :
     A                                  2 40'MENU'
     A            F1             2Y 0B 24 02PSHBTNFLD
     A                                     PSHBTNCHC(1 'Cmd1' CF01)
     A                                     PSHBTNCHC(2 'Enter')
     A 01
     A
```

In this example, when using a graphical display station attached to a controller that supports an enhanced interface for non-programmable workstations, the push-button fields look like this:



PULLDOWN (Pull-Down Menu) keyword for display files

You use this record-level keyword to define a record as a pull-down menu for a menu bar.

When this record is written, it is saved by the system and displayed later as a pull-down menu for a menu-bar choice.

The format of the keyword is:

```
PULLDOWN[(*SLTIND | *NOSLTIND)]
[(*NORSTCSR | *RSTCSR)]
```

The parameters are optional.

The SLTIND parameter specifies whether the selection indicators (such as radio buttons) for a selection field in the pull-down menu are displayed. *SLTIND specifies that the selection indicators should be displayed. *NOSLTIND specifies that the selection indicators should not be displayed. The default is *SLTIND.

The RSTCSR parameter specifies if the user should be allowed limited function when the cursor is outside of the pull-down window. When *NORSTCSR is specified and the cursor is outside of the pulldown window, the user will be allowed to press a function key and have it function as if the cursor were within the window. When *RSTCSR is specified, if the user attempts to press a function key while the cursor is outside of the pulldown window, the user will receive a beep and the cursor will be placed inside the pulldown-window. Control will not be returned to the application. The default is *NORSTCSR.

A record containing the PULLDOWN keyword is considered a window record, although the WINDOW keyword cannot be used. The system calculates the dimensions of the pull-down window and generates the border.

The following keywords cannot be specified on a record with the PULLDOWN keyword:

ALARM	FRCDTA	OVRDTA
ALTNAME	HLPCLR	PUTOVR
ALWGPH	HLPSEQ	PUTRETAIN
ALWROL	INVITE	RTNDTA
ASSUME	INZRCD	SFL
CLEAR	MDTOFF	SLNO
CLRL	MNUBAR	USRDFN
ERASE	OVERLAY	WDWTITLE
ERASEINP	OVRATR	WINDOW

Option indicators are not valid for this keyword.

Example

The following example shows how to specify the PULLDOWN keyword:

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A
     A          R MENUBAR                   MNUBAR
     A            MNUFLD         2Y 0B  1  2
     A                                       MNUBARCHC(1 PULLFILE 'File       ')
     A                                       :
     A                                       :
```

```
     A          R PULLFILE                  PULLDOWN
     A            :
     A            :
     A
```

In this example, record PULLFILE is defined as a pull-down menu for a menu-bar choice. When record PULLFILE is written, the system saves it and displays it when it is selected from the menu bar. When the system displays the PULLFILE record, it calculates the dimensions needed for the pull-down window based on the contents of the PULLFILE record, and generates the pull-down border accordingly.

PUTOVR (Put with Explicit Override) keyword for display files

You use this record-level keyword to enable the override of either display attributes or data contents (or both) of specific fields within a record that is displayed on a workstation device. By using the PUTOVR keyword, you can reduce the amount of data sent to the display device.

This keyword has no parameters.

If you use the PUTOVR keyword and subfiles, certain restrictions apply. See the Application Display Programming book for more information about these restrictions and how to use PUTOVR in files that are used in the System/36 environment..

When selected fields in a record that has already been displayed are to be changed, an output or an input/output operation sent to the record with the PUTOVR, OVRDTA, and OVRATR keywords in effect changes only the fields for which the OVRDTA or OVRATR keyword is in effect. The OVRDTA keyword permits a change in the data contents of the field and the OVRATR keyword permits a change in the display attributes of the field. The way in which fields are to be changed is controlled by setting option indicators.

The following conditions cause the Put-Override keywords to be ignored and no error to occur:

- PUTOVR is not in effect at the time of the output operation.
- Neither the OVRDTA nor OVRATR keyword is in effect at the time of the output operation.
- The record format is not already on the display.

The PUTOVR and OVRDTA keywords must be specified when DFT is specified for a named output-capable field. When the PUTOVR and OVRDTA keywords are both in effect for a field, the default value specified with the DFT keyword is displayed only on the first display of the field. On subsequent displays with the PUTOVR and OVRDTA keywords in effect, the program value is displayed.

If a field is not displayed on the first output operation to a record format, certain restrictions apply. These restrictions apply when, on a subsequent output operation, the field is selected for display and the put-override keywords are also in effect:

- For output-only fields for which the OVRDTA or OVRATR keyword is selected, the IBM i operating system does not send an ending attribute character. Any display attributes (such as reverse image) are continued across the display until the beginning attribute character of the next field on the display. You should display output-only fields on the first output operation (perhaps with the DSPATR(ND) keyword so they cannot be seen) in order to provide an ending attribute character for later overrides.
- For input-capable or message fields for which the OVRDTA or OVRATR keyword is selected, the IBM i operating system sends an ending attribute character. This field must be displayed on the initial output operation.

The PUTRETAIN keyword and the PUTOVR keyword cannot be specified on the same record format.

The OVRDTA keyword is permitted only with output-only (usage O), input/output (usage B), program-to-system (usage P), or message (usage M) fields.

The OVRATR keyword is permitted only with output-only (usage O), input-only (usage I), or input/output (usage B) fields.

If you specify PUTOVR, you should also specify RSTDSP(*YES) on the Create Display File (CRTDSPF) or Change Display File (CHGDSPF) command. Otherwise, data on the display can be lost if the file is suspended.

A warning message is sent at file creation time if the PUTOVR keyword is specified on a record with the DSPMOD keyword. At run time, the PUTOVR keyword is ignored when the display mode changes.

The OVRATR keyword can be used only to override the following display attributes:

CHECK(ER)
CHECK(ME)
DUP
DSPATR (all except OID and SP)

An output operation with the OVRDTA keyword in effect does not need to have the OVRATR keyword in effect to override display attributes, as well as data contents, of the field or fields being overridden.

Option indicators are valid for the PUTOVR, OVRATR, and OVRDTA keywords.

Example

The following example shows how to specify the PUTOVR, OVRATR, and OVRDTA keywords.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A          R INVRCD                  PUTOVR
00020A            FLD1                 1 26'INVENTORY REMAINING IN WAREHOUSE 1'
00030A*
00040A                                 3  2'Remaining on hand:'
00050A                                    OVRATR
00060A  11                                DSPATR(HI)
00070A*
00080A            INVBAL         5Y 0     +2
00090A  12                                OVRDTA
00100A*
00110A                                   +2'Low on stock' OVRATR
00120A N70                                DSPATR(ND)
00130A  70                                DSPATR(HI)
00140A*
00150A            SUPPPL        20   B  5  2DFT('INTERNAL')
00160A  13                                OVRDTA
00170A*
00180A            ACCT          20      6  2OVRDTA
00190A                                    DSPATR(HI)
00200A  14                                DSPATR(RI)
     A
```

An initial output operation generates a full display of information; on a second output operation, PUTOVR is in effect and the program can set option indicators to make the following changes to the display:

- If option indicator 11 is set on, the *Remaining on hand:* constant field will be changed to a highlighted field. To reset the display attribute to normal, display the record format again with option indicator 11 off.
- If option indicator 12 is set on, the program can change the displayed value of the field INVBAL.
- If option indicator 70 is set off, the *Low on stock* constant field is a nondisplay field. If option indicator 70 is set on, the field is changed to a highlighted field.
- If option indicator 13 is set on, the program can set the value of the field SUPPL to override the default value (INTERNAL). The first displayed value is always INTERNAL; to display the value INTERNAL again after changing it to something else, the program must set the value of the field to INTERNAL before displaying it again.
- If option indicator 14 is set on, the display attribute of the field ACCT is changed from highlight to highlight and reverse image at the same time that new data is sent to the field. If option indicator 14 is set off, the display attribute is changed back to highlight. New data is sent to the display on each output operation.

Related reference

DFT (Default) keyword for display files

You use this field-level keyword to specify the constant value for constant fields (unnamed fields) and to specify a default value for named fields.

PUTRETAIN (Put-Retain) keyword for display files

You use this record-level or field-level keyword with the OVERLAY keyword to prevent the IBM i operating system from deleting data that is already on the display when the system displays the record again. The PUTOVR keyword has a function similar to, but more effective than, the PUTRETAIN keyword.

This keyword has no parameters.

To understand what effect this keyword has on output operations, consider the following sequence of steps:

1. Your program sends an output operation to RECORD1, displaying RECORD1. PUTRETAIN, if in effect, is ignored. Any data in the record area for RECORD1 is deleted before RECORD1 is displayed.
2. At some later time, with RECORD1 still on the display, your program sends a second output operation to RECORD1. Two conditions can occur:
   - If the PUTRETAIN keyword is not in effect, the IBM i operating system first deletes the record area for RECORD1, then displays RECORD1. Fields selected for display at this time are displayed with new data contents and new display attributes, which can be the same as before. The record area includes every line on which a field or part of a field for RECORD1 appears.
   - If PUTRETAIN is in effect, the IBM i operating system does not delete the record area for RECORD1. The data contents of selected fields are not changed. However, the display attributes for selected fields are sent to the display and can be changed (by selecting which DSPATR keyword is in effect for this output operation). Fields not selected for display are written over character-by-character by fields selected for display. (For more information, see When fields are selected by option indicators.)

Note: When using the field-level PUTRETAIN keyword, the entire record area is deleted if none of the fields in the record has PUTRETAIN optioned on. If you specify at least one field with unoptioned field-level PUTRETAIN keyword, this ensures that the record area is not deleted.

If you specify the PUTRETAIN keyword, you should also specify RSTDSP(*YES) on the Create Display File (CRTDSPF) or Change Display File (CHGDSPF) command. Otherwise, data on the display can be lost if the file is suspended.

Option indicators are valid for this keyword.

Conditions affecting the PUTRETAIN keyword

PUTRETAIN applies only to the record format for which it is specified, and then only if the record is already displayed. If the record on which PUTRETAIN is specified is not on the display, PUTRETAIN is ignored.

If you specify this keyword at the record level, the keyword applies to all fields in the record format that are selected for display.

This keyword can be specified for more than one field of a record format, but only once per field. This keyword can be specified at the record level and at the field level within the same record format.

PUTRETAIN cannot be specified with the PUTOVR keyword.

A warning message appears at file creation time if the PUTRETAIN keyword is specified on a record with the DSPMOD keyword. At run time, the PUTRETAIN keyword is ignored when the display mode changes.

The OVERLAY keyword must be specified whenever PUTRETAIN is specified.

If the OVERLAY keyword is not in effect, PUTRETAIN is ignored and the entire display deleted before the record is displayed.

When fields are selected by option indicators

When PUTRETAIN is in effect on an output operation involving field selection, fields in the record format that are not selected for redisplay are not deleted; they can be partially or completely rewritten by newly selected fields.

If PUTRETAIN is in effect only for a newly selected field (specified at the field level), only the beginning attribute character of the field is sent to the display; the ending attribute character is not sent. For fields without PUTRETAIN in the same record format, the IBM i operating system sends the display attribute and the data. If PUTRETAIN is in effect for the whole record (specified at the record level), only the beginning and ending attribute characters are sent to the display. Thus, the display attribute of a field can be reset to normal if the field immediately preceding this field is selected and this field is not selected.

For example, assume that the DSPATR(UL) keyword is in effect for two consecutive fields with overlapping attribute characters. If on an output operation with PUTRETAIN in effect, the first of these fields is selected and the second field is not selected, the display attribute of the second field is reset to the normal display attribute. This is because the IBM i operating system sends the first field to the display with beginning and ending attribute characters, and its ending attribute character overrides the beginning attribute character of the second field.

Example

The following example shows how to specify the PUTRETAIN keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00101A          R CUST
00102A                                  PUTRETAIN OVERLAY
     A
```

RANGE (Range) keyword for display files

You use this field-level keyword for input-capable fields. The RANGE keyword directs the IBM i operating system to perform validity checking on the data that the workstation user types into the field.

The data typed in must be greater than or equal to the lower value, and less than or equal to the higher value. Note that the IBM i operating system performs this checking only if the field is changed by the workstation user or if its changed data tag (MDT) is set on using DSPATR(MDT).

Note: See “CHKMSGID (Check Message Identifier) keyword for display files” on page 63 for information about defining user-specified error messages.

The format of the keyword is:

```
RANGE(low-value high-value)
```

When the field is a character field, the parameter values must be enclosed in single quotation marks. When the field is numeric, single quotation marks must not be specified.

You cannot specify RANGE on a floating-point field (F in position 35).

Option indicators are not valid for this keyword.

Example

The following example shows how to specify the RANGE keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A          R RECORD1
00020A* Character fields
00030A            FIELD1         1   I  2  2RANGE('B' 'F')
00040A            FIELD2         1   I  3  2RANGE('2' '5')
00050A* Numeric fields
00070A            FIELD3         1  0I  4  2RANGE(2 5)
00080A            FIELD4         4  0B  5  2RANGE(1 1500)
00090A            FIELD5         7  2B  6  2RANGE(100 99999.99)
00100A            FIELD6         3  0B  7  2RANGE(-100 -50)
```

```
00110A            FIELD7         3  2I  8  2RANGE(.50 1.00)
00120A            FIELD8         3  2I  9  2RANGE(.5 1)
00130A            FIELD9         5Y 2I 10  2RANGE(.01 999.99)
     A
```

FIELD7 and FIELD8 have equivalent RANGE parameter values. The reason is that for numeric fields, decimal alignment is based on the number of decimal positions specified in positions 36 through 37. For FIELD7 and FIELD8, the low value is 0.50 and the high value is 1.00.

Data entered into a numeric field is aligned on the decimal positions specified (in positions 36 through 37), and leading and trailing blanks are filled with zeros. For example, if 1.2 is typed into FIELD9, 00120 is returned to your program. If 100 is typed into FIELD9, 10000 is returned to your program.

REF (Reference) keyword for display files

You use this file-level keyword to specify the name of a file from which field descriptions are to be retrieved. You can also use this keyword when you want to duplicate descriptive information from several fields in a previously described record format.

You can code the file name once here rather than on REFFLD keywords with each of the field descriptions that refer to the file. To refer to more than one file, use the REFFLD keyword. (REF can be specified only once.)

The format of the keyword is:

```
REF([library-name/]database-file-name [record-format-name])
```

If there is more than one record format in the referenced file, specify a record format name as a parameter value for this keyword to tell the IBM i operating system which one to use unless the record formats should be searched sequentially.

The database-file-name is a required parameter for this keyword. The library-name and the record-format-name are optional.

If you do not specify the library name, the current library list (*LIBL) at file creation time is used. If the record-format-name is not specified, each record format is searched in order (as they are specified). The first occurrence of the field name is used. See When to specify REF and REFFLD keywords for DDS files for the search sequences determined by your choice of REF and REFFLD keywords.

You can specify a distributed data management (DDM) file on this keyword.

When using a DDM file, the database-file-name and library-name are the DDM file and library names on the source system. The record-format-name is the record format name in the remote file on the target system.

Note: IDDU files cannot be used as reference files.

Option indicators are not valid for this keyword.

Examples

The following examples show how to specify the REF keyword.

In this example, FLD1 has the same attributes as the first (or only) FLD1 in FILE1.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A                                  REF(FILE1)
00020A          R RECORD
00030A            FLD1      R           2  2
     A
```

In this example, FLD1 has the same attributes as FLD1 in RECORD2 in FILE1 in LIB1.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A                                  REF(LIB/FILE1 RECORD2)
00020A          R RECORD
```

```
00030A            FLD1      R           2  2
     A
```

Related concepts

Reference for display files (position 29)

You can specify R in this position to use the reference function of the IBM i operating system. This function copies the attributes of a previously defined, named field (called the *referenced field*) to the field you are defining.

REFFLD (Referenced Field) keyword for display files

You use this field-level keyword to refer to a field when the name, record format, file, or library of the referenced field differs from its equivalent in positions 19 through 28.

Use this field-level keyword when referring to a field under one of these conditions:

- The name of the referenced field is different from the name in positions 19 through 28.
- The name of the referenced field is the same as the name in positions 19 through 28, but the record format, file, or library of the referenced field is different from that specified with the REF keyword.
- The referenced field occurs in the same DDS source file as the referencing field.

The format of the keyword is:

```
REFFLD([record-format-name/]referenced-field-name
[{*SRC | [library-name/]database-file-name}])
```

The referenced-field-name is required even if it is the same as the referencing field. Use the record format name when the referenced file contains more than one record format. Use *SRC (rather than the database-file-name) when the referenced field name is in the same DDS source file as the referencing field. *SRC is the default value when the database-file-name and library-name are not specified and the REF keyword is not specified at the file level.

Note: When you refer to a field in the same DDS source file, the field you are referring to must precede the field you are defining.

Specify the database-file-name (with its library-name, if necessary) to search a particular database file.

If, in the same DDS source file, you specify the REF keyword at the file level and REFFLD at the field level, the particular search sequence depends on both the REF and REFFLD keywords.

You must specify an R in position 29. In some cases, some keywords specified with the field in the database file are not included in the display file.

You can specify a distributed data management (DDM) file on this keyword.

When using a DDM file, the database-file-name and library-name are the DDM file and library names on the source system. The referenced-field-name and the record-format-name are the field name and the record format name in the remote file on the target system.

Note: IDDU files cannot be used as reference files.

Option indicators are not valid for this keyword.

Example

The following example shows how to specify the REFFLD keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A          R FMAT1
00020A            ITEM           5      3  1
00030A            ITEM1     R            5  2REFFLD(ITEM)
00040A            ITEM2     R            5 12REFFLD(FMAT1/ITEM)
00050A            ITEM3     R            5 22REFFLD(ITEM FILEX)
00060A            ITEM4     R            5 32REFFLD(ITEM LIBY/FILEX)
00070A            ITEM5     R            5 42REFFLD(FMAT1/ITEM LIBY/FILEX)
```

```
00080A            ITEM6     R           5 52REFFLD(ITEM *SRC)
     A
```

Related concepts

Reference for display files (position 29)
You can specify R in this position to use the reference function of the IBM i operating system. This function copies the attributes of a previously defined, named field (called the *referenced field*) to the field you are defining.

When to specify REF and REFFLD keywords for DDS files

RETKEY (Retain Function Keys) and RETCMDKEY (Retain Command Keys) keywords for display files

You use these record-level keywords to indicate that function keys, command function (CF*nn*) keys, or command attention (CA*nn*) keys, which are enabled on a display, should be retained when the record you are defining is displayed.

These keywords have no parameters.

See “System/36 environment considerations for display files” on page 247 for information about how to specify the RETKEY and RETCMDKEY keywords.

RETLCKSTS (Retain Lock Status) keyword for display files

You use this record-level keyword to specify that the system should not unlock the keyboard on the next input operation. This keyword prevents the loss of data when an input operation is started and data is already being transmitted from the keyboard.

This keyword has no parameters. Option indicators are valid for this keyword.

Notes:

- Normally an input operation explicitly unlocks the keyboard, even if it is already unlocked. Any data being transmitted from the keyboard at the time of the unlock can be lost.
- Use this keyword only when the keyboard is already unlocked. Use of this keyword when the keyboard is locked can prevent input from the keyboard, because the unlock does not occur on the input operation. The workstation remains in an input-inhibited state.

Example

The following example shows how to specify the RETLCKSTS keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A
00020A          R REC1                      INVITE
00030A  10                                  RETLCKSTS
     A
```

If indicator 10 is on when record REC1 is put to the display, the keyboard is not explicitly unlocked by the system when the display device is invited.

RMVWDW (Remove Window) keyword for display files

You use this record-level keyword to remove all existing windows on the display before this record is displayed.

This keyword has no parameters.

When the RMVWDW keyword is specified, a WINDOW keyword must be specified on the same record format. The RMVWDW keyword functions only when the WINDOW keyword defines a window. The RMVWDW keyword does not function if the WINDOW keyword specified a record format name.

Option indicators are valid for this keyword.

Example

The following example shows how to specify the RMVWDW keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A          R WINDOW1                   WINDOW(6 15 9 30)
     A            FIELD1         5A  B  2  2
     A            FIELD2        20A  B  8  5
     A          R WINDOW2                    WINDOW(&LINE &POS 9 30)
     A  01                                   RMVWDW
     A            FIELD3         5A  B  2  2
     A            FIELD4        20A  B  8  5
     A            LINE           2S  P
     A            POS            3S  P
     A
```

WINDOW1 is already on the display. If indicator 01 is set on and WINDOW2 is written to the display, WINDOW1 is removed before WINDOW2 is displayed. If indicator 01 is off when WINDOW2 is written to the display, WINDOW1 remains on the display when WINDOW2 is displayed.

ROLLUP/ROLLDOWN (Roll Up/Roll Down) keywords for display files

You can use these file-level or record-level keywords to specify that your program handles any situation where the workstation user has pressed the Roll Up or Roll Down keys and the IBM i operating system cannot move the text lines on the display.

If this situation occurs and you have not specified this keyword (whichever one is appropriate), the IBM i operating system sends an error message indicating that the key is not valid at that time.

See “System/36 environment considerations for display files” on page 247 for special considerations when you specify the ROLLUP and ROLLDOWN keywords in files that are used in the System/36 environment.

The format for each of these keywords is:

```
ROLLUP[(response-indicator ['text'])]
ROLLDOWN[(response-indicator ['text'])]
```

You can specify a response indicator with these keywords. If you do, and the appropriate paging key is pressed, the IBM i operating system sets on the specified response indicator within the input record and returns control to your program after it processes the input data. If you do not specify a response indicator and the specified paging key is pressed, the IBM i operating system performs normal input record processing.

The optional text is included on the computer printout created at program compilation to explain the intended use of the indicator. This text functions only as a comment in the file or program. The single quotation marks are required. If you specify more than 50 characters between the single quotation marks, the text is truncated to 50 characters on the program computer printout.

These keys cause data to be returned from the display device to your program (similar to command function (CF) and Enter keys).

The PAGEDOWN keyword cannot be specified with ROLLUP. The PAGEUP keyword cannot be specified with ROLLDOWN.

Note: The ROLLUP keyword is the same as the PAGEDOWN keyword. The ROLLDOWN keyword is the same as the PAGEUP keyword. Roll is the same as page.

If the operating system is performing the paging function for subfiles (SFLSIZ value does not equal SFLPAG value), you do not need to specify these keywords. For a description of what happens when the ROLLUP and ROLLDOWN keywords are specified for a subfile, see “SFLROLVAL (Subfile Roll Value) keyword for display files” on page 217.

Option indicators are valid for these keywords.

Example

The following example shows how to specify the ROLLUP and ROLLDOWN keywords.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A N64                                ROLLDOWN(52 'Roll Down')
     A                                    ROLLUP(61)
     A
```

RTNCSRLOC (Return Cursor Location) keyword for display files

You use this record-level keyword to return the location of the cursor to an application program.

This keyword can be specified in two formats. These formats are:

- Return the name of the record and field in which the cursor is currently positioned. Optionally, a third parameter can be specified that will contain the relative cursor position within the field.
- Return the row and column position of the cursor relative to the display. Optionally, two additional parameters can be provided that will return either the row and column position of the cursor relative to the active window (if one exists) or the location of the cursor at the beginning of the two-event mouse button definition.

The formats of the keyword is:

```
RTNCSRLOC([*RECNAME]
&cursor-record &cursor-field
[&cursor-position])
or
RTNCSRLOC({*WINDOW | *MOUSE}
&cursor-row &cursor-column
[&cursor-row2
[&cursor-column2]])
```

The parameter for the first format are:

- The *RECNAME parameter indicates that RTNCSRLOC should return the name of the record and field on which the cursor is positioned. Optionally, it will also return the relative position of the cursor with the field. This parameter is optional.
- The cursor-record parameter specifies the name of a hidden field that, on input, will contain the name of the record on which the cursor is located. The field must be defined in the record format as a character (A in position 35) field of length 10, with usage H (hidden). If the cursor is not in a record area on input, the cursor-record field will contain blanks.
- The cursor-field parameter specifies the name of a hidden field that, on input, will contain the name of the field on which the cursor is located. The field must be defined in the record format as a character (A in position 35) field of length 10, with usage H (hidden). If the cursor is not located on a field on input, the cursor-field field will contain blanks.
- The optional cursor-position parameter specifies the name of a hidden field that, on input, will contain the relative position of the cursor within the field on which it is located. The field must be defined in the record format as a signed numeric (S in position 35) field of length 4, with 0 decimal positions and usage H (hidden). If the cursor is in the first position of the field, the cursor-position field will contain the value 1. If the cursor is in the *i*th position, the cursor-position field will contain the value *i*. If the cursor is not located on a field, the cursor-position field will contain the value 0. If the cursor is located in a menu-bar or selection-field list, then the cursor position parameter returns to the choice number on which the cursor is located.

All three fields specified on the RTNCSRLOC keyword will contain values on input if the cursor is outside the area of the record that contains the RTNCSRLOC keyword. The fields also contain values on input if the cursor is located in a subfile. The cursor-record field will contain a value if the cursor is located anywhere inside the subfile. The cursor-field and cursor-position fields will contain values if the cursor is located on a field within the subfile.

The parameters for the second format are:

- The *WINDOW or *MOUSE parameter is used to qualify the cursor-row2 and cursor-column-2 parameters. *WINDOW causes these parameters to return the cursor location relative to the first usable location in the active window. *MOUSE causes these parameters to return the location of the cursor just before a two-event mouse definition is processed.
- The cursor-row parameter specifies the name of a hidden field that, on input, contains the number of row on which the cursor is located. The field must be defined in the record format as a data type S, field length of 3, usage of H, and zero decimal positions. The value returned in this hidden field will be relative to the entire display where the first row of the display is row 1.
- The cursor-column parameter specifies the name of a hidden field that, on input, contains the number of the column on which the cursor is located. The field must be defined in the record format as a data type S, field length of 3, usage of H, and zero decimal positions. The value returned in this hidden field will be relative to the entire display where the first column of the display is column 1.
- The optional cursor-row2 parameter specifies the name of a hidden field. If *WINDOW is specified as the first parameter, the hidden field will contain the relative row position of the cursor to the first usable location of the active window. If there is no active window, this value will be the same as &cursor-row. If the cursor is in the first usable position of the window, the cursor-row2 field will contain the value 1. If the cursor is outside of the active window, it is possible for this value to be a negative number. If *MOUSE is specified as the first parameter, the hidden field contains the row number of the cursor at the instant just before a two-event mouse definition is called. If a two-event mouse definition has not been processed, this field will be set to zero. The field must be defined in the record format as data type S, field length of 3, usage H, and zero decimal positions.
- The optional cursor-column2 parameter specifies the name of a hidden field. If *WINDOW is specified as the first parameter, the hidden field will contain the relative column position of the cursor to the first usable location in the active window. If there is no active window, this value will be the same as &cursor-column. If the cursor is in the first usable position of the window, the cursor-column2 field will contain the value 1. If the cursor is outside of the active window, it is possible for this value to be a negative number. If *MOUSE is specified as the first parameter, the hidden field will contain the column number of the cursor when the first event of a two-event mouse definition has occurred. If a two-event mouse definition has not been processed, this field will be set to zero. The field must be defined in the record format as data type S, field length of 3, usage of H, and zero decimal positions.

Both formats of this keyword can be specified with the same record. If the same hidden field is used multiple times for any of the parameters, unpredictable results will occur.

Option indicators are not valid for this keyword.

Example

The following example shows how to specify the RTNCSRLOC keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A          R REC01                       RTNCSRLOC(&RCD &FLD &POS);
     A                                        RTNCSRLOC(*MOUSE &ROW &COL);
     A            FLD           10A  H
     A            RCD           10A  H
     A            POS            4S 0H
     A            FLD1A          2A  I  3  2
     A  10        FLD2A          6A  O  3 18
     A N10        FLD3A         10A  O  3 18
     A*
     A          R REC02                        OVERLAY
     A            FLD1A          2A  I  5  2
     A            FLD2A         10A  O  5  5
     A            FLD3A          6A  O  5 18
     A
```

Both REC01 and REC02 are displayed on the screen and option indicator 10 is off.

The following table shows the values returned when the cursor is at the specified positions.

Row	Col	Cursor record	Cursor field	Cursor position	Cursor row	Cursor column
3	2	REC01	FLD1A	1	3	2
3	19	REC01	FLD3A[1]	2	3	19
3	25	REC01	FLD3A	8	3	25
3	40	REC01	blanks	0	3	40
4	40	blanks	blanks	0	4	40
5	5	REC02	FLD2A	1	5	5
5	40	REC02	blanks	0	5	40

[1] If Option indicator 10 were on, FLD2A would be returned when the cursor is at row 3 column 19.

RTNDTA (Return Data) keyword for display files

You use this record-level keyword to specify that when your program sends an input operation to this record format, the IBM i operating system is to return the same data that was returned on the previous input operation sent to this record format.

The RTNDTA keyword is ignored if the record format has not already been read. When the RTNDTA keyword is in effect, your program can reread data on the display without requiring the IBM i operating system to actually pass data from the display device to your program.

This keyword has no parameters.

The RTNDTA keyword is ignored in the following situations:

- On the input portion of an input/output operation (Put-Get operation)
- On an input operation that is preceded by an output operation to the same record format

The RTNDTA keyword has effect only on an input operation sent to the same record format without an intervening output operation to that record format.

You can use RTNDTA as follows:

- Use RTNDTA to allow a main program to read a record format that is changed by a workstation user. The data read tells the main program which subprogram to call. The subprogram sends an input operation to the same record format, with RTNDTA in effect, to read the same data. This procedure can substitute for passing parameters to subprograms.

  Note: SHARE(*YES) must be specified for both display files.
- Use RTNDTA to allow an RPG III program to perform file maintenance with less locking of records in the database. For instance, the program reads a database record and displays the record at the display device. The workstation user reviews the record, makes any required changes, and presses the Enter key. While the workstation user is making changes, the database record, if locked, is unavailable to other programs. Hence it is recommended to leave the database record unlocked. However, when the program reads the record from the display and updates the database record, the database record overlays the internal representation of the display record in the program. Instead of preventing the overlay by using different field names for the display record and the database record, the program rereads the display file. With RTNDTA specified, the program retrieves the display record again and can then finish updating the database.

If the UNLOCK keyword is specified, the RTNDTA keyword cannot be specified.

Option indicators are not valid for this keyword.

Example

The following example shows how to specify the RTNDTA keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A          R RECORD1                  RTNDTA
00020A            FLD1           5   I  2  2
00030A            FLD2           5   B  3  2
     A
```

SETOF (Set Off) keyword for display files

You use this record-level keyword to specify that when an input operation sent to this record format is completed, the specified response indicator is to be set off.

See “System/36 environment considerations for display files” on page 247 for information about how to specify the SETOF keyword in files that are used in the System/36 environment.

The format of the keyword is:

```
SETOF(response-indicator ['text'])
```

The optional text is included on the computer printout created at program compilation to explain the intended use of the indicator. This text functions only as a comment in the file or program. The single quotation marks are required. If you specify more than 50 characters between the single quotation marks, the text is truncated to 50 characters on the program listing.

This keyword can be used to cause an option indicator that is on for an output operation to be returned in the off condition when the next input operation to the record is completed. (If no input operation is performed, the response indicator remains unchanged.) Your program does not need to turn off the indicator.

SETOF is equivalent to the SETOFF keyword.

Any indicator is valid for this keyword. It does not need to be previously defined as an option or a response indicator. The indicator becomes a response indicator when you specify SETOF.

If the indicator used with the SETOF keyword is also used with another keyword, such as CHANGE, the on/off status of the indicator is controlled by the other keyword.

Option indicators are not valid for this keyword.

Example

The following example shows how to specify the SETOF keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A          R CUSMST                   SETOF(63 'On=display MSG2000 +
00020A                                        CONSOLEMSG')
00030A            QRYORD         3  0I  5  3
00040A  63                                 ERRMSGID(MSG2000 CONSOLEMSG)
     A
```

A MSG2000 message is displayed on the message line when the program sends an output operation to CUSMST with indicator 63 set on. On the next input operation to CUSMST, the SETOF keyword sets off indicator 63. (Indicator 63 is used as both an option and a response indicator.)

Related reference

SETOFF (Set Off) keyword for display files
The SETOFF keyword is equivalent to the SETOF keyword.

SETOFF (Set Off) keyword for display files

The SETOFF keyword is equivalent to the SETOF keyword.

The format of the keyword is:

```
SETOFF(response-indicator ['text'])
```

The SETOF keyword is preferred.

Related reference

SETOF (Set Off) keyword for display files

You use this record-level keyword to specify that when an input operation sent to this record format is completed, the specified response indicator is to be set off.

SFL (Subfile) keyword for display files

You use this record-level keyword to specify that this record format is to be a subfile record format.

This record format (including its related field descriptions) must immediately precede the subfile-control record format (identified by the SFLCTL keyword).

This keyword has no parameters.

At least one displayable field must be specified in the subfile record format, unless the subfile is a message subfile (SFLMSGRCD keyword). The locations specified for fields in this record format are the locations on the display where the first subfile record in any one page of the subfile is displayed. The remaining part of the page of records is displayed below the first record.

The number of records in a page is determined by the parameter value specified for the SFLPAG keyword.

Displayable fields specified on the subfile-control record format can be displayed at the same time as subfile records. However, fields specified in the subfile-control record format cannot overlap fields specified in the subfile record format, even if they are specified with option indicators.

Overlap errors can occur if the first field of either the subfile record or the subfile control record starts in position 1. A field starting in position 1 has a beginning attribute byte on the previous line. Therefore, the previous line is also part of the record format.

The number of subfiles (each having one SFL and one SFLCTL keyword specified) that can be specified in a display file is limited only by the number of record formats permitted in a display file (1024 record formats, or 512 subfiles, maximum). Twelve subfiles can contain active records or be displayed at one time.

For display size *DS3, the field will wrap to the next line if the field extends beyond column 80.

Option indicators are not valid for this keyword.

Besides SFL, the following keywords are also valid on the subfile record format:

- For message subfiles:

  SFLMSGRCD (required at the record level)
  SFLMSGKEY (required at the field level)
  SFLPGMQ

- For all other subfiles (at the record level):

  CHANGE
  CHECK(AB)
  CHECK(RL)
  CHGINPDFT
  INDTXT
  KEEP
  LOGINP
  LOGOUT
  SETOF
  SETOFF
  SFLNXTCHG
  TEXT

  The following otherwise valid keywords are not valid at the field level when specified for the subfile record format:

DATE	ERRMSGID
DFTVAL	MSGID
ERRMSG	TIME

Example

The following example shows how to specify the SFL keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A          R SFLR                      SFL
     A*
00020A*           (at least one displayable field)
     A*
00030A          R SFLCTLR                   SFLCTL(SFLR)
00040A                                      SFLPAG(17)
00050A                                      SFLSIZ(17)
00060A                                      SFLDSP
00070A                                      SFLDSPCTL
     A
```

Related reference

SFLMSGRCD (Subfile Message Record) keyword for display files
You use this record-level keyword on the subfile record format to specify that this subfile is to be a message subfile and that the records displayed when the subfile is displayed are messages from a program message queue.

SFLPAG (Subfile Page) keyword for display files
You use this record-level keyword on the subfile-control record format to specify the number of records in the subfile to be displayed at the same time.

SFLCHCCTL (Subfile Choice Control) keyword for display files

You use this field-level keyword on a selection list to control the availability of choices for the list.

The format of the keyword is SFLCHCCTL.

When the SFLCHCCTL keyword is specified on a field, that field will be considered the control field for that record. That field must be the first field defined in the subfile record. That field must have a length of 1, data type of Y, decimal positions of zero, and have a usage of H. That field must be defined as the first field in the subfile. The control field works as follows:

Table 11. Control field for the SFLCHCCTL keyword

Control value	Meaning on output	Meaning of input
0	Available	Not selected
1	Selected	Selected
2	Unavailable. Cannot place cursor on choice unless help for choice is available.[1]	
3	Unavailable. Placing cursor on choice is allowed.	
4	Unavailable. Cannot place cursor on choice even if help for the choice is available.[1]	

[1] Applies only to displays attached to a controller that supports an enhanced interface for nonprogrammable workstations.

Option indicators are not valid for this keyword.

SFLNXTCHC keyword cannot be specified in a record that contains a field with the SFLCHCCTL keyword.

Only one SFLCHCCTL keyword can be used in one subfile record.

Example

The following example shows how to specify the SFLCHCCTL keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A
     A          R SFLRCD                    SFL
     A            CTLFLD         1Y 0H      SFLCHCCTL
     A            F1             4A  O  6 10
     A          R SFLCTLRCD                 SFLCTL(SFLRCD)
     A                                      SFLMLTCHC
     A                                      SFLPAG(5) SFLSIZ(&SFLSIZ);
     A                                      SFLDSP SFLDSPCTL
     A                                      ROLLUP(10)
     A   10                                 SFLEND(*SCRBAR)
     A            F3             5S 0H      SFLSCROLL
     A            F2             4S 0H      SFLRCDNBR(CURSOR *TOP)
     A            SFLSIZ         5S 0P
     A                                 1 30'Panel Title'
     A                                 4  5'Multiple selection list:'
```

SFLCLR (Subfile Clear) keyword for display files

You use this record-level keyword on the subfile-control record format so that your program can clear the subfile of all records. This keyword differs from the SFLDLT keyword in that the subfile is not deleted.

It differs from the SFLINZ keyword in that after being cleared, the subfile contains no data. Clearing the subfile does not affect the display. However, after being cleared, the subfile contains no active records.

This keyword has no parameters.

When active records already exist in the subfile and all are to be replaced, your program can send an output operation to the subfile-control record format after selecting SFLCLR. This clears the subfile and permits your program to write new records to the subfile (by issuing output operations to the subfile record format while incrementing the relative record number). Issuing an output operation to an already active subfile record causes an error message to be returned to your program.

If SFLCLR is in effect on an output operation and no records exist in the subfile, SFLCLR is ignored.

This optional keyword is valid only for the subfile-control record format. Display size condition names are not valid for this keyword.

An option indicator is required for this keyword to prevent the IBM i operating system from clearing the subfile on every output operation to the subfile-control record format.

Example

The following example shows how to specify the SFLCLR keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A          R SFLR                     SFL
     A*
     A*            (at least one displayable field)
     A*
00020A          R SFLCTLR                  SFLCTL(SFLR)
00030A                                     SFLPAG(17)
00040A                                     SFLSIZ(17)
00050A  01                                 SFLDSP
00060A  01                                 SFLDSPCTL
00070A N01                                 SFLCLR
     A
```

The subfile is displayed when option indicator 01 is set on for an output operation to SFLCTLR, and the subfile is cleared when option indicator 01 is set off for an output operation to SFLCTLR. Normally, the

option indicators specified for SFLCLR are the reverse of the option indicators specified for the SFLDSP and SFLDSPCTL keywords.

SFLCSRPRG (Subfile Cursor Progression) keyword for display files

You use this field-level keyword to define that when the cursor leaves the field, it goes to the same field in the next subfile record instead of the next field in the same record.

The SFLCSRPRG keyword is ignored when the workstation is not attached to a controller that supports an enhanced data stream.

This keyword has no parameters.

Option indicators are not valid for this keyword.

The SFLLIN keyword is not allowed in a record that contains the SFLCSRPRG.

Example

The following example shows how to specify the SFLCSRPRG keyword:

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A
     A          R SFL01                       SFL
     A            S1            10A  B  5  5SFLCSRPRG
     A            S2            10A  B  5 25
     A          R CTL01                       SFLCTL(SFL01)
     A                                        SFLPAG(5) SFLSIZ(20)
     A                                        SFLDSP    SFLDSPCTL
```

In this example, when the cursor leaves field S1, the cursor goes to S1 in the next subfile record.

SFLCSRRRN (Subfile Cursor Relative Record Number) keyword for display files

You use this record-level keyword on the subfile-control record format to return the relative record number of the record on which the cursor is located within a subfile.

If the subfile records occupy more than one line, use this keyword in conjunction with the SFLMODE keyword to determine the location of the cursor.

The format of the keyword is:

```
SFLCSRRRN(&relative-record);
```

The relative-record parameter is required. It specifies the name of a hidden field that, on input, will contain the relative record number of the subfile record on which the cursor is located. The field must be defined in the subfile-control record format as a signed numeric (S in position 35) field of length 5, with 0 decimal positions and usage H (hidden).

The relative-record field will contain the value 0 if the cursor is not located in the subfile associated with this subfile control record, or if the cursor is located within the subfile, but is not in an active record within the subfile. If the SFLMODE keyword is specified, the mode of the subfile will be returned in either case.

This keyword can be used on subfiles with field selection or subfiles with the SFLLIN keyword. If the cursor is located between two horizontal subfile records, the relative record number returned is 0.

For an example of how to specify the SFLCSRRRN keyword, see “SFLMODE (Subfile Mode) keyword for display files” on page 205.

SFLCTL (Subfile Control) keyword for display files

You use this record-level keyword to specify that this record format is to be a subfile-control record format. This record format must immediately follow the subfile record format.

The format of the keyword is:

```
SFLCTL(subfile-record-format-name)
```

You must specify the name of the subfile record format as the parameter value for this keyword. The subfile-control record format can contain field descriptions as well as subfile control keywords.

Your program can display subfile records only by issuing an output operation to the subfile-control record format.

The subfile record format (SFL keyword) defines the format of the records in the subfile as opposed to the subfile-control record format (SFLCTL keyword), which defines how the subfile can be displayed, cleared, deleted, and initialized. The program sends output operations to the subfile record format to build the subfile. It also sends output operations to the subfile-control record format, setting option indicators for various subfile keywords to display, clear, delete, and initialize the subfile.

The following tables are a summary of subfile keywords used with the SFLCTL keyword. (Field-level keywords are used with fields in the subfile-control record format.)

Table 12. Required keywords

Required keywords
SFLCTL
SFLDSP
SFLPAG
SFLSIZ

Table 13. Optional keywords

Optional keywords	**Optional keywords**
CHCAVAIL	SFLLIN
CHCSLT	SFLMCLTCHC
CHCUNAVAIL	SFLMSG
SFLCLR	SFLMSGID
SFLDLT	SFLPGMQ
SFLDROP	SFLRCDNBR
SFLDSPCTL[1]	SFLRNA
SFLEND	SFLROLVAL
SFLENTER	SFLSCROLL
SFLFOLD	SFLSNGCHC
SFLINZ	

[1]The SFLDSPCTL keyword is required if your program sends an input operation to the subfile-control record format.

If subfile size equals subfile page, the following keywords are ignored. When several display sizes are used (DSPSIZ keyword specified), these keywords are ignored only for display sizes for which subfile size equals subfile page:

SFLDROP
SFLFOLD
SFLROLVAL

If the subfile record format contains field selection, the following keywords are not valid on the subfile-control record format:

SFLDROP
SFLFOLD
SFLINZ
SFLLIN

SFLRCDNBR
SFLRNA (because SFLINZ is not valid)
SFLROLVAL

The USRDFN keyword is not valid for the subfile-control record format.

The keywords CHCAVAIL, CHCSLT, and CHCUNAVAIL can be used only if either SFLSNGCHC or SFLMLTCHC is also used.

Option indicators are not valid for this keyword.

Example

The following example shows how to specify the SFLCTL keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A          R SFLR                      SFL
     A*
     A*            (at least one displayable field)
     A*
00020A          R SFLCTLR                   SFLCTL(SFLR)
00030A                                      SFLPAG(17)
00040A                                      SFLSIZ(17)
00050A                                      SFLDSP
00060A                                      SFLDSPCTL
     A
```

Related reference

SFLMSGKEY (Subfile Message Key) keyword for display files
You use this field-level keyword on the first field in the subfile record format for a message subfile (with Subfile (SFL) and Subfile Message Record (SFLMSGRCD) keywords specified).

SFLMSGRCD (Subfile Message Record) keyword for display files
You use this record-level keyword on the subfile record format to specify that this subfile is to be a message subfile and that the records displayed when the subfile is displayed are messages from a program message queue.

SFLPGMQ (Subfile Program Message Queue) keyword for display files
You use this field-level keyword on the second (and last) field in the subfile record format for a message subfile.

SFLDLT (Subfile Delete) keyword for display files

You use this record-level keyword with an option indicator on the subfile-control record format to enable your program to delete the subfile.

When the maximum number of subfiles in a display file is already active (24) and another subfile is to be made active, your program must delete one of the active subfiles before making another active.

This keyword has no parameters.

To make a subfile active, your program sends an output operation to the subfile record format or sends an output operation to the subfile-control record format with the SFLINZ keyword in effect. To delete a subfile, your program sends an output operation to the subfile-control record format with SFLDLT in effect. (Closing the display file deletes all the active subfiles.)

If your program sends an output operation with SFLDLT in effect to a subfile that is not active, the SFLDLT keyword is ignored.

Option indicators are required for this keyword; display size condition names are not valid.

Example

The following example shows how to specify the SFLDLT keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A          R SFLR                      SFL
```

```
     A*
     A*            (at least one displayable field)
     A*
00040A          R SFLCTLR                   SFLCTL(SFLR)
00050A                                      SFLPAG(17)
00060A                                      SFLSIZ(17)
00070A  01                                  SFLDSP
00080A  01                                  SFLDSPCTL
00090A  04                                  SFLDLT
     A
```

The subfile is displayed when option indicator 01 is set on for an output operation to SFLCTLR, and the subfile is deleted when option indicator 04 is set on for an output operation to SFLCTLR. Normally, the option indicators specified for SFLDLT are different from the option indicators specified for the SFLDSP and SFLDSPCTL keywords.

SFLDROP (Subfile Drop) keyword for display files

You use this record-level keyword on the subfile-control record format to assign a command attention (CA) key or a command function (CF) key that the workstation user can press to fold or truncate subfile records that require more than one display line.

The format of the keyword is:

```
SFLDROP(CAnn | CFnn)
```

Without SFLDROP, the IBM i operating system displays the entire subfile record and folds it where needed. When SFLDROP is specified, the IBM i operating system first displays the subfile in truncated form; subfile records are truncated to fit on one display line. When the workstation user presses the specified key, the IBM i operating system displays the records again in folded form. Each record continues onto subsequent lines immediately following the line the record starts on.

By pressing the specified key, the form of the displayed subfile changes from one state to the other.

In the truncated form, more records are displayed than are specified on the SFLPAG keyword. In the folded form, as many records are displayed as are specified on the SFLPAG keyword.

The IBM i operating system truncates subfile records in the middle of output-only fields. However, if the truncation is in the middle of an input-capable field, the whole field is omitted from the display. If this results in omitting the entire record from the display, an error message is sent to the display and the record is not truncated. Instead, it is displayed in folded form.

Notes:

1. A warning message is sent at file creation if the entire record fits on a single display line.
2. If subfile size equals subfile page, SFLDROP is ignored. When several display sizes are used (DSPSIZ keyword specified), SFLDROP is ignored only for display sizes for which the subfile size equals subfile page. If the subfile record format contains field selection, SFLDROP is not valid.
3. If the subfile contains input-capable fields, it is recommended that you specify a CF key rather than a CA key. If you specify a CA key in this situation, changed data is lost when the key is pressed.
4. If several subfiles using SFLDROP are displayed at one time, the same function key should be specified on each SFLDROP keyword. If the function keys are different, only the key specified for the most recently displayed subfile is in effect. Pressing the function key affects the subfile containing the cursor. If the cursor is not positioned in a subfile, the function key affects the upper subfile.
5. SFLDROP can be specified on the same subfile-control record format as the SFLFOLD keyword. If both keywords are active, the SFLFOLD keyword is used. Indicators are checked at the time the subfile is displayed. Both keywords must use the same key.

Option indicators are valid for this keyword.

Example

The following example shows how to specify the SFLDROP keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A          R SFLR                      SFL
     A*
     A*           (subfile records should not fit on one screen line)
     A*
00040A          R SFLCTLR                   SFLCTL(SFLR)
00050A                                      SFLPAG(17)
00060A                                      SFLSIZ(34)
00070A                                      SFLDSP SFLDSPCTL
00090A                                      SFLDROP(CF03)
     A
```

When the subfile is displayed, the workstation user can press the CF03 key to change the subfile from truncated to folded form and from folded to truncated form.

SFLDSP (Subfile Display) keyword for display files

You use this record-level keyword on the subfile-control record format so that the IBM i operating system displays the subfile when your program sends an output operation to the subfile-control record format.

If you do not use an option indicator with this keyword, a page of subfile records is displayed on every output operation to the subfile-control record format.

This keyword has no parameters.

To determine which page of subfile records is displayed and when the subfile is displayed, see "SFLRCDNBR (Subfile Record Number) keyword for display files" on page 215.

If your program sends an output operation to the subfile-control record format when the SFLDSP keyword is in effect and the subfile is not activated (by adding records to it or by using SFLINZ), an error message is sent to your program.

This keyword is required and is valid only for the subfile-control record format. Display size condition names are not valid for this keyword.

Option indicators are valid for this keyword.

Example

The following example shows how to specify the SFLDSP keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A          R SFLR                      SFL
     A*
     A*              (at least one displayable field)
     A*
00040A          R SFLCTLR                   SFLCTL(SFLR)
00050A                                      SFLPAG(17)
00060A                                      SFLSIZ(17)
00070A  01                                  SFLDSP
00080A                                      SFLDSPCTL
     A
```

The subfile is displayed when option indicator 01 is set on for an output operation to SFLCTLR.

SFLDSPCTL (Subfile Display Control) keyword for display files

You use this record-level keyword on the subfile-control record format so that the IBM i operating system displays fields in the subfile-control record format when your program sends an output operation to the record format.

If you do not use an option indicator with this keyword, the subfile control record is displayed on every output operation to the subfile-control record format.

This keyword has no parameters.

This optional keyword is valid only for the subfile-control record format. Display size condition names are not valid for this keyword.

Option indicators are valid for this keyword.

Note: SFLDSPCTL must be in effect when the subfile is displayed for an input operation to the subfile control record to be valid, even if there are no displayable fields in the subfile-control record format.

Example

The following example shows how to specify the SFLDSPCTL keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A          R SFLR                      SFL
     A*
     A*            (at least one displayable field)
     A*
00040A          R SFLCTLR                   SFLCTL(SFLR)
00050A                                      SFLPAG(17)
00060A                                      SFLSIZ(17)
00070A  01                                  SFLDSP
00080A                                      SFLDSPCTL
     A                                    2 10'NAME'
     A                                    2 34'ADDRESS'
     A
```

Both the subfile and displayable fields in the subfile-control record format are displayed when option indicator 01 is set on for an output operation to SFLCTLR.

SFLEND (Subfile End) keyword for display files

You use this record-level keyword on the subfile-control record format to enable the display of a plus sign (+) or text (More or Bottom) in the lower-right display location occupied by the subfile or a scroll bar.

The plus sign or More text indicates that the workstation user can move the text lines on the subfile to display more records by pressing the Page Up key.

The format of the keyword is:

```
SFLEND[(*PLUS | *MORE | {*SCRBAR [*SCRBAR | *PLUS |*MORE ]})]
```

The scroll bar indicates different types of information about the subfile:

- Where the user is at in the subfile
- How big the subfile is
- What proportion of the subfile the user is viewing

The parameter values *PLUS, *SCRBAR, and *MORE are optional. If no parameter is specified, *PLUS is used. The second set of *PLUS, *MORE, and *SCRBAR can only be specified if *SCRBAR is specified as the first parameter. *SCRBAR is the default for the second parameter.

*PLUS tells the system to use the plus sign to indicate that you can use the Page Down key to see more records.

*MORE tells the system to use the More text to indicate that you can use the Page Down key to see more records. *MORE also tells the system to use the Bottom text to indicate that the last subfile record is displayed.

When *MORE is specified, the subfile takes up one more line on the screen (SFLPAG + 1). This line is needed for the text More and Bottom. If there is not room for the extra line on the display or in a window, a message is issued at file-creation time and the file is not created.

*SCRBAR tells the system to use a graphical scroll bar for a graphical display. When *SCRBAR is specified, the last 3 columns of the lines that the subfile is using is reserved for the scroll bar. When *SCRBAR is used, a second parameter can be specified. The second parameter tells the system what scrolling indicator should be used for nongraphical displays. *SCRBAR is the default for those displays. *MORE and

*PLUS can be used for the second parameter. When *SCRBAR is used the subfile must occupy at least 3 lines. SFLFOLD or SFLDROP will work with scroll bars. Both versions of the subfile (folded or truncated) must occupy three lines.

If the device configuration that is being used supports a pointer device, the scroll bar can also navigate through the subfile. For more information about how to support different device configurations and how scroll bars are controlled using the PAGEUP, PAGEDOWN, and SFLSIZ keywords, see the Application Display Programming book.

An option indicator must be specified for this keyword.

Paging through your program (SFLPAG equals SFLSIZ)

Your program controls the display of the plus sign or the More or Bottom text through the use of the indicators on SFLEND. Set the indicators off to display the plus sign or the More text. Set the indicators on to remove the plus sign from the display or to display the Bottom text. When the Page Up key is pressed, your program handles the processing. For instance, it reads the subfile, clears it, rewrites the subfile with new records, and displays it again. If your program does this, the display shows the plus sign or the More text. If not, the plus sign disappears from the display or the Bottom text appears.

Note: *SCRBAR can be used when SFLPAG equals SFLSIZ. The scroll bar will be displayed with buttons, a shaft, and a scroll box which covers the entire shaft.

Paging through IBM i (SFLPAG does not equal SFLSIZ)

The IBM i operating system displays the plus sign as long as there are more records in the subfile to be displayed, no matter how the option indicator is set. The scroll bar is displayed with the scroll box placed at a position on the scroll bar that best represents where the user is in the subfile. When the last page of the subfile is displayed, the operating system displays the plus sign, the More text, or the scroll bar with the scroll box one page size above the scroll button if the indicator is off. It does not display the plus sign, the Bottom text, or the scroll bar with the scroll box on top of the bottom scroll button if the indicator is on.

Your program must set the indicator on or off when displaying the subfile. (Your program cannot find out, when the IBM i operating system is paging through the subfile, which page of the subfile is displayed.)

If your program sets off the indicator for SFLEND when displaying the subfile, either the plus sign, the More text, or the scroll bar with the scroll box one page size above, the scroll button is displayed with the last page of the subfile. Because the plus sign is displayed but the IBM i operating system cannot page the subfile any further, your program must provide for any further paging. Specify the PAGEDOWN keyword on the subfile-control record format so that control is passed to your program when the Page Down key is pressed again. When your program receives control, it can add more records to the end of the subfile and use the SFLRCDNBR keyword to display a new page.

Note: If the PAGEDOWN keyword is specified with a scroll bar, then control is passed back to the program when a PAGEDOWN key is pressed or a manipulation of the graphical scroll bar will display a partial page.

Position of plus sign with *PLUS option

For the 24 x 80 display size, positions 78 through 80 of the last line occupied by the subfile are used for the beginning attribute character, plus sign, and ending attribute character. For the 27 x 132 display size, positions 130 through 132 of the last line occupied by the subfile are used for the beginning attribute character, plus sign, and ending attribute character.

Note: If an input field occupies the location of the plus sign and the field is changed, the plus sign and its attribute characters are returned to the program as data in the field. For selection lists, the plus will be positioned to the right of the choices for the list.

Position of More and Bottom text with *MORE option

For the 24 x 80 display size, positions 67 through 80 of the line immediately following the last line occupied by the subfile are used for the beginning attribute character, the right-aligned More or Bottom text, and the ending attribute character. For the 27 x 132 display size, positions 119 through 132 of the line immediately following the last line occupied by the subfile are used for the beginning attribute character, the right-aligned More or Bottom text, and the ending attribute character. For selection lists, the More and Bottom text is positioned to the right of the choices for the list.

Position of the scroll bar with *SCRBAR option

For the 24 x 80 display size, positions 77 through 80 of every line of the subfile will be reserved for the scroll bar. No fields of the subfile can use those columns. Thus no fields can occupy more than one line of the subfile. Multiple line subfiles can be used. For the 27 x 132 display size, positions 129 through 132 of every line of the subfile will be reserved for the scroll bar. For selection lists, the scroll bar will be positioned to the right of the choices for the list. For other subfiles, the scroll bar will be positioned on column 79.

Example 1

The following example shows how to specify the SFLEND keyword without parameters.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A          R SETSFL1                   SFL
00020A  50                                  SFLNXTCHG
00030A            SETSEL         1Y 0B  6  2VALUES(1 2 9) CHECK(AB)
00040A            SETNAME       10A  O  6  4
00050A          R SETCTL1                   SFLCTL(SETSFL1)
00060A                                      SFLSIZ(34)
00070A                                      SFLPAG(17)
00080A  40                                  SFLDSP
00090A  41                                  SFLDSPCTL
00100A  42                                  SFLDLT
00110A  43                                  SFLCLR
00120A  49                                  SFLEND
00130A N49                                  ROLLUP(26)
00140A            SETRRN         4S 0H      SFLRCDNBR(CURSOR)
     A
```

Paging is provided by the IBM i operating system and the plus sign (+) appears in the lower right corner of the display. When the last record is written to the subfile, indicator 49 is set on, which disables the Page Down key and omits the plus sign from the display.

Example 2

The following example shows how to specify the SFLEND keyword with *MORE as a parameter.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A          R SETSFL2                   SFL
00020A  50                                  SFLNXTCHG
00030A            SETSEL         1Y 0B  6  2VALUES(1 2 9) CHECK(AB)
00040A            SETNAM        10A  O  6  4
00050A          R SETCTL2                   SFLCTL(SETSFL2)
00060A                                      SFLSIZ(34)
00070A                                      SFLPAG(17)
00080A  40                                  SFLDSP
00090A  41                                  SFLDSPCTL
00100A  42                                  SFLDLT
00110A  43                                  SFLCLR
00120A  49                                  SFLEND(*MORE)
00130A N49                                  ROLLUP(26)
00140A            SETRRN         4S 0H      SFLRCDNBR(CURSOR)
     A
```

Paging is provided by the IBM i operating system. The More text appears at the lower right corner of the display on the line immediately following the subfile if there are more records to see in the subfile. When the last record is written to the subfile, indicator 49 is set on, which disables the Page Down key and causes the Bottom text to appear instead of the More text.

Example 3

The following example shows how to specify the SFLEND keyword with *SCRBAR as a parameter.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A          R SETSFL2                   SFL
00020A  50                                  SFLNXTCHG
00030A            SETSEL         1Y 0B  6  2VALUES(1 2 9) CHECK(AB)
00040A            SETNAM        10A  O  6  4
00050A          R SETCTL2                   SFLCTL(SETSFL2)
00060A                                      SFLSIZ(34)
00070A                                      SFLPAG(17)
00080A  40                                  SFLDSP
00090A  41                                  SFLDSPCTL
00100A  42                                  SFLDLT
00110A  43                                  SFLCLR
00120A  49                                  SFLEND(*SCRBAR *MORE)
00130A N49                                  ROLLUP(26)
00140A            SETRRN         4S 0H      SFLRCDNBR(CURSOR)
     A
```

Paging is provided by the IBM i operating system. The scroll bar is shown for graphical displays. If a graphical display is not used, then the More text appears at the lower right corner of the display on the line immediately following the subfile if there are more records to see in the subfile. When the last record is written to the subfile, indicator 49 is set on, which disables the Page Down key and causes the Bottom text to appear instead of the More text. The scroll bar is displayed with the scroll box just above the bottom scroll button.

Related reference

SFLROLVAL (Subfile Roll Value) keyword for display files
You use this field-level keyword in the subfile-control record format to specify that the workstation user can type a value in this field. The value tells the IBM i operating system how many records to page up or down when the appropriate paging key is pressed.

SFLENTER (Subfile Enter) keyword for display files

You use this record-level keyword on the subfile-control record format to specify that the Enter key is to be used as a Page Up key. This allows the IBM i operating system to page through a subfile of more than one page when the Enter key is pressed.

The format of the keyword is:

```
SFLENTER(CAnn | CFnn)
```

This optional keyword is valid only for the subfile-control record format.

If a subfile is not currently displayed or you cannot page through the subfiles displayed (they have one page or less) after you press the Enter key, control is returned to your program. The parameter value with this keyword is required. Use it to specify a function key to replace the Enter key while this function is active. This keyword is normally used when the subfile is entirely entered by the workstation user or when the user changes some records in the subfile and adds others.

Option indicators are not valid for this keyword.

Note: This keyword is in effect only until the next output operation. At the next output operation, the specifications for that record apply.

If more than one subfile using SFLENTER is displayed at the same time, the only CA or CF key in effect as an Enter key is the CA or CF key specified for SFLENTER on the most recently displayed subfile. The cursor position at the time the Enter key is pressed determines which subfile is affected.

Example

The following example shows how to specify the SFLENTER keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A          R SFLR                      SFL
```

```
00020A*
00030A*           (at least one displayable field)
00040A          R SFLCTLR                   SFLCTL(SFLR)
00050A                                      SFLPAG(17)
00060A                                      SFLSIZ(51)
00070A                                      SFLDSP SFLDSPCTL
00080A                                      SFLENTER(CF01)
     A
```

The Enter key is used as a Page Up key. To enter data, the workstation user presses CF01.

SFLFOLD (Subfile Fold) keyword for display files

You use this record-level keyword on the subfile-control record format to assign a command attention (CA) key or a command function (CF) key that the workstation user can press to truncate or fold subfile records that require more than one display line.

The format of the keyword is:

```
SFLFOLD(CAnn | CFnn)
```

When the SFLFOLD keyword is specified, the subfile is first displayed in folded form. When the workstation user presses the specified key, the IBM i operating system displays the records again in truncated form. By pressing the specified key, the form of the displayed subfile changes from one state to the other. When truncated, subfile records fit on one display line.

Without SFLFOLD, the IBM i operating system displays the entire subfile record folded where needed but the workstation user is not given the option to display the subfile record in truncated form.

In the folded form, as many records are displayed as are specified on the SFLPAG keyword. In the truncated form, more records are displayed than are specified on the SFLPAG keyword.

The IBM i operating system truncates subfile records in the middle of output-only fields. However, if the truncation is in the middle of an input-capable field, the whole field is omitted from the display. If this results in omitting the entire record from the display, an error message is sent to the display and the record is not truncated. Instead, it is displayed in folded form.

Notes:

1. A warning message (severity 10) is sent at file creation if the entire record fits on a single display line.
2. If subfile size equals subfile page, an error message (severity 20) is issued and SFLFOLD is ignored. When several display sizes are used (DSPSIZ keyword specified), SFLFOLD is ignored only for display sizes for which the subfile size equals subfile page. If the subfile record format contains field selection, SFLFOLD is not valid.
3. If the subfile contains input-capable fields, it is recommended that you specify a CF key rather than a CA key. If you specify a CA key in this situation, changed data is lost when the key is pressed.
4. If several subfiles using SFLFOLD are displayed at one time, the same function key should be specified on each SFLFOLD keyword. If the function keys are different, only the key specified for the most recently displayed subfile is in effect. Pressing the function key affects the subfile containing the cursor. If the cursor is not positioned in a subfile, the function key affects the upper subfile.
5. SFLFOLD can be specified on the same subfile-control record format as the SFLDROP keyword. If both keywords are active, SFLFOLD is used. Indicators are checked at the time the subfile is displayed. Both keywords must use the same key.

Option indicators are valid for this keyword.

Example

The following example shows how to specify the SFLFOLD keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A          R SFLR                      SFL
00020A*
00030A*           (subfile records should not fit on one screen line)
```

```
00040A          R SFLCTLR                   SFLCTL(SFLR)
00050A                                      SFLPAG(17)
00060A                                      SFLSIZ(34)
00070A                                      SFLDSP SFLDSPCTL
00080A                                      SFLFOLD(CF03)
     A
```

When the subfile is displayed, the workstation user can press the CF03 key to change the subfile from folded to truncated form and from truncated to folded form.

SFLINZ (Subfile Initialize) keyword for display files

You use this record-level keyword on the subfile-control record format to specify that the IBM i operating system is to initialize all records in the subfile on an output operation to the subfile-control record format identified by the SFLCTL (Subfile Control) keyword.

The fields in each subfile record are initialized to blanks for character type fields, to nulls for floating-point type fields, to zeros for other numeric type fields, or to the constant value specified on input-only fields if the DFT keyword is specified.

When the subfile is displayed (on an output operation to the subfile control record), all records in the subfile are displayed with the same value. Any record previously written is overwritten and no longer has its earlier value.

This keyword has no parameters.

The following case is true when SFLINZ is in effect on an output operation to the subfile-control record format. If keywords (such as DSPATR(HI)) are specified on fields in the subfile record format and if option indicators are specified on those keywords, the subfile is displayed as though all option indicators are off (hex F0). Note that a keyword can be selected if N is specified for the option indicator.

After your program sends an output operation to the subfile control record with SFLINZ in effect, all records in the subfile are considered active but not changed. They are considered changed only when the workstation user changes them or when your program sends an output operation to the subfile record format with the SFLNXTCHG keyword in effect.

To initialize a subfile with no active records, use the SFLRNA (Subfile Records Not Active) keyword.

In general, use SFLINZ for the following purposes:

- Specify SFLINZ with the SFLRNA keyword so that your program can initialize a subfile, then add records to that subfile without having the initialized records considered active.
- Specify SFLINZ with the SFLPGMQ keyword so that your program can build a message subfile with a single output operation.

Notes:

1. If field selection is used in the subfile record format, SFLINZ is not valid. Your program can only initialize the subfile by a series of output operations to the subfile record format, selecting fields as needed.
2. SFLINZ cannot be specified on the subfile-control record format for a message subfile (SFLMSGRCD) unless the SFLPGMQ keyword is also specified at the field level in the same subfile-control record format.

Option indicators are valid for this keyword. Display size condition names are not valid.

Example

The following example shows how to specify the SFLINZ keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A          R SFLR                      SFL
00020A*
00030A*           (at least one displayable field)
00040A          R SFLCTLR                   SFLCTL(SFLR)
00050A                                      SFLPAG(17)
00060A                                      SFLSIZ(17)
```

```
00070A  01                                 SFLDSP SFLDSPCTL
00080A  02                                 SFLINZ
00090A                                     UNLOCK(*ERASE *MDTOFF)
     A
```

Related reference

SFLRNA (Subfile Records Not Active) keyword for display files
You use this record-level keyword with the Subfile Initialize (SFLINZ) keyword on the subfile-control record format so that your program can initialize a subfile with no active records. To do this, your program sends an output operation to the subfile-control record format with the SFLINZ keyword selected.

SFLPGMQ (Subfile Program Message Queue) keyword for display files
You use this field-level keyword on the second (and last) field in the subfile record format for a message subfile.

SFLMSGRCD (Subfile Message Record) keyword for display files
You use this record-level keyword on the subfile record format to specify that this subfile is to be a message subfile and that the records displayed when the subfile is displayed are messages from a program message queue.

SFLLIN (Subfile Line) keyword for display files

You use this record-level keyword on the subfile-control record format to specify that the subfile is to be displayed as a horizontal subfile. The subfile has more than one column of records displayed.

The format of the keyword is:

```
SFLLIN(spaces)
```

The parameter value specifies the number of spaces (including attribute characters) between columns of records.

For example, specifying the SFLLIN keyword causes a subfile of four records to be displayed as:

```
REC1    REC3
REC2    REC4
```

If SFLLIN is not specified, these records appear as:

```
REC1
REC2
REC3
REC4
```

If the subfile record format contains field selection, this keyword is not valid.

To use SFLLIN for secondary display sizes, specify a SFLLIN keyword with a display size condition name for each secondary display size.

Because the SFLPAG keyword specifies the number of subfile records that can be displayed at a single time, you must consider SFLLIN when specifying the SFLPAG value.

SFLLIN is not valid for a message subfile.

Option indicators are not valid for this keyword.

Example

The following example shows how to specify the SFLLIN keyword. Columns of subfile records appear five spaces apart.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A          R SFLR                     SFL
     A*
     A*            (at least one displayable field)
     A*
00040A          R SFLCTLR                  SFLCTL(SFLR)
00050A                                     SFLPAG(17)
```

```
00060A                                      SFLSIZ(17)
00070A                                      SFLDSP SFLDSPCTL
00080A                                      SFLLIN(5)
     A
```

SFLMLTCHC (Subfile Multiple Choice Selection List) keyword for display files

You use this record-level keyword to define a subfile as a multiple-choice selection list. A *multiple-choice selection list* is a scrollable group of items from which the user can select multiple items.

The format of this keyword is:

```
SFLMLTCHC[(&number-selected] [*NORSTCSR | *RSTCSR]
[*NOSLTIND | *SLTIND])]
```

Parameters are optional and can be entered in any order.

The &number-selected parameter allows the application to find the number of items that were selected in the multiple-selection list. This parameter must name a hidden field with a length of 4, data type of Y, and zero decimal positions.

The *RSTCSR parameter specifies whether the arrow keys should be allowed to move the selection cursor outside of the selection list. *RSTCSR specifies that the arrow keys will not cause the selection cursor to move outside of the push button field. *NORSTCSR specifies that the arrow keys will cause the selection cursor to leave the field. If the SFLMLTCHC subfile control record is defined in a pulldown, the default is *RSTCSR. Otherwise, the default is *NORSTCSR.

The *SLTIND parameter specifies whether selection indicators are used when this selection list is displayed on a graphical display. *SLTIND specifies that the check boxes should be used on color graphical displays as selection indicator. *NOSLTIND specifies that no selection indicator should be used on a color graphical display and only a selection cursor can be used to make a selection. The default is *NOSLTIND.

A subfile containing the SFLMLTCHC keyword must:

- Contain only one output field
- Cannot contain input capable fields
- Might contain hidden fields

This optional keyword is valid only for the subfile-control record format.

The following subfile control record keywords cannot be specified on a record with the SFLMLTCHC keyword:

SFLDROP
SFLFOLD
SFLSNGCHC

The CHCAVAIL, CHCSLT, and CHCUNAVAIL keywords can be used to indicate the color of the items within the selection list, when the list is displayed on a color display station. The CHCAVAIL keyword indicates the color of the items within the list which are available for selection. The CHCSLT keyword indicates the color of the selected items. The CHCUNAVAIL keyword indicates the items on the list which are not available for selection. These keywords can be used in a subfile control record only if SFLSNGCHC or SFLMLTCHC keywords are also used.

Option indicators are not valid for this keyword.

Example 1

The following example shows how to specify the SFLMLTCHC keyword:

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A
```

```
     A          R SFLRCD                    SFL
     A            CTLFLD         1Y 0H      SFLCHCCTL
     A            F1            10A  O  6 10
     A          R SFLCTLRCD                 SFLCTL(SFLRCD)
     A                                      SFLMLTCHC
     A                                      SFLPAG(5) SFLSIZ(&SFLSIZ);
     A                                      SFLDSP SFLDSPCTL
     A                                      ROLLUP(10)
     A   10                                 SFLEND(*SCRBAR)
     A            F3             5S 0H      SFLSCROLL
     A            F2             4S 0H      SFLRCDNBR(CURSOR *TOP)
     A            SFLSIZ         5S 0P
     A                                  1 30'Panel Title'
     A                                  4  5'Select Multiple Items:'
```

In this example, when using a graphical display station attached to a controller that supports an enhanced interface for nonprogrammable workstations, a multiple-choice list looks like this:



Example 2

The following example shows how to specify what color the items on the list should have on a color display. Available items are displayed in red. Selected items are displayed in blue. Unavailable items are displayed in yellow. The CHCAVAIL, CHCSLT, and CHCUNAVAIL keywords can also be used to set the display attributes of the items on the list.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A
     A          R SFLRCD                    SFL
     A            CTLFLD         1Y 0H      SFLCHCCTL
     A            F1            10A  O  6 10
     A          R SFLCTLRCD                 SFLCTL(SFLRCD)
     A                                      SFLMLTCHC
     A                                      SFLPAG(5) SFLSIZ(&SFLSIZ);
     A                                      SFLDSP SFLDSPCTL
     A                                      ROLLUP(10)
     A                                      CHCAVAIL((*COLOR RED))
     A                                      CHCSLT((*COLOR BLU))
     A                                      CHCUNAVAIL((*COLOR YLW))
     A   10                                 SFLEND(*SCRBAR)
     A            F3             5S 0H      SFLSCROLL
     A            F2             4S 0H      SFLRCDNBR(CURSOR *TOP)
     A            SFLSIZ         5S 0P
     A                                  1 30'Panel Title'
     A                                  4  5'Select Multiple Items:'
```

Related reference

CHCAVAIL (Choice Color/Display Attribute when Available) keyword for display files
You use this field-level keyword to specify the color or display attributes to be used when the system is displaying the available choices in a menu bar, push button, selection field, or subfile single-choice or multiple-choice selection list.

CHCSLT (Choice Color/Display Attribute when Selected) keyword for display files
You use this field-level keyword to specify the color or display attributes to be used when the system is displaying a selected choice in a menu bar or selection field.

CHCUNAVAIL (Choice Color/Display Attribute when Unavailable) keyword for display files

You use this field-level keyword to specify the color or display attributes to be used when the system displays the unavailable choices in a selection field or a push button field.

SFLMODE (Subfile Mode) keyword for display files

You use this record-level keyword on the subfile-control record format to return an indication of whether the subfile was in folded or truncated mode on input. You use this keyword with the SFLCSRRRN (Subfile Cursor Relative Record Number) keyword to determine the location of the cursor within a subfile.

The format of the keyword is:

```
SFLMODE(&mode);
```

The mode parameter is required. It specifies the name of a hidden field that, on input, will contain the subfile mode. The field must be defined in the subfile-control record format as a character (A in position 35) field of length 1 with usage H (hidden). The field will contain the value 0 if the subfile is in folded mode; it will contain the value 1 if the subfile is in truncated mode.

If a SFLDROP or SFLFOLD keyword is not specified on the subfile control record, the mode value returned is 0.

Option indicators are not valid for this keyword.

Example

The following example shows how to specify the SFLMODE and SFLCSRRRN keywords.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A          R SFL01                     SFL
     A            FLD2A          2A  I  3  2
     A            FLD2B         30A  O  3  5
     A            FLD2C          6A  O  4 18
     A          R CTL01                     SFLCTL(SFL01)
     A                                      SFLSIZ(25)
     A                                      SFLPAG(4)
     A                                      SFLDSP
     A                                      SFLEND
     A                                      SFLCSRRRN(&RELRCD);
     A                                      SFLMODE(&MODE);
     A  10                                  SFLDROP(CF03)
     A  11                                  SFLFOLD(CF03)
     A                                      RTNCSRLOC(&CSRRCD &CSRFLD);
     A            RELRCD          5S 0H
     A            MODE            1A  H
     A            CSRFLD         10A  H
     A            CSRRCD         10A  H
     A
     A          R SFL02                     SFL
     A            FLD2A          2A  I 13  2
     A            FLD2B         30A  O 13  5
     A            FLD2C          6A  O 14 18
     A          R CTL02                     SFLCTL(SFL02)
     A                                      SFLSIZ(25)
     A                                      SFLPAG(4)
     A                                      SFLDSP
     A                                      SFLEND
     A                                      SFLCSRRRN(&RELRCD);
     A                                      SFLMODE(&MODE);
     A  12                                  SFLDROP(CF03)
     A  13                                  SFLFOLD(CF03)
     A                                      RTNCSRLOC(&CSRRCD &CSRFLD);
     A            RELRCD          5S 0H
     A            MODE            1A  H
     A            CSRFLD         10A  H
     A            CSRRCD         10A  H
     A
```

Indicators 10 and 13 are on. Two records were added to both subfiles. Both subfiles are displayed.

The following table shows the values returned for CTL02 when the cursor is at the specified positions.

Row	Column	RELRCD	MODE	CSRRCD	CSRFLD
13	2	1	0	SFL02	FLD2A
14	18	1	0	SFL02	FLD2C
15	2	2	0	SFL02	FLD2A
15	62	1	0	SFL02	blanks
17	2	0	0	SFL02	blanks
24	2	0	0	blanks	blanks
3	2	0	0	SFL01	FLD2A

The following table shows the values returned for CTL01 when the cursor is at the specified positions.

Row	Column	RELRCD	MODE	CSRRCD	CSRFLD
3	2	1	1	SFL01	FLD2A
4	18	2	1	SFL01	FLD2B
5	18	0	1	SFL01	blanks
13	2	0	1	SFL02	FLD2A

SFLMSG (Subfile Message) and SFLMSGID (Subfile Message Identifier) keywords for display files

You use these record-level keywords on the subfile-control record format to identify a message to be displayed on the message line when your program performs an output operation to the subfile-control record format.

Your program has the responsibility to reverse the images of any fields and to position the cursor appropriately in the subfile being displayed.

The formats of the keywords are:

```
SFLMSG('message-text' [response-indicator])
```

```
SFLMSGID(msgid [library-name/]msg-file [response-indicator] [&msg-data])
```

SFLMSG keyword

Specify SFLMSG as you do the ERRMSG keyword. The parameters specify a message text and, optionally, a response indicator. The message text is the message to be displayed.

If you specify a response indicator, it should be the same as the option indicator used to condition SFLMSG. On the input operation that follows the display of the error message, the IBM i operating system turns off the indicator. If the response and option indicators are the same, they are both turned off. One exception to this rule is that if the response indicator is also specified for another keyword, such as CHANGE, CA*nn*, or CF*nn*, the on/off setting of the response indicator is based on the results of the function provided by the CHANGE or CFnn keyword.

When a response indicator is specified, the first 50 characters of the message text are also used as indicator text. Separate response indicator text is not valid for SFLMSG.

SFLMSGID keyword

Specify SFLMSGID as you do the ERRMSGID keyword.

For SFLMSGID, the parameters specify:

- The message identifier for the message to be displayed
- The message file and, optionally, the library
- A response indicator
- A msg-data field name

The response indicator, if specified, should be the same as the option indicator used to condition the SFLMSGID keyword. On the subsequent input operation, after the display of the error message, the IBM i operating system turns off the indicator. However, if the response indicator is also specified on another keyword, such as CHANGE, CA*nn*, or CF*nn*, the on/off setting of the response indicator is based on the results of the function provided by the CHANGE, CA*nn*, or CF*nn* keyword.

Note: Indicator text cannot be specified on the SFLMSGID keyword.

The msg-data field, if specified, contains the replacement text for the specified message. The field must exist in the record format, and the field must be defined as a character field (data type A) with usage P. For more information about how replacement text works, see the Send Program Message (SNDPGMMSG) command.

Conditions occurring during message display

The display of messages using SFLMSG and SFLMSGID is similar to the display of messages by the IBM i operating system when field validation errors are detected. An important difference from ERRMSG and ERRMSGID is that the program, and not the IBM i operating system, must position the cursor to the appropriate field within the subfile, reverse the image of that field within the subfile, and also optionally reverse the image of more than one field at a time. On the 5250 workstation, blinking cursor and message highlighting are allowed.

Note: The SFLDSP keyword must be in effect for SFLMSG and SFLMSGID to be processed.

Restoration of reversed image fields

See the Restoration of reversed image fields section in “ERRMSG (Error Message) and ERRMSGID (Error Message Identifier) keywords for display files” on page 112.

Priority among selected keywords

You can specify either SFLMSG or SFLMSGID several times for a single subfile-control record format. In your program, set option indicators to select a particular message to be displayed and to select particular fields to be displayed in reverse image. Several fields can be displayed in reverse image in different records of a subfile when the subfile is displayed again. However, only one message can be displayed at one time.

If more than one error message is selected at a time, the IBM i operating system displays the first of the following keywords:

1. ERRMSG (If more than one ERRMSG keyword is selected, the first one selected is displayed.)
2. ERRMSGID (If more than one ERRMSGID keyword is selected, the first one selected is displayed.)
3. SFLMSG (If more than one SFLMSG keyword is selected, the first one selected is displayed.)
4. SFLMSGID (If more than one SFLMSGID keyword is selected, the first one selected is displayed.)
5. Message fields (M in position 38) (If more than one message field is selected, the first one selected is displayed.)

Multiple subfile messages (SFLMSG and SFLMSGID) are allowed on error subfiles.

Option indicators are valid for these keywords.

Example

The following example shows how to specify the SFLMSG and SFLMSGID keywords.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A          R SFLR                      SFL
00020A*
00030A*           (at least one displayable field)
00040A*
00050A          R SFLCTLR                   SFLCTL(SFLR)
00060A                                      SFLPAG(17)
00070A                                      SFLSIZ(17)
00080A                                      SFLDSP SFLDSPCTL
00090A  11                                  SFLMSGID(USR0006 PAYROLL/UMSGF1 +
00100A                                      11 &RPLTXT);
00110A  12                                  SFLMSGID(USR0007 PAYROLL/UMSGEF1 +
00120A                                      12 &RPLTXT);
00130A            RPLTXT        78A  P
     A
```

SFLMSGKEY (Subfile Message Key) keyword for display files

You use this field-level keyword on the first field in the subfile record format for a message subfile (with Subfile (SFL) and Subfile Message Record (SFLMSGRCD) keywords specified).

The SFLMSGKEY keyword is not valid on the subfile-control record format (SFLCTL). To select messages from a program message queue for display, your program places a message reference key in this field. Your program also places the name of the program message queue in the second field in the subfile record format.

This keyword has no parameters.

This field is predefined as a four-position, character data type, hidden field. The following rules apply:

- This field must always be the first field defined in the subfile record format.
- The field name and SFLMSGKEY are the only DDS you can specify for this field.

Option indicators are not valid for this keyword or with the associated field.

For more information about building and displaying message subfiles, see SFLPGMQ (Subfile Program Message Queue) keyword for display files.

Example

The following example shows how to specify the SFLMSGKEY keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A          R RCDMSG                    SFL SFLMSGRCD(3)
00020A            FLDKEY                    SFLMSGKEY
00030A            FLDPGM                    SFLPGMQ
00040A          R SFLCTL                    SFLCTL(RCDMSG)
00050A  01                                  SFLINZ
00060A                                      SFLPAG(17)
00070A                                      SFLSIZ(17)
00080A                                      SFLDSP SFLDSPCTL
00090A            FLDPGM                    SFLPGMQ
     A
```

Related reference

SFLCTL (Subfile Control) keyword for display files
You use this record-level keyword to specify that this record format is to be a subfile-control record format. This record format must immediately follow the subfile record format.

SFLPGMQ (Subfile Program Message Queue) keyword for display files

You use this field-level keyword on the second (and last) field in the subfile record format for a message subfile.

SFLMSGRCD (Subfile Message Record) keyword for display files

You use this record-level keyword on the subfile record format to specify that this subfile is to be a message subfile and that the records displayed when the subfile is displayed are messages from a program message queue.

The format of the keyword is:

```
SFLMSGRCD(line-number)
```

The parameter value specified with SFLMSGRCD specifies the first line on the display on which messages are displayed. The value specified must not be greater than the maximum line number for the display size being used. The number of messages displayed depends on the SFLPAG value specified for the subfile.

For more information about building and displaying message subfiles, see "SFLPGMQ (Subfile Program Message Queue) keyword for display files" on page 213.

The TEXT keyword is valid at the record-level for SFLMSGRCD.

There can be only two predefined fields specified on the subfile record format for a message subfile as follows:

- Message identifier. A 4-position, character data type, hidden field. Your program uses this field to pass a message identifier to the IBM i operating system. This field must always be the first field defined in the message subfile. You must specify the SFLMSGKEY keyword with this field.
- Program queue name. A 10-position, character data type, hidden field. Your program passes the name of the program message queue that contains the messages in this field. It must be the second field of a subfile message record and must immediately follow the first field. If specified also on the subfile control record, it can be anywhere within the record specification. You must specify SFLPGMQ with this field.

Display size condition names can be specified for SFLMSGRCD and are required if the line number for the first message displayed is to change, based on display size.

Data is not returned in your input buffer if your program does an input operation for a message subfile.

The messages are displayed as follows:

- Each message is displayed on a separate line and is truncated if it is longer than the display line length.
- Each message starts in position 2. The maximum message length for the 24 x 80 display size is 76 characters. The maximum message length for the 27 x 132 display size is 128 characters.
- Each message is displayed with the high intensity (HI) field attribute.

When a message subfile is paged through the IBM i operating system, the cursor is positioned at the same location as it was when the Page key was pressed.

Message help is supported for these messages. The workstation user chooses which message help is to be displayed by placing the cursor on the line containing the message and pressing the Help key.

Notes:

1. When SFLMSGRCD is specified, you cannot specify the SFLINZ keyword without specifying SFLPGMQ.
2. The message subfile, starting on the line specified on SFLMSGRCD, must not overlap any displayable fields in the subfile control record, even if option indicators are specified on the displayable fields.

Option indicators are not valid for this keyword; display size condition names are valid.

Example

The following example shows how to specify the SFLMSGRCD keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00030A          R RCDMSG                    SFL SFLMSGRCD(3)
00040A            FLDKEY                    SFLMSGKEY
00050A            FLDPGM                    SFLPGMQ
00060A          R SFLCTL                    SFLCTL(RCDMSG)
00070A                                      SFLPAG(17)
00080A                                      SFLSIZ(17)
00090A                                      SFLDSP SFLDSPCTL
     A
```

The highlighted keywords are required on the subfile record format for a message subfile. The SFLMSGKEY and SFLPGMQ keywords must be specified in the order shown.

Related reference

SFL (Subfile) keyword for display files
You use this record-level keyword to specify that this record format is to be a subfile record format.

SFLCTL (Subfile Control) keyword for display files
You use this record-level keyword to specify that this record format is to be a subfile-control record format. This record format must immediately follow the subfile record format.

SFLINZ (Subfile Initialize) keyword for display files
You use this record-level keyword on the subfile-control record format to specify that the IBM i operating system is to initialize all records in the subfile on an output operation to the subfile-control record format identified by the SFLCTL (Subfile Control) keyword.

SFLNXTCHG (Subfile Next Changed) keyword for display files

You use this record-level keyword on the subfile record format to force the workstation user to correct program-detected typing errors in subfile records that have been read by the program.

The program does this by causing a record to be changed so that a get-next-changed operation must read the record as described in the following section.

This keyword has no parameters.

Subfile operations with SFLNXTCHG

A typical use of SFLNXTCHG can be as follows:

A workstation user changes some records in a displayed subfile (this can be for a data-entry application or a data-update application). After changing some records, the workstation user presses the Enter key, and the program reads only the changed records with get-next-changed operations. (For example, READC in RPG III and READ-SUBFILE-NEXT-MODIFIED in COBOL.)

If the program detects typing errors in the changed records, it can send update operations (UPDATE in RPG IV, REWRITE SUBFILE in COBOL) to the subfile records in error, setting indicators so that SFLNXTCHG is in effect during the update operations. These update operations are sent to the subfile record format.

After all the records in error have been updated, the program sends an input/output operation to the subfile-control record format to display the subfile again.

With the subfile displayed again, the workstation user types the data again and presses the Enter key. If the data is correct, the program does not display the subfile again.

The records in error (and any other records changed by the workstation user) are returned to the program on the next get-next-changed operation. This is because SFLNXTCHG caused the subfile records to be considered changed even though the workstation user did not change them. This allows the program to prohibit the workstation user from ignoring program-detected typing errors in subfile records.

Subfile operations without SFLNXTCHG

If SFLNXTCHG is not specified, or is specified but not selected on the update operations to the subfile records, then the workstation user can press the Enter key instead of correcting the program-detected errors. The program then reads no records because the get-next-changed operations find no changed records the second time the Enter key is pressed.

Option indicators are valid for this keyword.

You cannot specify SFLNXTCHG with the SFLMSGRCD keyword.

Example

The following example shows how to specify the SFLNXTCHG keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A          R SFLR                      SFL
00020A  14                                  SFLNXTCHG
     A*
     A*            (at least one input-capable field should be specified)
     A*
00040A          R SFLCTLR                   SFLCTL(SFLR)
00050A                                      SFLPAG(17)
00060A                                      SFLSIZ(17)
00070A                                      SFLDSP SFLDSPCTL
     A
```

SFLPAG (Subfile Page) keyword for display files

You use this record-level keyword on the subfile-control record format to specify the number of records in the subfile to be displayed at the same time.

For an exception to this rule, see Field selection.

The format of the keyword is:

```
SFLPAG(number-of-records-to-be-displayed)
```

The SFLPAG parameter value and the number of lines required by each subfile record determine the number of actual lines required to display the page of records. Not all records within a subfile must be displayed at the same time, and not all lines of the display are required to display a page of subfile records.

This keyword is required for the subfile-control record format.

Subfile page equals subfile size

When you specify the same parameter values for SFLPAG and SFLSIZ, the maximum number of records that can be contained in the subfile equals the maximum number of subfile records that can appear on the display at one time. For this condition, the IBM i program does not automatically page through the subfile when the Page Up or the Page Down key is pressed. If the ROLLUP and ROLLDOWN keywords are specified and one of the Page keys is pressed, the IBM i operating system returns control to your program instead. If ROLLUP and ROLLDOWN are not specified, a message is sent to the workstation user, indicating that a key is not supported on the display.

If subfile size equals subfile page, the following keywords are not allowed:

SFLDROP
SFLFOLD
SFLROLVAL

When several display sizes are used (DSPSIZ keyword specified), these keywords are ignored only for display sizes for which subfile size equals subfile page.

Field selection

When subfile page equals subfile size, you can specify option indicators for fields in the subfile record format. This is called *field selection*. When field selection is used in the subfile record, SFLPAG(value) specifies the number of display lines available to display the records of this subfile. (Without field selection, SFLPAG(value) specifies the number of subfile records that can be displayed at one time.) This specification must be considered when a subfile record occupies more than one display line. The value of SFLPAG must be greater than or equal to the number of display lines occupied by the subfile.

If the subfile record format contains field selection, the following keywords are not valid on the subfile-control record format:

SFLDROP
SFLFOLD
SFLINZ
SFLLIN
SFLRCDNBR
SFLRNA (because SFLINZ is not valid)
SFLROLVAL

Subfile page does not equal subfile size

When you specify different parameter values for SFLPAG and SFLSIZ, the IBM i operating system recognizes the Page Up and Page Down keys and automatically pages through the subfile according to the value specified in the field for which the SFLROLVAL keyword is specified. If you do not specify the SFLROLVAL keyword, the IBM i program pages through the subfile by the parameter value specified for the SFLPAG keyword except for subfiles using SFLDROP. If you use the SFLDROP keyword, more records are displayed than the SFLPAG value when records are displayed in the truncated format. For truncated records, the IBM i operating system pages through the display by the number of records displayed in the truncated format.

Option indicators are not valid for this keyword. Display size condition names are valid. Display size condition names are required if you want the number of records that can be displayed at one time to change, based on the size of the display.

Example

The following example shows how to specify the SFLPAG keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A          R SFLR                      SFL
     A*
     A*            (at least one displayable field)
     A*
00040A          R SFLCTLR                   SFLCTL(SFLR)
00050A                                      SFLPAG(17)
00060A                                      SFLSIZ(17)
00070A                                      SFLDSP SFLDSPCTL
     A
```

Because the value specified for the SFLPAG keyword equals the value specified for SFLSIZ(17), subfile page equals subfile size.

Related reference

SFL (Subfile) keyword for display files

You use this record-level keyword to specify that this record format is to be a subfile record format.

SFLPGMQ (Subfile Program Message Queue) keyword for display files

You use this field-level keyword on the second (and last) field in the subfile record format for a message subfile.

This field contains the name of the program message queue used by the IBM i operating system to build a message subfile. In addition, SFLPGMQ can be specified on the subfile-control record format when the SFLINZ keyword is specified on the subfile-control record format.

The format of the keyword is:

```
SFLPGMQ([10] | [276])
```

When 10 is specified, SFLPGMQ generates a 10-byte field. 10 is the default.

When 276 is specified, SFLPGMQ generates a 276-byte field.

This field is predefined as a character data type, hidden field. The following rules apply:

- The field name and the SFLPGMQ keyword and parameters are the only DDS you can specify for this field.
- If the name of the program message queue placed in this field at processing time is less than the field length (10 or 276 bytes), it must be left-aligned and padded with blanks.

For Integrated Language Environment® (ILE) programs using the 276-byte parameter value, the format of the field data must be as follows:

- The first 256 bytes contains the ILE call message queue name. The call message queue name is the same as the ILE procedure name. The name must be left-aligned and padded with blanks.
- Bytes 257 through 266 will optionally contain the ILE module name. The name, when specified, must be left-aligned and padded with blanks. If no module name is provided, these bytes must be set to blanks.
- Bytes 267 through 276 will optionally contain the name of the ILE bound program name. The name, when specified, must be left-aligned and padded with blanks. If no bound program name is provided, these bytes must be set to blanks.

  Notes:

  1. If a parameter value of 10 is used on SFLPGMQ and an ILE procedure name longer than 10 bytes is placed into this field at processing time, the procedure name is truncated to 10 bytes. The results will be unpredictable.
  2. If a parameter value of 276 is used on SFLPGMQ and a program message queue name is placed into this field at processing time, bytes 257 through 276 must be set to blanks. If these bytes are not blank, the system assumes that a call message queue name has been given and will not find the specified program message queue.
- If the SFLPGMQ keyword is specified on both the subfile and subfile control record, the SFLPGMQ parameter values must match. However, different subfiles within the same file can use different SFLPGMQ parameter values.

This field is required on the subfile record format (identified by the SFL keyword) to build the subfile one message at a time through multiple output operations to the subfile record format.

You can also specify this field on the subfile-control record format (identified by the SFLCTL keyword) to build the subfile all at once through a single output operation to the subfile control record. Specify option indicators with the SFLINZ keyword to control the way the subfile is built.

Multiple output operation

If you specify the field name and SFLPGMQ on the subfile record, you build the subfile one message at a time with separate output operations to the subfile record format. For each output operation, the message reference key must be in the first field of the record (SFLMSGKEY keyword), and the name of

the program message queue must be in the second field. At the time of the output operation, the IBM i program retrieves the identified message from the queue and places it in the subfile as a record.

Note: A CL program cannot be used for a multiple output operation. The relative record number required each time a message is built is not supported for CL.

Single output operation

If you specify SFLPGMQ (with its named field) and the SFLINZ keyword on the subfile-control record format, you build the entire subfile with one output operation directed to the subfile-control record format. On the output operation, the IBM i operating system initializes the subfile with all messages that are on the program message queue whose name is in the SFLPGMQ field. If necessary, the IBM i operating system extends the subfile to contain all messages on the queue. For this function, the SFLMSGRCD, SFLMSGKEY, and SFLPGMQ keywords must be specified with the subfile record format (SFL keyword). The SFLPGMQ and SFLMSGKEY keywords are ignored for this function and your program need not set the values of their fields.

Special value

The SFLPGMQ field can contain a special value, * (asterisk), instead of a program message queue name. If the program moves an asterisk to the SFLPGMQ field, the IBM i operating system uses the message queue of the program issuing the output operation. You cannot use an asterisk if your program is a CL program.

Both multiple and single output operations

If you specify SFLPGMQ with both the subfile record format and subfile-control record format, you can use the single operation function one time and the multiple operation function some other time. Do this by setting indicators before issuing the output operation; however, all operations to a particular subfile must be consistent (multiple or single, but not intermixed) when preparing for a single display of the subfile.

Option indicators and display size condition names are not valid for this keyword.

Example

The following example shows how to specify the SFLPGMQ keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A          R RCDMSGILE                 SFL SFLMSGRCD(3)
00020A            FLDKEY                    SFLMSGKEY
00030A            FLDPGM                    SFLPGMQ(276)
00040A          R SFLCTLILE                 SFLCTL(RCDMSG)
00050A  01                                  SFLINZ
00060A                                      SFLPAG(17)
00070A                                      SFLSIZ(17)
00080A                                      SFLDSP SFLDSPCTL
     A            :
     A            :
00110A            FLDPGM                    SFLPGMQ(276)
     A          R RCDMSGOPM                 SFL SFLMSGRCD(3)
     A            FLDKEY                    SFLMSGKEY
     A            FLDPGM                    SFLPGMQ
     A          R SFLCTLOPM                 SFLCTL(RCDMSG)
     A  02                                  SFLINZ
     A                                      SFLPAG(17)
     A                                      SFLSIZ(17)
     A                                      SFLDSP SFLDSPCTL
     A              :
     A              :
     A            FLDPGM                    SFLPGMQ(10)
     A
```

In this example, the program can build the subfile with more than one output operation (indicator 01 off) or a single output operation (indicator 01 on) to the subfile-control record format.

In the first record, the name of the subfile program queue can be as long as 276 bytes, while the name of the subfile program queue in the third record format can only be 10 bytes long.

Related reference

SFLCTL (Subfile Control) keyword for display files
You use this record-level keyword to specify that this record format is to be a subfile-control record format. This record format must immediately follow the subfile record format.

SFLINZ (Subfile Initialize) keyword for display files
You use this record-level keyword on the subfile-control record format to specify that the IBM i operating system is to initialize all records in the subfile on an output operation to the subfile-control record format identified by the SFLCTL (Subfile Control) keyword.

SFLMSGKEY (Subfile Message Key) keyword for display files
You use this field-level keyword on the first field in the subfile record format for a message subfile (with Subfile (SFL) and Subfile Message Record (SFLMSGRCD) keywords specified).

SFLRCDNBR (Subfile Record Number) keyword for display files

You use this field-level keyword on the subfile-control record format to specify that the page of the subfile to be displayed is the page that contains the record whose relative record number is in this field.

If you do not specify this keyword, the IBM i operating system displays the first page of the subfile by default.

The format of the keyword is:

```
SFLRCDNBR[([CURSOR] [*TOP])]
```

If CURSOR is specified, the cursor is placed in the subfile record whose relative record number is identified by the contents of this field. The cursor is positioned at the first input-capable field in the subfile record. If there is no input-capable field, the cursor is positioned at the first output-only or constant field. For example, if a page can contain three records, and nine records are contained in the subfile, a SFLRCDNBR field value of 8 causes records 7, 8, and 9 to be displayed. If CURSOR is specified, the cursor appears in record 8.

If *TOP is specified, the subfile record whose relative record number is identified by the contents of this field will display as the first record of the page of the subfile records being displayed.

This field must be a zoned decimal field with zero decimal positions. It must have the keyboard shift attribute of signed numeric (S in position 35), and it can be up to 4 digits in length. It must be defined as an output-only, an input/output, or a hidden field. If a value less than 1 or a value greater than the number of records in the subfile is contained in this field on an output operation to the subfile-control record format, an error is returned to your program.

This optional keyword is valid only for the subfile-control record format.

You cannot specify both SFLRCDNBR and SFLROLVAL for the same field.

If the subfile record format contains field selection, this keyword is not allowed.

Option indicators are not valid for this keyword.

Example

The following example shows how to specify the SFLRCDNBR keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A          R SFLR                      SFL
     A*
     A*            (at least one displayable field)
     A*
00040A          R SFLCTLR                   SFLCTL(SFLR)
00050A                                      SFLPAG(17)
00060A                                      SFLSIZ(17)
00070A                                      SFLDSP SFLDSPCTL
```

```
00080A            DSPREC         4S 0B  4 12SFLRCDNBR(CURSOR)
     A
```

In this example, either the program or the workstation user can set the value of the field before displaying the subfile.

Related reference

ALWROL (Allow Roll) keyword for display files
This record-level keyword enables your program to page through data in a window on the display when the system is displaying the record format you are defining.

SFLRNA (Subfile Records Not Active) keyword for display files

You use this record-level keyword with the Subfile Initialize (SFLINZ) keyword on the subfile-control record format so that your program can initialize a subfile with no active records. To do this, your program sends an output operation to the subfile-control record format with the SFLINZ keyword selected.

The subfile itself becomes active. Subfile records are not considered active unless one of the following conditions occurs:

- Your program sends an output operation to the subfile record format, placing data in one of the subfile records. The subfile record becomes active but is not considered changed unless the SFLNXTCHG keyword is also in effect.
- After your program displays the subfile, the workstation user types data into subfile records. The records typed in become active and changed.

This keyword has no parameters.

SFLRNA is normally used so that a program can write some records to the subfile before displaying it, and the workstation user can then add records to the subfile.

When your program displays a subfile initialized with the SFLINZ and SFLRNA keywords in effect, fields in inactive records have the following values:

- Character fields are blank.
- Numeric fields are zero-filled.
- An input-only field with a constant value specified has the constant value.

Your program cannot send an input operation to an inactive subfile record. Issuing a get-next-changed operation to one of the subfile records returns the record only when the record has become active and has been changed.

Your program cannot send output operations to active records (SFLRNA not specified). It must send update operations. Also, your program cannot send update operations to inactive records (SFLRNA specified). It must send output operations.

The SFLINZ keyword is required when SFLRNA is specified.

SFLRNA cannot be specified for a message subfile (identified by the SFLMSGRCD keyword on the subfile record format).

If the subfile record format contains field selection, SFLRNA is not valid.

Option indicators are not valid for this keyword.

Example

The following example shows how to specify the SFLRNA keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A          R SFLR                      SFL
     A*
     A*            (at least one displayable field)
     A*
00040A          R SFLCTLR                   SFLCTL(SFLR)
00050A                                      SFLPAG(17)
```

```
00060A                                         SFLSIZ(17)
00070A                                         SFLDSP SFLDSPCTL
00080A                                         SFLINZ
00090A                                         SFLRNA
     A
```

Related reference

SFLINZ (Subfile Initialize) keyword for display files

You use this record-level keyword on the subfile-control record format to specify that the IBM i operating system is to initialize all records in the subfile on an output operation to the subfile-control record format identified by the SFLCTL (Subfile Control) keyword.

SFLROLVAL (Subfile Roll Value) keyword for display files

You use this field-level keyword in the subfile-control record format to specify that the workstation user can type a value in this field. The value tells the IBM i operating system how many records to page up or down when the appropriate paging key is pressed.

This field must have the keyboard shift attribute of signed numeric with zero decimal positions. It can be up to 4 digits in length, and it must be defined as an input/output or input-only field.

This keyword has no parameters.

The workstation user can page through the data being displayed up or down by first typing in the number of records to page through and then pressing the Page Up or the Page Down key. (On subsequent pages, the SFLROLVAL value stays the same unless a new number is typed in before paging.) If a negative number or zero is typed into this field and a Page key is pressed, an error message is displayed at the workstation.

This keyword is valid only for the subfile-control record format. You must specify it if the IBM i operating system is to support the page-by-record function.

If this keyword is not specified, the IBM i operating system pages through the display by the SFLPAG value except for subfiles using the SFLDROP keyword. If using the SFLDROP keyword, more records are displayed than the SFLPAG value when records are displayed in the truncated format. For truncated records, the IBM i operating system pages through the display by the number of records displayed in the truncated format.

If subfile size equals subfile page, SFLROLVAL is ignored. When several display sizes are used (DSPSIZ keyword specified), SFLROLVAL is ignored only for display sizes for which subfile size equals subfile page. If the subfile record format contains field selection, SFLROLVAL is not valid.

This field is returned to your program as part of the input for this subfile control record.

If pressing the Page Up key pages beyond the first page of records of the subfile, one of the following conditions occurs:

- If the first page of records is not currently displayed, paging up will display it.
- If the first page of records is currently displayed, paging up will cause a message to be displayed.

If pressing the Page Down key pages beyond the last active record of the subfile, one of the following conditions occurs:

- If the last full page of records is not already displayed, typing Page Down will display it.
- If the last full page of records is already displayed, typing Page Down will cause a message to be displayed. One exception to this rule is that when the SFLROLVAL value is less than the SFLPAG value, the IBM i operating system pages through the subfile again and no message is displayed.

Certain keywords are helpful when specified with SFLROLVAL:

- The SFLEND keyword notifies the workstation user when the last subfile record is displayed.
- The PAGEUP or PAGEDOWN keywords cause control to return to the program when pressing a Page Up or a Page Down key can page beyond the end of the subfile. Without PAGEUP(ROLLDOWN) or PAGEDOWN(ROLLUP), a message is displayed (as described previously).

Note: The ROLLUP keyword is the same as the PAGEDOWN keyword and the ROLLDOWN keyword is the same as the PAGEUP keyword.

The following examples illustrate the use of SFLROLVAL:

- Paging Up. Assume that the value specified for SFLPAG is 3, and there are 11 active records in the subfile. If records 8 through 10 are currently being displayed, and the user types a page value greater than 7, pressing the Page Up key displays records 1 through 3.

  If records 1 through 3 are currently displayed, and a Page Up key is entered with a SFLROLVAL value greater than 0, either a message is sent to the workstation user (PAGEUP not specified) or control is returned to the user program (PAGEUP specified; the program has responsibility for paging down).
- Paging Down. Assume that the value specified for SFLPAG is 3, and there are 11 active records in the subfile. If records 8 through 10 are currently being displayed, and the user enters a 3 into the SFLROLVAL field, pressing the Page Down key displays record 11 in the uppermost page area of the display. Any lines not occupied by that record are blank. If the Page Down key is pressed again, the last full page of subfile records (records 9 through 11) are displayed. Finally, if the Page Down key is pressed a third time, either a message is sent to the workstation user (PAGEDOWN not specified) or control is returned to the user program (PAGEDOWN specified; the program has responsibility for paging down).

The following shows the conditions that occur when paging beyond the ends of the subfile (when the SFLROLVAL value is greater than the SFLPAG value).

- On pressing the Page Down key:
  - If the last full page of records is not already displayed, then it is displayed.
  - If the last full page of records is already displayed, then a message is displayed.
- On pressing the Page Down key a second time:
  - If PAGEDOWN is specified, control returns to your program.
  - If PAGEDOWN is not specified, a message is displayed.
- On pressing the Page Up key:
  - If the first page of the subfile is not already displayed, then it is displayed.
  - If the first page of the subfile is already displayed, then:
    - If PAGEUP is specified, control returns to your program.
    - If PAGEUP is not specified, a message is displayed.

You cannot specify both the SFLROLVAL and the SFLRCDNBR keywords for the same field.

Option indicators are not valid for this keyword.

Example

The following example shows how to specify the SFLROLVAL keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A          R SFLR                      SFL
     A*
     A*            (at least one displayable field)
     A*
00040A          R SFLCTLR                   SFLCTL(SFLR)
00050A                                      SFLPAG(17)
00060A                                      SFLSIZ(17)
00070A                                      SFLDSP SFLDSPCTL
00080A            ROLVAL         4S 0B  1 47SFLROLVAL
     A
```

Related reference

ALWROL (Allow Roll) keyword for display files
This record-level keyword enables your program to page through data in a window on the display when the system is displaying the record format you are defining.

SFLEND (Subfile End) keyword for display files

You use this record-level keyword on the subfile-control record format to enable the display of a plus sign (+) or text (More or Bottom) in the lower-right display location occupied by the subfile or a scroll bar.

SFLRTNSEL (Subfile Return Selected Choices) keyword for display files

You use this record-level keyword on a selection-list subfile control record to control how choices are returned to the application with the GET-NEXT-CHANGED operation.

This keyword has no parameters.

If this keyword is specified then SFLMLTCHC or SFLSNGCHC must be specified.

Specifying this keyword causes the GET-NEXT-CHANGED operation to return all selected choices. This includes default choices, which did not actually change. For example, the user never specifically selected the choice.

If this keyword is not specified, only the changed records will be returned to the application by the GET-NEXT-CHANGED operation. This means that a default selection will not be returned because that choice was not changed by the user.

Option indicators are not valid for this keyword.

Example

The following example shows how to specify the SFLRTNSEL keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A
     A          R SFLRCD                    SFL
     A            CTLFLD         1Y 0H      SFLCHCCTL
     A            F1             4A  O  6 10
     A          R SFLCTLRCD                 SFLCTL(SFLRCD)
     A                                      SFLMLTCHC
     A                                      SFLRTNSEL
     A                                      SFLPAG(5) SFLSIZ(&SFLSIZ);
     A                                      SFLDSP SFLDSPCTL
     A                                      ROLLUP(10)
     A   10                                 SFLEND(*SCRBAR)
     A            F3             5S 0H      SFLSCROLL
     A            F2             4S 0H      SFLRCDNBR(CURSOR *TOP)
     A            SFLSIZ         5S 0P
     A                                 1 30'Panel Title'
     A                                 4  5'Single selection list:'
     A
```

In this example, the GET-NEXT-CHANGED operation returns all selected choices. This includes default choices, which did not actually change, for example, when the user never specifically selected the choice.

SFLSCROLL (Subfile Scroll) keyword for display files

You use this field-level keyword in the subfile-control record format to return the relative record number of the subfile record that is at the top of the subfile when control is given back to your program.

This keyword has no parameters.

This field must have the keyboard shift attribute of signed numeric with zero decimal positions. It has to be 5 digits in length, and it must be defined as a hidden field. The hidden field will not display an input field on the screen.

This field is returned to your program as part of the input for this subfile control record. If control is returned to your program by pressing the enter key, then the value returned will be the relative record number of the top subfile record currently displayed.

With the ROLLUP or ROLLDOWN keywords, control is returned to the program when pressing a Page Up or a Page Down key will page beyond the end of the subfile. Without ROLLUP or ROLLDOWN, a message is displayed. If control is returned to your program because of the ROLLUP keyword, then the value returned will be the relative record number of the top subfile record on the next page. If control is returned to your program because of the ROLLDOWN keyword, then a 1 will be returned in the relative record number field.

Note: The ROLLUP keyword is the same as the PAGEDOWN keyword and the ROLLDOWN keyword is the same as the PAGEUP keyword.

SFLSCROLL is not allowed when SFLSIZ equals SFLPAG.

This keyword is valid only for the subfile-control record format.

This keyword is helpful when scroll bars are used (when SFLEND(*SCRBAR) is specified). When the user interacts with the scroll bar, the hidden field that contains SFLSCROLL contains the relative record number of the record the user wants displayed. Control is only returned to your program when the user attempts to scroll into parts of the subfile that are not written to, or when the Enter key is pressed. Another keyword that might be useful is SFLRCDNBR with *TOP as a parameter. After you have added records to the subfile, redisplay the subfile with the SFLRCDNBR in effect. Use the same number for this keyword that was returned to the SFLSCROLL keyword.

You cannot specify the SFLROLVAL, the SFLSCROLL and the SFLRCDNBR keywords for the same field.

Only one SFLSCROLL keyword is allowed in the subfile control record.

Option indicators are not valid for this keyword.

Example

The following example shows how to specify the SFLSCROLL keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A
     A          R SFLRCD                    SFL
     A            CTLFLD         1Y 0H      SFLCHCCTL
     A            F1             4A  O  6 10
     A          R SFLCTLRCD                 SFLCTL(SFLRCD)
     A                                      SFLSNGCHC
     A                                      SFLPAG(5) SFLSIZ(&SFLSIZ);
     A                                      SFLDSP SFLDSPCTL
     A                                      ROLLUP(10)
     A   10                                 SFLEND(*SCRBAR)
     A            F3             5S 0H      SFLSCROLL
     A            F2             4S 0H      SFLRCDNBR(CURSOR *TOP)
     A            SFLSIZ         5S 0P
     A                                 1 30'Panel Title'
     A                                 4  5'Multiple selection list:'
```

In this example, field F3 contains the relative record number of the subfile record that is at the top of the subfile when control is given back to the program.

SFLSIZ (Subfile Size) keyword for display files

You use this record-level keyword on the subfile-control record format to specify the number of records in the subfile.

The maximum number of records allowed is 9999. This keyword is required for the subfile-control record format.

The format of the keyword is:

```
SFLSIZ(number-of-records-in-subfile | &number-of-records-in-subfile-field);
```

The number-of-records-in-subfile parameter can be specified in 2 ways; as a number or program-to-system field. The program-to-system field must be defined with a length of 5 and data type S.

P-fields can be used for the size of the subfile when using SFLEND with the *SCRBAR parameter. The application can communicate to the IBM i operating system the number of records that the applications will be adding to the subfile. Therefore, the scroll bar can show a better picture of the subfile.

Note: The value for the p-field must be greater than the subfile page value. If the value of the p-field is not greater than the subfile page value, the size of the subfile will be page value plus one.

Subfile size equals subfile page

When you specify the same parameter values for SFLSIZ and the SFLPAG keyword, you can specify option indicators for fields in the subfile record format. (This is called field selection.)

When the subfile is built, the records can vary in length depending on which fields are selected, and each output operation places records into successive positions within the subfile. When the subfile is displayed, each record can require a different number of display lines. The number of records that actually fit in the subfile depends on the fields selected for each record written to the subfile.

If the last subfile record written to the subfile fits exactly into the subfile, a status message (CPF5003) is returned to your program. If the last subfile record written to the subfile overflows the subfile, a notify message (CPF5043) is returned to your program.

The specified SFLPAG value is increased to equal the maximum number of records that fit on the display if the number of subfile records to be displayed do not occupy a full display.

The SFLSIZ value is increased by the same value. For example, if SFLPAG(13) and SFLSIZ(13) are specified, and the subfile record format and SFLLIN value are specified such that three records can fit on a single display line, SFLPAG and SFLSIZ are increased to 15.

Option indicators are not valid for this keyword. Display size condition names are valid and are required if the number of records within the subfile changes depending on the display size. You cannot use display size condition names for this keyword when a program-to-system field is used as a parameter for it.

Subfile size does not equal subfile page

When you specify different parameter values for SFLPAG keyword and SFLSIZ, the SFLSIZ value specifies the number of records that can be placed into the subfile. If your program places a record with a relative record number larger than the SFLSIZ value into the subfile, the subfile is automatically extended to contain it (up to a maximum of 9999 records). The parameter value you specify should be large enough to accommodate the maximum number of records you normally have in the subfile.

Example

The following example shows how to specify the SFLSIZ keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A          R SFLR                         SFL
00020A  21        FIELD1        78      2  2
00030A  22        FIELD2        40      3  2
00040A*
00050A          R SFLCTLR                      SFLCTL(SFLR)
00060A                                         SFLPAG(5)
00070A                                         SFLSIZ(5)
00080A                                         SFLDSP
00090A                                         SFLDSPCTL
     A
```

Your program issues the following output operations:

Output operation to	Option indicators set	Result
SFLR	21 on 22 off	Only FIELD1 written to subfile
SFLR	21 on 22 on	FIELD1 and FIELD2 written to subfile
SFLR	21 off 22 on	Only FIELD2 written to subfile
(The IBM i operating system sends status message CPF5003 to your program.)		
SFLCTLR	No indicator necessary	Subfile displayed

The resulting display is as follows:

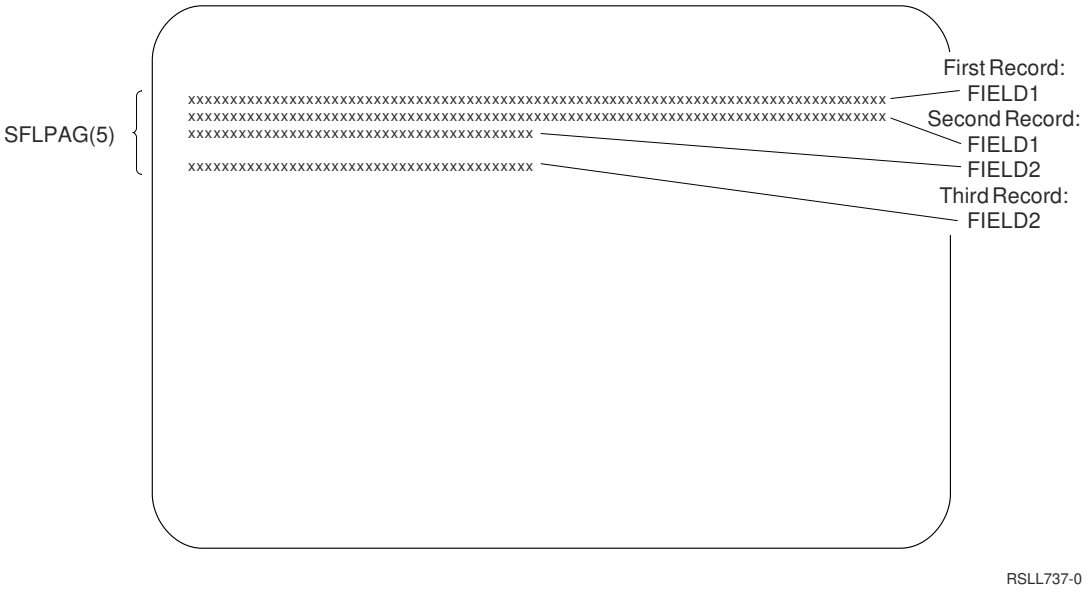


Figure 20. Subfile display

In Figure 20 on page 222, a fourth record cannot be written to the subfile because there is no room on the display for it (SFLPAG(5) has been specified in the DDS).

SFLSNGCHC (Subfile Single Choice Selection List) keyword for display files

You use this record-level keyword to define a subfile as a single-choice selection list. A *single-choice selection list* is a scrollable group of items from which the user can select only one item.

The format of this keyword is:

```
SFLSNGCHC[([*NORSTCSR | *RSTCSR] [*NOSLTIND | *SLTIND]
[*NOAUTOSLT | *AUTOSLT | *AUTOSLTENH])]
```

Parameters are optional and can be entered in any order.

The *RSTCSR parameter specifies whether the arrow keys should be allowed to move the selection cursor outside of the field. *RSTCSR specifies that the arrow keys will not cause the selection cursor to move outside of the selection list field. *NORSTCSR specifies that the arrow keys will cause the selection cursor to leave the field. If the SFLSNGCHC subfile control record is defined in a pulldown, the default is *RSTCSR. Otherwise, the default is *NORSTCSR.

The *SLTIND parameter specifies whether selection indicators are used when this selection list is displayed on a graphical display. *SLTIND specifies that the radio buttons should be used on graphical color displays as selection indicator. *NOSLTIND specifies that no selection indicator should be used on a graphical color display and only a selection cursor can be used to make a selection. The default is *NOSLTIND.

The *AUTOSLT parameter indicates if the ENTER key should automatically select the choice currently being indicated by the selection cursor. *NOAUTOSLT indicates that the user must select the choice. *AUTOSLTENH indicates that auto-select is only in effect if the device is connected to an enhanced controller. If the SFLSNGCHC subfile control record is defined in a pulldown, the default is *AUTOSLT. Otherwise, the default is *NOAUTOSLT.

A subfile containing the SFLSNGCHC keyword must:

- Contain only one output field
- Cannot contain input capable fields
- Can contain hidden fields

This optional keyword is valid only for the subfile-control record format.

The following subfile control record keywords cannot be specified on a record with the SFLSNGCHC keyword:

SFLDROP
SFLFOLD
SFLMLTCHC

The CHCAVAIL, CHCSLT and CHCUNAVAIL keywords can be used to indicate the color of the items within the selection list, when the list is displayed on a color display station. The CHCAVAIL keyword indicates the color of the items within the list which are available for selection. The CHCSLT keyword indicates the color of the selected item. The CHCUNAVAIL keyword indicates the items on the list which are not available for selection. These keywords can be used in a subfile control record only if SFLSNGCHC or SFLMLTCHC keywords are also used.

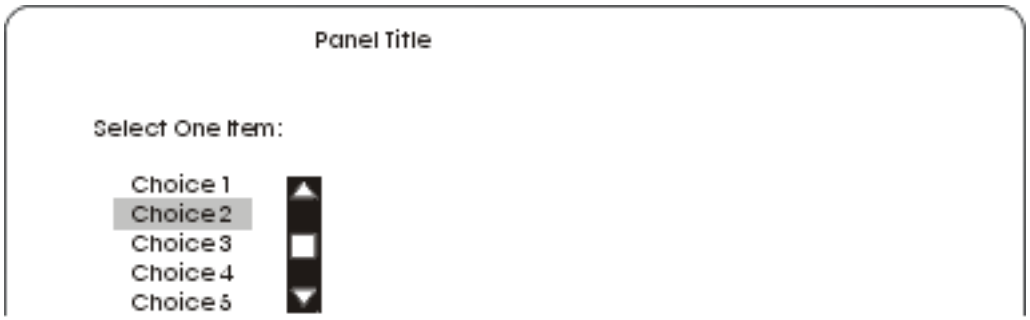
Option indicators are not valid for this keyword.

Example 1

The following example shows how to specify the SFLSNGCHC keyword:

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A
     A          R SFLRCD                      SFL
     A            CTLFLD         1Y 0H       SFLCHCCTL
     A            F1            10A  O  6 10
     A          R SFLCTLRCD                   SFLCTL(SFLRCD)
     A                                        SFLSNGCHC
     A                                        SFLPAG(5) SFLSIZ(&SFLSIZ);
     A                                        SFLDSP SFLDSPCTL
     A                                        ROLLUP(10)
     A   10                                   SFLEND(*SCRBAR)
     A            F3             5S 0H       SFLSCROLL
     A            F2             4S 0H       SFLRCDNBR(CURSOR *TOP)
     A            SFLSIZ         5S 0P
     A                                  1 30'Panel Title'
     A                                  4  5'Select One Item:'
```

In this example, when using a graphical display station attached to a controller that supports an enhanced interface for nonprogrammable workstations, a single-choice list looks like this:



Example 2

The following example shows how to specify what color the items on the list should have on a color display. Available items are displayed in red. A selected item is displayed in blue. Unavailable items are displayed in yellow. The CHCAVAIL, CHCSLT, and CHCUNAVAIL keywords can also be used to set the display attributes of the items on the list. See the description of these keywords in this topic collection for examples of setting display attributes.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A
```

```
     A          R SFLRCD                    SFL
     A            CTLFLD         1Y 0H      SFLCHCCTL
     A            F1            10A  O  6 10
     A          R SFLCTLRCD                 SFLCTL(SFLRCD)
     A                                      SFLSNGCHC
     A                                      SFLPAG(5) SFLSIZ(&SFLSIZ);
     A                                      SFLDSP SFLDSPCTL
     A                                      ROLLUP(10)
     A                                      CHCAVAIL((*COLOR RED))
     A                                      CHCSLT((*COLOR BLU))
     A                                      CHCUNAVAIL((*COLOR YLW))
     A   10                                 SFLEND(*SCRBAR)
     A            F3             5S 0H      SFLSCROLL
     A            F2             4S 0H      SFLRCDNBR(CURSOR *TOP)
     A            SFLSIZ         5S 0P
     A                                 1 30'Panel Title'
     A                                 4  5'Select One Item:'
```

SLNO (Starting Line Number) keyword for display files

You use this record-level keyword to specify a starting line number for the record format you are defining.

If specified, SLNO adjusts the actual line numbers for each field in the record format. If you do not specify SLNO, fields in the record format are displayed on the lines you specify for them in positions 39 through 41.

The format of the keyword is:

```
SLNO(n | *VAR)
```

You can specify one of two parameter values for this keyword:

- Specify *n*, where n is a value between 1 and 27. All fields in the record format are offset n-1 lines down the display from their specified locations. If SLNO(1) is specified, the record format must not contain a field starting at line 1, position 1.
- Specify *VAR to enable your program to set the starting line number at runtime before displaying the record format. At file creation time, the IBM i program sets the starting line number to one. A warning message appears at file creation time if the record contains a field starting at line 1, position 1. If your program does not set the starting line number or sets it to zero, the IBM i operating system assumes its value is one. If your program sets the starting line number to a value such that the first field in the record format does not all fit on the display, or sets it to a negative value, the IBM i operating system sends a notify message (CPF5002) to your program, and the record is not displayed. If the starting line number is set to one and the record format contains a field starting at line 1, position 1, the IBM i operating system sends an error message (CPF5398) to your program. The record is not displayed.

  To calculate the line on which a field is actually displayed, subtract one from the line number specified in positions 39 through 41 and add the starting line number to the result. The record format begins on the line specified with SLNO unless a field is defined at line 1, position 1. In that case, the beginning attribute byte is in the last position of the previous line and the starting line of the format is one less than that specified by SLNO.

  When *VAR is specified, no field in the record can occupy the last position on the display.

If a CLRL(nn) or CLRL(*END) keyword is also in effect for this record when it is to be displayed, lines on the display are cleared beginning with the starting line number for the format.

If you use the SLNO(*VAR) keyword with the OVERLAY keyword but without the CLRL keyword and then write the record several times, each time with a different starting number, the previous record is deleted before the new record displays.

The IBM i program checks the starting line number to determine whether the previous output operation to the record had the same starting line number if you use the SLNO keyword with the following keywords:

ERRMSG
ERRMSGID
PUTOVR
PUTRETAIN

If the starting numbers are the same, the actions specified by the ERRMSG, ERRMSGID, PUTOVR, or the PUTRETAIN keyword is performed.

If the starting line numbers are not the same, the ERRMSG, ERRMSGID, PUTOVR, or PUTRETAIN keyword is ignored, and the record format displays with the lines adjusted to the new value.

The SLNO keyword is not allowed in a record format that has one of the following keywords specified:

ASSUME
KEEP
SFL
SFLCTL
USRDFN

SLNO cannot be specified for the record format specified by the PASSRCD keyword.

Option indicators are not valid for this keyword.

Example

The following example shows how to specify the SLNO keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A          R RECORD1                  SLNO(*VAR)
00020A            FIELD1         5   I  2  2
00030A            FIELD2         5   B  3  2
00040A
00050A          R RECORD2                  SLNO(2)
00060A            FIELD3         5     10  2
00070A            FIELD4         5   B 10 13
     A
```

In this example, when the starting line number is zero or one, FIELD1 is displayed on line 2 and FIELD2 is displayed on line 3, as specified. When the starting line number is set to 2 in your program, FIELD1 is displayed on line 3 (2 - 1 + 2 = 3) and FIELD2 is displayed on line 4 (2 - 1 + 3 = 4).

FIELD3 and FIELD4 are always displayed on line 11 (2 - 1 + 10 = 11).

SNGCHCFLD (Single-Choice Selection Field) keyword for display files

You use this field-level keyword to define a field as a single-choice selection field.

A single-choice selection field is a field that contains a fixed number of choices from which a user can select one choice. The field appears as a vertical list of choices, with a single input field at the upper left, or as a group of radio buttons.

The format of this keyword is:

```
SNGCHCFLD[([*NORSTCSR | *RSTCSR]
[*NOAUTOSLT | *AUTOSLT | *AUTOSLTENH]
[*NOSLTIND | *SLTIND]
[*NOAUTOENT | *AUTOENT | *AUTOENTNN]
[[(*NUMCOL nbr-of-cols) | (*NUMROW nbr-of-rows)]
[(*GUTTER gutter-width)]])]
```

Parameters are optional. If none are specified, the single-choice field choices will be arranged in a single vertical column. The user will be allowed to move the selection cursor out of this field using the arrow keys.

The RSTCSR parameter specifies whether the arrow keys should be allowed to move the selection cursor outside of the selection field. *RSTCSR specifies that the arrow keys will not cause the selection cursor to move outside of the selection field. *NORSTCSR specifies that the arrow keys will cause the selection cursor to leave the selection field. The default is *NORSTCSR.

Note: An exception to the restrictions imposed by *RSTCSR happens if the selection field is the only field contained within a pull-down window. In that case, when the selection cursor is within the leftmost or rightmost columns, the left and right arrow keys will close the current pull-down window and open

the pull-down window associated with the menu-bar choice to the left or right of the current menu-bar choice.

The *RSTCSR parameter is ignored on displays that are not connected to a controller that supports an enhanced interface for nonprogrammable workstations.

The AUTOSLT parameter indicates if the ENTER key should automatically select the choice currently being indicated by the selection cursor. *NOAUTOSLT indicates that the user must select the choice. *AUTOSLTENH indicates that auto-select is only in effect if the device is connected to an enhanced controller. The default is `*AUTOSLT`.

The SLTIND parameter indicates whether selection indicators (such as radio buttons) should be displayed. *NOSLTIND specifies that the selection indicators should not be displayed. The default is `*SLTIND`.

Auto-enter will cause the record to be returned as soon as a choice is selected (as if the user had also pressed the enter key). The AUTOENT parameter indicates to what extent Auto-Enter should be enabled. *NOAUTOENT indicates that no auto-enter will be in effect. *AUTOENT will enable auto-enter on all displays unless a double digit selection number is required for any of the choices. *AUTOENTNN will enable auto-enter only if numeric selection of the choices is not required. If not specified, this parameter is set to *NOAUTOENT as default.

*NUMCOL specifies that this selection field should be displayed in multiple columns with the choices spread across the columns in this manner:

```
choice1   choice2   choice3
choice4   choice5   choice6
choice7   choice8   choice9
```

The nbr-of-cols portion of the parameter specifies how many columns the selection field should contain. Nbr-of-cols must be a positive integer and the entire single-choice selection field must be able to fit on the display when placed in the specified number of columns.

*NUMROW specifies that this selection field should be displayed in multiple rows with the choices spread across the columns in this manner:

```
choice1   choice4   choice7
choice2   choice5   choice8
choice3   choice6   choice9
```

The nbr-of-rows portion of the parameter specifies how many rows the selection field should contain. Nbr-of-rows must be a positive integer and the entire single-choice selection field must be able to fit on the display when placed in the specified number of rows.

The *GUTTER parameter is optional and specifies the number of spaces to be placed between columns of the single-choice selection field. It can only be specified if either *NUMCOL or *NUMROW has been specified and must follow the (*NUMxxx #) parameter. The gutter-width must be a positive integer of at least 2. If *GUTTER is not specified, the default of gutter-width is set to three spaces (including leading and trailing choice text attributes).

A field containing the SNGCHCFLD keyword must also contain one or more CHOICE keywords defining the choices for the field.

The field containing the SNGCHCFLD keyword must be defined as an input-capable field with data type Y, length equal to 2, and zero decimal positions. The position specified for the field is the position of the input field displayed to the left of the first choice, or of the uppermost radio button. On input, the field contains the number of the choice selected, or 0 if no choice was selected. On output, if the field contains a choice number, that choice is displayed as the default selection.

Provide a minimum of two spaces between the end of the previous field and the position specified for the single choice field. This allows an ending attribute for the previous field, and a beginning attribute for the single choice field. These attributes cannot overlap.

The following keywords can be specified on a field with the SNGCHCFLD keyword:

ALIAS
AUTO(RA)
CHANGE
CHCACCEL
CHCAVAIL
CHCUNAVAIL
CHCSLT[1]
CHCCTL
CHECK(ER)[2]
CHECK(FE)[3]
CHOICE
CHGINPDFT
COLOR[4]
DSPATR(RI UL BL CS HI ND PC)[4]
ERRMSG
ERRMSGID
INDTXT
PUTRETAIN
TEXT

Notes:

1. CHCSLT functions only if the single-choice selection field is displayed in a pull-down menu that does not display selection indicators (for example, PULLDOWN(*NOSLTIND) specified).
2. Check(ER) is not allowed with SNGCHCFLD if the AUTOENT or AUTOENTNN parameters have been specified.
3. Check(FE) applies only to a display attached to a controller that does not support an enhanced interface.
4. If the COLOR or DSPATR keyword is specified for a field with the SNGCHCFLD keyword, it applies only to the input field portion of the selection field on character-based displays.

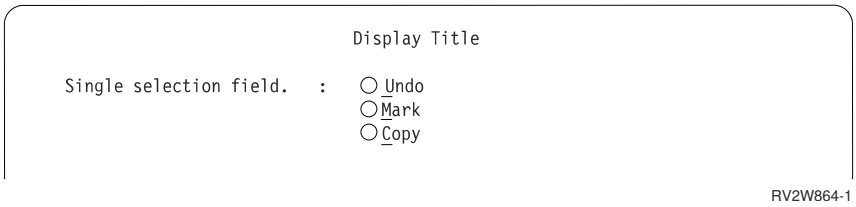
Option indicators are not valid for this keyword.

Example

The following example shows how to specify the SNGCHCFLD keyword:

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A
     A          R RECORD
     A            :
     A            :
     A                                  3  3'Single selection field.     :'
     A            F1             2Y 0B  3 35SNGCHCFLD
     A  01                                   CHOICE(1 '>Undo       ')
     A                                       CHOICE(2 &MARKTXT);
     A                                       CHOICE(3 '>Copy       ')
     A            MARKTXT       10A  P
     A
```

In this example, when using a graphical display station attached to a controller that supports an enhanced interface for nonprogrammable workstations, the selection fields look like this:



SYSNAME (System Name) keyword for display files

You use this field-level keyword to display the current system name as a constant (output-only) field that is 8 characters long.

You can specify the location of the field, the SYSNAME keyword, and, optionally, the COLOR, DSPATR, and TEXT keywords. Positions 17 through 38 must be blank.

This keyword has no parameters.

Option indicators are not valid for this keyword, although option indicators can be used to condition the field on which it is specified.

Example

The following example shows how to specify the SYSNAME keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A          R RECORD1
00020A                                2 62'SYSTEM:'
00030A                                2 72SYSNAME
     A
```

TEXT (Text) keyword for display files

You use this record- or field-level keyword to supply a text description (or comment) for the record format or field that is used for program documentation.

TEXT is valid for any record format or field, except a SFLMSGKEY or SFLPGMQ field.

The format of the keyword is:

```
TEXT('description')
```

The text must be enclosed in single quotation marks. If the length of the text is greater than 50 positions, only the first 50 characters are used by the high-level language compiler.

Option indicators are not valid for this keyword.

Example

The following example shows how to specify the TEXT keyword at the record and field levels.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A          R CUSMST                    TEXT('Customer Master Record')
00020A            FLD1           3  0      TEXT('ORDER NUMBER FIELD')
     A
```

TIME (Time) keyword for display files

You use this field-level keyword to display the current system time as a constant (output-only) field.

This keyword has no parameters.

You can specify only the location of the field, TIME, and optionally, the EDTCDE, EDTWRD, COLOR, DSPATR, or TEXT keyword. Positions 17 through 38 must be blank.

The edit word `'0_:__:__'` (_ represents a blank) is assumed for a TIME field. You can specify another edit word or one of the user-defined edit codes (5 through 9) to change the IBM-supplied editing.

Option indicators are not valid for this keyword, although option indicators can be used to condition the field with which it is specified.

Example

The following example shows how to specify the TIME keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A  20                            1 56TIME
00020A  21                            1 56TIME
00030A                                    EDTWRD('0 &HRS&;  &MINS&;  &SECS')
     A
```

In this example, the system time is 110645.

- If option indicator 20 is on, the time is displayed as:

```
11:06:45
```

- If option indicator 21 is on (and option indicator 20 is off), the time is displayed as:

```
11 HRS 06 MINS 45 SECS
```

TIMFMT (Time Format) keyword for display files

You use this field-level keyword to specify the format of a time field. This keyword is valid for time fields (data type T).

The format of the keyword is:

```
TIMFMT(time-format)
```

The following table describes the valid time formats and their default separators.

Format name	Time format parameter	Time format and separator	Field length	Example
Hours:Minutes:Seconds	*HMS	hh:mm:ss	8	14:00:00
International Standards Organization	*ISO	hh.mm.ss	8	14.00.00
IBM USA Standard	*USA	hh:mm AM or hh:mm PM	8	2:00 pm
IBM European Standard	*EUR	hh.mm.ss	8	14.00.00
Japanese Industrial Standard Christian Era	*JIS	hh:mm:ss	8	14:00:00

If you do not specify the TIMFMT keyword, the default is *ISO.

If you specify the time-format parameter value as *ISO, *USA, *EUR, or *JIS, you cannot specify the TIMSEP keyword. These formats have fixed separators.

The format of DFT, DFTVAL, and MAPVAL keyword values must match the format that the TIMFMT keyword specifies. If the TIMFMT keyword is set to *ISO as default, these values must have a format of *ISO.

The TIMFMT keyword overrides the job attribute for a time field. It does not change the system default.

It is the responsibility of the high-level language and the application to format the time field according to the format specified for the TIMFMT keyword and use the separators specified on the TIMSEP keyword. The system does not format fields on output. The system validates the time field on input according to the format the TIMFMT keyword specifies and the separator that the TIMSEP keyword specifies.

Option indicators are not valid for this keyword, although you can use option indicators to condition the field for which it is specified.

Example

The following example shows how to specify the TIMFMT keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A
00020A          R RECORD
00030A            TIMFLD1         T  B  5  2TIMFMT(*ISO)
00040A            TIMFLD2         T  B  5 22TIMFMT(*USA)
```

```
00050A            TIMFLD3         T  B  5 42TIMFMT(*HMS) TIMSEP(',')
     A
```

If you want to display 2 o'clock p.m., the following values are displayed where RECORD1 appears.

```
TIMFLD1    14.00.00
TIMFLD2    02:00 PM
TIMFLD3    14,00,00
```

TIMSEP (Time Separator) keyword for display files

You use this field-level keyword to specify the separator character used for a time field. This keyword is valid only for time fields (data type T).

The format of the keyword is:

```
TIMSEP(*JOB | 'time-separator')
```

The time-separator parameter specifies the separator character that appears between the hour, minute, and second values. Valid values are a colon (:), a period (.), a comma (,), and a blank ( ). Single quotation marks must enclose the parameter.

If you specify the *ISO, *USA, *EUR, or *JIS time-format values for the TIMFMT keyword, you should not specify the TIMSEP keyword. These formats have fixed separators.

If you do not specify the TIMSEP keyword and you specify TIMFMT as a format that does not have a fixed date separator, TIMSEP is set to *JOB as default.

If you specify *JOB or if TIMSEP is set to *JOB as default, the high-level language and the application handle the separator as a colon (:). On output the system converts the separator that was specified by the Time Separator Job Definition Attribute. On input, the system converts the separator to a colon (:) before passing control to the application.

The separator for DFT, DFTVAL, and MAPVAL keyword values must match the separator that the TIMSEP keyword specifies. If the TIMSEP keyword specifies *JOB the TIMSEP keyword is set to *JOB as default, these values must a colon ':' format.

The TIMSEP keyword overrides the job attribute for a time field. It does not change the system default.

It is the responsibility of the high-level language and the application format the time field according to the format specified for the TIMFMT keyword and use the separators specified for the TIMSEP keyword. The system does format fields on output. The system validates the time field on input according to the format that the TIMFMT keyword specifies and the separator that the TIMSEP keyword specifies.

Option indicators are not valid for this keyword, although you can use option indicators to condition the field for which it is specified.

Example

The following example shows how to specify the TIMSEP keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A
00020A          R RECORD
00030A            TIMFLD1         T         TIMFMT(*HMS) TIMSEP(' ')
00040A            TIMFLD2         T         TIMFMT(*HMS) TIMSEP('.')
00050A            TIMFLD3         T         TIMFMT(*HMS) TIMSEP(*JOB)
```

If you want to display 2 o'clock p.m. and the time separator that is defined by the Definition Attribute is :, the following values are displayed when RECORD1 appears.

```
TIMFLD1    14 00 00
TIMFLD2    14.00.00
TIMFLD3    14:00:00
```

UNLOCK (Unlock) keyword for display files

You use this record-level keyword to specify that the IBM i operating system is to unlock the keyboard immediately after issuing an input operation to the record format you are defining.

Without the UNLOCK keyword, the IBM i operating system leaves the keyboard locked after reading the data on the display. The workstation user cannot key data into input-capable fields while the data that has just been read is being processed.

The format of the keyword is:

```
UNLOCK[(*ERASE) | (*MDTOFF)] | [(*ERASE *MDTOFF)] |  [(*MDTOFF *ERASE)]
```

The parameter values *ERASE and *MDTOFF are optional. If you do not specify any parameter value, *ERASE is the default.

When your program sends an input operation, the following sequence of operations typically occurs:

1. The keyboard is unlocked (if it is not already unlocked) to allow the workstation user to key into input-capable fields on the display.
2. The workstation user presses the Enter key (or a valid function key).
3. Modified data tags (MDTs) for input-capable fields in the record format are set on if they have been typed into or if they were displayed with the DSPATR(MDT) keyword in effect.
4. When the input operation is completed, the parameter values for UNLOCK affect the input-capable fields with MDTs set on as described in the following sections.

UNLOCK (without GETRETAIN) or UNLOCK(*ERASE)

The keyboard remains unlocked, input-capable fields on the display are erased, and their MDTs remain on following the input operation.

UNLOCK(*MDTOFF) or UNLOCK (with GETRETAIN)

The keyboard remains unlocked, input-capable fields on the display are not erased, and their MDTs are set off following the input operation.

UNLOCK(*ERASE *MDTOFF) or UNLOCK(*MDTOFF *ERASE)

The keyboard remains unlocked, input-capable fields on the display with their MDTs set on are erased, and their MDTs are set off following the input operation.

The GETRETAIN keyword is ignored and an error message results at file creation time if the GETRETAIN keyword is specified with UNLOCK(any parameter).

Note: This keyword does not prevent your program from issuing an output operation immediately after an input operation. However, the keyboard is unlocked and the workstation user can be typing input data when the output operation changes the display.

Option indicators are not valid for this keyword.

Example

The following example shows how to specify the UNLOCK keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A          R RECORD1                   UNLOCK(*ERASE)
00020A            FLD1          4   B  2  2
00030A            FLD2          4   B  3  2
     A
00040A          R RECORD2                   UNLOCK(*MDTOFF)
00050A            FLD21         4   B  4  2
00060A            FLD22         4   B  5  2
     A
00070A          R RECORD3                   UNLOCK(*ERASE *MDTOFF)
```

```
00080A            FLD31          4   B  6  2
00090A            FLD32          4   B  7  2
     A
```

Related reference

GETRETAIN (Get Retain) keyword for display files

You use this record-level keyword with the UNLOCK keyword to specify that the IBM i operating system is not to erase input-capable fields on input operations as described under the UNLOCK keyword.

USER (User) keyword for display files

You use this field-level keyword to display the user profile name for the current job as a constant (output-only) field that is 10 characters long.

You can specify the location of the field, the USER keyword, and, optionally, the COLOR, DSPATR, and TEXT keywords. Positions 17 through 38 must be blank.

This keyword has no parameters.

Option indicators are not valid for this keyword, although option indicators can be used to condition the field on which it is specified.

Note: For a System/36 environment Multiple Requester Terminal (MRT) job, the displayed user profile name is the same user profile name of the interactive job for the display station where the display file is shown.

Example

The following example shows how to specify the USER keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A          R RECORD1
00020A  10                             2 12'USER:'
00030A  10                             2 20USER
00040A                                     DSPATR(HI)
00050A N10                            15 18'USER:'
00060A N10                            15 26USER
00070A                                     DSPATR(HI)
     A
```

In this example, if indicator 10 is on, the user name is displayed on line 2 starting in column 20. If indicator 10 is off, the user name is displayed on line 15 starting in column 26.

USRDFN (User-Defined) keyword for display files

You use this record-level keyword to specify that the data for this record is in the form of a user-defined data stream.

This keyword has no parameters.

No fields are valid for this record because the data stream formats the display. No file- or record-level keywords apply to this record except INVITE, KEEP, PASSRCD, HLPRTN, HELP, HLPCLR, PRINT, OPENPRT, and TEXT. However, the HELP, HLPRTN, and INVITE keywords will only apply if they are specified on this record. They will not apply if they are specified at the file-level. Help specifications are valid for this record. Once you do an output operation for this record, the IBM i program no longer holds knowledge of the status of the records on the device. You should have in-depth knowledge of the device before using this keyword.

Option indicators are not valid for this keyword.

Example

The following example shows how to specify the USRDFN keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00020A          R USRREC                  USRDFN
```

```
00030A
     A
```

USRDSPMGT (User Display Management) keyword for display files

You use this file-level keyword to specify that all data written to the display is held until the data is overwritten or cleared by using the CLRL keyword.

This keyword has no parameters.

Related concepts

System/36 environment considerations for display files
The User Display Management (USRDSPMGT) keyword causes the record formats in the display file to function similarly to System/36 SFGR display formats.

USRRSTDSP (User Restore Display) keyword for display files

You use this record-level keyword on a window record to specify that the application will manage the display.

Window records are not automatically removed. If this keyword is not specified, the system saves and restores the underlying display when a window record is displayed.

This keyword has no parameters.

The WINDOW keyword must be specified on the same record as the USRRSTDSP keyword. The USRRSTDSP keyword functions only when the window keyword defines a window. The USRRSTDSP keyword does not function if the window keyword specified a record format name.

Option indicators are valid for this keyword.

Example

The following example shows how to specify the USRRSTDSP keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A          R APPRCD
     A            FIELD1        10A  O  5 40
     A            FIELD2         5S 0B  6 45
     A          R WINDOW1                  WINDOW(6 15 9 30)
     A  25                                 USRRSTDSP
     A            FIELD1         5A  B  2  2
     A            FIELD2        20S  B  8  5
     A
```

In this example, suppose APPRCD is already on the display. If indicator 25 is set on when WINDOW1 is written to the display, the system does not save the underlying display (which contains APPRCD). When the user exits WINDOW1, the application must restore the underlying display, possibly by rewriting APPRCD to the display.

Note: With USRRSTDSP, there is no limit on the number of windows. The limit is 12 without using the USRRSTDSP keyword.

Related reference

WINDOW (Window) keyword for display files

You use this record-level keyword to specify that the record format you are defining will be displayed using a window.

VALNUM (Validate Numeric) keyword for display files

You use this file-level, record-level, or field-level keyword to enhance the error checking that is performed against fields with the data type numeric-only.

When specified on a numeric only field, this keyword causes an error message to be returned if the user attempts to embed a SPACE, PLUS SIGN or MINUS SIGN between numeric digits in the field or when the PLUS SIGN or MINUS SIGN precedes the numeric digits.

This keyword has no parameters.

The field containing the VALNUM keyword must be defined as an input-capable field with the data type Y.

Option indicators are not valid for this keyword.

Example

The following example shows how to specify the VALNUM keyword:

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A          R RECORD
00020A            F1             5Y 0B  3  4VALNUM
00030A            F2             5Y 0B  4  4
```

In this example, field F1 does not allow the user to embed a space, plus sign, or minus sign within the numeric value, nor to precede the numeric value with a plus sign or minus sign. Field F2 is treated as described in “Data type and keyboard shift for display files (position 35)” on page 11.

VALUES (Values) keyword for display files

You use this field-level keyword to specify a list of values that are valid for the user to type into the field.

The IBM i operating system performs an implied equals test on the data typed in against the value(s) you specify here. Note that the IBM i operating system performs this checking only if the field is changed by the workstation user or if its modified data tag (MDT) is set on using DSPATR(MDT).

Note: The CHKMSGID keyword information defines user-specified error messages.

The format of the keyword is:

```
VALUES(value-1 [value-2... [value-100]])
```

There can be 1 to 100 values; specify them as parameter values with the keyword and separate them by at least one blank.

Note: You cannot specify more than 5000 characters in a single DDS statement. Therefore, you cannot specify character values that cause VALUES to be longer than 5000 characters. If you specify other keywords for the same field, they also count toward the 5000-character limit. For example, specifying DSPATR(HI) for the field reduces the number of characters left for VALUES.

A value can be a numeric or a character value, corresponding in length to the field that is to be tested. A character value must be enclosed in single quotation marks. A numeric value is restricted to the digits 0 through 9 and can be preceded by a minus sign (-) for negative values. Alignment is on the low-order character position.

Defining a numeric field for display files

When a workstation user types in data, the IBM i operating system aligns the entered characters according to the number of decimal positions in the field. Leading and trailing blanks are filled with zeros when the field is passed to your program. If no decimal character is typed, the IBM i program places a decimal character to the right of the farthest right character typed in. For example, for a numeric field with

a length of 5 (specified in position 34) and 2 decimal positions (specified in position 37), 1.2 is interpreted as 001.20, and 100 is interpreted as 100.00.

The compare is based on the value as it is passed to your program (for example, right-aligned and padded or left-aligned and padded).

You cannot specify VALUES on a floating-point field (F in position 35).

Option indicators are not valid for this keyword.

Example

The following example shows how to specify the VALUES keyword for a character and numeric values test.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A            RESPC         11   I  8  2VALUES('A' 'B' 'C' 'D')
00020A            RESPN         31  0I  9  2VALUES(33 -42 01)
00030A            DECFLD         1  2I 11  2VALUES(1.2 100)
     A
```

Related reference

CHKMSGID (Check Message Identifier) keyword for display files
You use this field-level keyword to identify an error message that is issued when a validity check error is detected.

VLDCMDKEY (Valid Command Key) keyword for display files

You use this file-level or record-level keyword to specify that the IBM i licensed program is to set on the specified response indicator when any valid command key other than the Enter key is pressed by the workstation user.

One use of this function is to perform a simple test to determine if the workstation user has requested a function you want to watch for in your program.

See “System/36 environment considerations for display files” on page 247 for information about how to specify the VLDCMDKEY keyword in files that are used in the System/36 environment.

The format of the keyword is:

```
VLDCMDKEY(response-indicator ['text'])
```

The response-indicator parameter is required.

The optional text is included on the computer printout created at program compilation to explain the intended use of the indicator. This text's only function in the file or the program is a comment. The single quotation marks are required. If you specify more than 50 characters between the single quotation marks, the text is truncated to 50 characters on the program computer printout.

For a command key to be considered valid, you must have activated the key by specifying it with one of the following keywords:

Keyword
Comments

ALTHELP(CAnn)
With or without response indicator on the HELP keyword. Causes the command attention key specified to be considered a valid command key.

ALTPAGEUP(CFnn)
With or without response indicator on the PAGEUP keyword. Causes the command function key specified to be considered a valid command key if PAGEUP is also specified.

ALTPAGEDWN(CFnn)
With or without response indicator on the PAGEDOWN keyword. Causes the command function key specified to be considered a valid command key if PAGEDOWN is also specified.

CAnn
With or without response indicator.

CFnn
With or without response indicator.

CLEAR
With or without response indicator.

HELP
Valid only when HELP key is passed back to application as follows:

- HELP and HLPRTN (with or without a response indicator).
- HELP (with or without a response indicator) and no help areas are defined for any records currently being displayed.

HOME
With or without response indicator.

PAGEDOWN
With or without response indicator.

PAGEUP
With or without response indicator.

PRINT
Valid only when PRINT key is passed back to application as follows:

- PRINT (with a response indicator).
- PRINT (*PGM).

ROLLUP
With or without response indicator.

ROLLDOWN
With or without response indicator.

Option indicators are not valid for this keyword.

Example

The following example shows how to specify the VLDCMDKEY keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A          R REC1                      VLDCMDKEY(90 'Any valid key')
00020A                                      CA01(91)
00030A                                      CA02(92)
00040A                                      CA03(93)
00050A                                      CLEAR(94)
     A
```

In this example, Indicator 90 is set on if any of four keys (CA01, CA02, CA03, or Clear) is pressed.

WDWBORDER (Window Border) keyword for display files

You use this file-level or record-level keyword to specify the color, display attributes, and characters used to form the border of a window.

The format of the keyword is:

```
WDWBORDER([color] [display-attribute] [characters])
```

At least one parameter must be specified.

The color parameter specifies the color of the border characters on a color display station (3179 Models C1 and C2, 5292 Color display stations only, or 5555 Models C01 and F01). The parameter is specified as an expression of the form (*COLOR value).

The valid values for the color parameter are:

Value
Meaning

BLU
Blue

GRN
Green

WHT
White

RED
Red

TRQ
Turquoise

YLW
Yellow

PNK
Pink

If the color parameter is not specified, the default is BLU. This parameter is ignored if it is specified for a window on a monochrome display.

The display-attribute parameter specifies the display attributes of the border characters. The parameter is specified as an expression of the form (**DSPATR [value1 [value2 [value3...]]]*). If more than one DSPATR value is used, they are combined to form one DSPATR that is used for the entire border.

The valid values for the display-attributes values are:

Value
Meaning

BL
Blink

CS
Column separator

HI
High intensity

ND
Nondisplay

RI
Reverse image

UL
Underline

There is no default for display-attributes.

Note: Display-attributes CS, HI, and BL can cause fields on 5292, 3179, and 3197 Model C1 and C2 display stations to appear as color fields. Display-attributes HI, RI, and UL cause a border not to be displayed.

The characters parameter specifies the characters that make up the border. The parameter is specified as an expression of the form (*CHAR 'border-characters'). The border character value is an 8-character string that defines the border characters in the following order:

```
top-left-corner
top-border
top-right-corner
left-border
right-border
bottom-left-corner
```

```
bottom-border
bottom-right-corner
```

If this parameter is not specified, the default border characters are period (.) for the upper-left and right corners and the top and bottom borders, colon (:) for the left and right borders and lower-left and right corners. Although any displayable character can be specified as a border character, it is recommended that you use invariant characters.

The following table shows the invariant characters:

Table 14. Character set for system data

Hexadecimal	Character	Description
40		Blank
4B	.	Period
4C	<	Less than sign
4D	(	Left parenthesis
4E	+	Plus sign
50	&	Ampersand
5C	*	Asterisk
5D	)	Right parenthesis
5E	;	Semicolon
60	-	Minus sign
61	/	Slash
6B	,	Comma
6C	%	Percent sign
6D	_	Underline
6E	>	Greater than sign
6F	?	Question mark
7A	:	Colon
7D	'	Single quotation mark
7E	=	Equal sign

Note: In addition, you can use any of the following characters:

- Uppercase alphabetic characters: A through Z
- Numeric characters: 0 through 9

If the WDWBORDER keyword is specified at the record level, the WINDOW or PULLDOWN keyword must also be specified on the same record. If a WINDOW keyword that references another window is also specified, a warning message is issued.

Option indicators are valid for this keyword.

You can specify more than one WDWBORDER on a record. If you specify the WDWBORDER keyword more than once at the file level or at a record level, the parameters for the keywords that are in effect are combined on the same level. If different values are specified for the same parameter, the parameter value of the first keyword is used.

If the WDWBORDER keyword is specified both at the file level and on a window or pull-down definition record, the parameter values defined at both levels are combined. If different values are specified for the same parameter, the parameter value at the record level is used.

Example

The following example shows how to specify the WDWBORDER keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A  01                                WDWBORDER((*COLOR PNK) +
     A                                    (*DSPATR BL))
     A          R RECORD1                 WINDOW(6 15 9 30)
     A N01                                WDWBORDER((*COLOR GRN))
     A  01                                WDWBORDER((*COLOR RED))
     A            FIELD1         5A  B  2  2
     A            FIELD2        20A  B  8  5
     A          R RECORD2                 WINDOW(8 20 9 30)
     A                                    WDWBORDER((*COLOR YLW) +
     A                                     (*DSPATR RI))
     A            FIELD3         5A  B  2  2
     A            FIELD4        20A  B  8  5
     A            LINE           2S 0P
     A            POS            2S 0P
     A          R RECORD3                 WINDOW(&LINE &POS 9 30)
     A                                    WDWBORDER((*CHAR +
     A  02                                 '+-+||+-+'))
     A            FIELD3         5A  B  2  2
     A            FIELD4        20A  B  8  5
     A            LINE           2S 0P
     A            POS            3S 0P
     A
```

If the window defined by RECORD1 is written to the display with indicator 01 set off, it has a green border constructed of colons for the vertical borders and periods for the horizontal borders. If indicator 01 is set on, the window has a blinking red border.

When the window defined by RECORD2 is written to the display, it has a yellow border displayed in reverse image constructed of the default border characters.

When the window defined by RECORD3 is written to the display, the following output appears:

- If indicator 02 is set on and indicator 01 is off, the window has a blue border constructed of dashes for the top and bottom borders, vertical bars for the left and right borders, and plus signs for the corners.
- If indicator 02 is set off and indicator 01 is set on, the window has a pink border.
- If indicators 01 and 02 are on, the window has a pink border constructed of dashes for the top and bottom borders, vertical bars for the left and right borders, and plus signs for the corners.

Related reference

COLOR (Color) keyword for display files
You use this keyword to specify the color of a field on a color display.

WDWTITLE (Window Title) keyword for display files

You use this record-level keyword to specify the text, color, and display attributes for a title that is embedded within the top or bottom border of a window.

The format of the keyword is:

```
WDWTITLE([title-text] [title-text-color]
[title-text-display-attribute]
[*CENTER | *LEFT | *RIGHT]
[*TOP | *BOTTOM])
```

At least one parameter must be specified.

The title-text parameter is an optional parameter which specifies the text that will be placed in the border of the window. The length of the text should be less than or equal to the window-positions parameter of the associated WINDOW definition record. If blanks are placed at the beginning of the text string, the

title will be shortened so there are a equal number of blanks at the end. If the text string is too long (> window-positions), it will be truncated on the right. The parameter is specified as an expression of the form (*TEXT value) where **value** can be specified in one of two forms:

- As a character string: 'Title text '
- As a program-to-system field: &field-name

The field-name specified must exist in the window record and must be defined as a character field with usage P.

Notes:

1. A GRAPHIC literal should not be used for the title-text parameter.
2. If the title characters are blanks, then a blank title will be displayed.
3. If the title characters are nulls, then no title will be displayed.

The title-text-color specifies the color of the title text on a color display. The parameter is specified as an expression of the form (*COLOR value).

The valid values for the title-text-color parameter are:

Value
Meaning

BLU
Blue

GRN
Green

WHT
White

RED
Red

TRQ
Turquoise

YLW
Yellow

PNK
Pink

If the title-text-color parameter is not specified, it is set to the color of the border as default. The parameter is ignored if it is specified for a window on a noncolor display.

The title-text-display-attribute specifies the display attributes of the title text. The parameter is specified as an expression in the form (**DSPATR [value1 [value2 [value3...]]]*). If more than one DSPATR is used, they are combined to form one DSPATR that is used for the title text.

The valid values for the title-text-display-attribute values are:

Value
Meaning

BL
Blink

CS
Column separator

HI
High intensity

ND
Nondisplay

RI
Reverse image

UL
Underline

If the title-text-display-attribute parameter is not specified, it is set to the text attribute of the border as default.

If neither the title-text-color nor title-text-display-attribute parameter is specified, the window border will flow up to the first character of the window title and resume immediately after the last character. If either parameter is specified, there will be a space immediately before and after the window title.

The *CENTER/*LEFT/*RIGHT parameter specifies whether the window title should be aligned to the CENTER, LEFT or RIGHT of the window border. If not specified, the window title will be aligned in the CENTER of the window border if the next parameter is *TOP or to the LEFT of the window border if the next parameter is *BOTTOM.

The *TOP/*BOTTOM parameter specifies if the text should be imbedded in the top or bottom border. If not specified, the text will be placed in the top border.

Notes:

- Not all controllers support alignment. On those controllers that do not, the title is centered.
- If *BOTTOM is specified on an enhanced controller that does not support text in the bottom windows border, the WDWTITLE keyword is ignored.
- By specifying ENHDSP(*NO) on the CRTDSPF or CHGDSPF command, the *BOTTOM, *LEFT, and *RIGHT parameters always work. However, all the other enhanced user interface functions are lost.

Option indicators are valid for this keyword.

The WDWTITLE keyword can only be specified on a record that contains a WINDOW keyword (in the definition format). If a WINDOW keyword that references another window is also specified, a warning message is issued.

You can specify more than one WDWTITLE on a record. If you specify the WDWTITLE keyword more than once at the record level, the parameters for the keywords that are in effect are combined. If different values are specified for the same parameter, the parameter value of the first keyword is used.

Example

The following example shows how to specify the WDWTITLE keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A
     A          R RECORD1                   WINDOW(6 15 9 30)
     A N01                                  WDWTITLE((*TEXT &TTL1) (*COLOR GRN))
     A  01                                  WDWTITLE((*TEXT &TTL1) (*COLOR RED))
     A            FIELD1         5A  B  2  2
     A            FIELD2        20A  B  8  5
     A            TTL1          10A  P
     A
     A          R RECORD2                   WINDOW(8 20 9 30)
     A                                      WDWTITLE((*TEXT &TTL2) +
     A                                               (*COLOR YLW) +
     A                                               (*DSPATR RI))
     A            FIELD3         5A  B  2  2
     A            FIELD4        20A  B  8  5
     A            TTL2          10A  P
     A
```

If the window defined by RECORD1 is written to the display, it will have whatever text is contained within TTL1 imbedded within the top border of the window. If indicator 01 is set off, this text will be green. If indicator 01 is set on, the text will be red.

When the window defined by RECORD2 is written to the display, the text contained within TTL2 will be imbedded within the top border of the window. This text will be display in reverse image and yellow.

WINDOW (Window) keyword for display files

You use this record-level keyword to specify that the record format you are defining will be displayed using a window.

A window is information that overlays part of the display. A window is typically smaller than the actual workstation display, and can be positioned anywhere on the display.

The WINDOW keyword has two formats that can be used. These formats do the following tasks:

- Define a window by specifying the location and size of a window; this is known as a window definition record.
- Refer to a record format name where the window location and size have been defined; this is known as a window-reference record.

The format for the keyword is specified as follows:

```
WINDOW(start-line | &start-line-field start-position
| &start-position-field window-lines window-positions
[*MSGLIN | *NOMSGLIN])
[*RSTCSR | *NORSTCSR])
or
WINDOW(*DFT window-lines window-position
[*MSGLIN | *NOMSGLIN]
[*RSTCSR | *NORSTCSR])
```

Specify this format of the WINDOW keyword to define a window. The record format you are defining is displayed in this window. Up to 12 windows can be shown on the display at one time. You can define more than 12 windows in DDS, but only 12 can be displayed at the same time. However, if USRRSTDSP is specified, the number of windows is unlimited. All fields defined in this record must fit within the window.

The parameters specify the following items:

- The number or the name of a field containing the number of the line that is to contain the upper-left corner of the window border. If a field name is specified, the field must exist in the record format and the field must be defined as a signed numeric (data type S) and program-to-system (usage P) field with length no greater than 3.
- The number or the name of a field containing the number of the position that is to contain the upper-left corner of the window border. If a field name is specified, the field must exist in the record format and the field must be defined as a signed numeric (data type S) and program-to-system (usage P) field with length no greater than 3.
- The number of window-lines within the window. The window-lines can be no more than the available lines for the display size minus 2. This is because the upper and lower window borders each occupy one line.

  The last window-line in a window is used as the message line and cannot contain any fields. For example, if a WINDOW keyword is coded that specifies 10 window-lines for the window, only nine of those lines can contain fields; the 10th line is the message line.
- The number of window-positions within the window. The window-positions can be no more than the available positions for the display size minus 4. This is because both right and left borders need an attribute byte inside the window. An attribute byte exists between the border character and the available window positions. For DBCS-capable windows, the system might need an additional 2 bytes on each side of the window for a shift-out character and shift-in character for any underlying DBCS fields.
- The MSGLIN parameter specifies if a window contains a message line. If this parameter is not specified, the default is *MSGLIN. *NOMSGLIN moves the message out of the window and places it at the bottom of the display or where the MSGLOC keyword defines the location. The last usable line in the window is reserved for error messages; no records are displayed there. If the error message is longer than the line, it is truncated to fit.
- The *RSTCSR parameter specifies if the user should be allowed limited function when the cursor is outside of the window. When *NORSTCSR is specified and the cursor is outside of the window, the user will be allowed to press a function key and have it function as if the cursor were within

the window. When the user specifies *RSTCSR on a controller that supports enhanced interface for nonprogrammable workstations, the user will be able to move the cursor out of the window (except with a mouse). For other workstations, when the user attempts to press a function key while the cursor is outside of the window, the user will receive a beep and the cursor will be placed inside the window. Control will not be returned to the application. *RSTCSR is the default.

The special value, *DFT, specified in place of the start-line and start-position parameters, indicates that the system will determine the start line and start position of the window. The window is positioned relative to the cursor location, similar to application help windows with variable starting locations. More information about the rules the system uses to position the window can be found in the Application Display Programming book  .

The second format for the WINDOW keyword is `WINDOW(record-format-name)`.

Specify this format of the WINDOW keyword to display the record format you are defining in a window that is defined on another record format.

The parameter specifies the record format name that has the window attributes specified. The record format that uses this parameter is displayed in the window defined on the referenced record.

The field locations specified within a record format with the WINDOW keyword are relative to the first usable window location in the upper-left corner of the window. The first usable window location is on the first line below the upper border and two positions to the right of the left border (an ending attribute byte occupies the first byte to the right of the border).

When a window is displayed, any records currently on the display are suspended if USRRSTDSP is not specified. The suspended records can be visible around the sides of the window. Input is allowed only within the active window. To remove the window from the display, a record can be written to an underlying window or a non-window record must be overlaid on the display.

The WINDOW keyword is not allowed on a record format that has any one of the following keywords specified:

ALWROL	PULLDOWN
ASSUME	SFL
MNUBAR	USRDFN

Note: The WINDOW keyword is allowed on a record with the SFLCTL keyword. This allows subfiles to be displayed within a window.

WINDOW cannot be specified for the record format specified by the PASSRCD keyword.

The ERRSFL keyword is ignored for records that have the WINDOW keyword specified.

The MSGLOC keyword is ignored for records that have the WINDOW keyword specified, unless NOMSGLIN is specified.

If a record format has both a WINDOW and WDWBORDER keyword specified, specify the start-line, start-position, window-lines, and window-positions parameters on the WINDOW keyword. The WINDOW keyword should not specify the record-format-name parameter.

Option indicators are not valid for this keyword. However, display size condition names can be used.

Example 1

The following example shows how to specify the WINDOW keyword to define a window.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A          R WINDOW1                   WINDOW(4 20 9 30 *NORSTCSR)
     A            FIELD1         8A  B  5 10
     A            FIELD2        10A  B  6 10
     A          R WINDOW2                   WINDOW(*DFT 9 30 *NOMSGLIN)
     A
```

When the WINDOW1 record is displayed, the upper-left corner of the window border is on line 4 position 20 of the display. The lower-right corner of the border is located 10 lines lower than the upper border and 33 positions to the right of the left border.

- Lower border line = upper border line + window-lines + 1
- Right border position = left border position + window-positions + 3

The FIELD1 field starts 2 lines lower than the upper border and 11 positions (the ending attribute byte for the border character has been taken into account) to the right of the left border character (line 6, position 31 on the display).

- Actual field line = upper border line + line number of field
- Actual field position = left border position + position of field + 1

The FIELD2 field starts 6 lines lower than the upper border and 11 positions to the right of the left border (line 10, position 31 on the display).

If the cursor is moved outside of the window, the function keys will remain active.

When the WINDOW 2 record is displayed, the upper-left corner of the window border is at the cursor position during run time. The message line does not appear inside the window, it appears at the bottom of the display.

If the cursor is moved outside of the window, the function keys are inactive. If the user presses a function key, they will receive a beep and the cursor will be place within the window.

Example 2

The following example shows how to use the WINDOW keyword to display multiple records in the same window.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A          R WINDOW1                   WINDOW(&LINE &POS 9 30)
     A            USERID         8A  O  2 10
     A            LINE           2S 0P
     A            POS            3S 0P
     A
     A          R RECORD1                     WINDOW(WINDOW1)
     A                                        OVERLAY
     A            FIELD1          5A  B  7  2
     A            FIELD2         20A  B  8  5
     A
     A          R RECORD2                     WINDOW(WINDOW1)
     A            FIELD3         10A  B  2  2
     A            FIELD4          8A  B  8  5
     A            FIELD4          8A  B  8  5
     A
```

When the WINDOW1 record is displayed, the upper-left corner of the border will be at the line and position numbers specified by the LINE and POS fields. The lower-right corner of the border is located 10 lines lower than the upper border and 33 positions to the right of the left border.

The USERID field starts 2 lines lower than the upper border and 11 positions to the right of the left border character.

If RECORD1 (from the previous example) is displayed, it is placed within WINDOW1. Its fields are positioned with respect to the upper-left corner of the window. The fields from record WINDOW1 which were on the display remain because the OVERLAY keyword was used on RECORD1 and the two records do not overlap.

If RECORD2 (from the previous example) is displayed, it is also placed within WINDOW1. Its fields are positioned with respect to the upper-left corner of the window. Because the OVERLAY keyword was not used, the fields from records WINDOW1 and RECORD1 are removed from the window.

Example 3

The following example shows how to use the WINDOW keyword with a subfile.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A          R SFLDATA                   SFL
     A            NAME          20A  B  4  5
     A            RANK          10A  B  4 27
     A            SERIAL         8A  B  4 38
     A
     A          R WINDOW1                   SFLCTL(SFLDATA)
     A                                      WINDOW(8 25 10 50)
     A                                      SFLPAG(4)
     A                                      SFLSIZ(17)
     A                                      SFLDSP
     A                                      SFLDSPCTL
     A                                   2  5'Full Name'
     A                                   2 27'Rank'
     A                                   2 38'Serial Nbr'
     A
```

When the WINDOW1 subfile control record is displayed, it and the subfile are displayed in a window. The upper-left corner of the window border is at line 8, position 25 on the display. The lower-right corner of the border is located on line 19, position 78.

The fields from both the subfile record and subfile control record are located with respect to the first usable window position in the upper-left corner of the window. For example, the NAME field in the SFLDATA record starts on the 4th window line and the 5th window position, which is the same as the 12th line on the display and the 31st position on the display.

Related reference

USRRSTDSP (User Restore Display) keyword for display files
You use this record-level keyword on a window record to specify that the application will manage the display.

OVERLAY (Overlay) keyword for display files
You use this record-level keyword to specify that the record format that you are defining should appear on the display without the entire display being deleted first.

WRDWRAP (Word Wrap) keyword for display files

You use this file-level, record-level, or field-level keyword for continued-entry fields, or for named fields so that they overflow onto subsequent display lines.

The keyword causes wrapping to occur at a blank in the data rather than at the end of the data line. It is used to make long text fields easier to read. The default is for data to be wrapped at the end of the physical line or continued-entry field segment.

This keyword can only be specified on fields that have a usage of input-only (I) or input/output (B).

This keyword has no parameters.

You cannot specify the WRDWRAP keyword on the following keyboard shifts:

- Signed Numeric (S)
- Numeric Only (Y)
- Digits Only (D)
- Numeric Only Character (M)
- Floating Point (F)
- DBCS Only (J)
- DBCS Open (O)
- DBCS Either (E)
- Graphic (G)

WRDWRAP cannot be specified with the following keywords:

- AUTO(RAZ, RAB)
- CHECK(MF, M10F, M11F, RB, RZ, RL, RLTB)
- CHGINPDFT(MF)
- DSPATR(OID, SP)
- DUP
- FLTFIXDEC
- IGCALTTYP

Option indicators are not valid for this keyword.

When WRDWRAP is used, the field length is not increased. Therefore, if too much data is entered the word wrapping effect will be turned off.

Notes:

1. WRDWRAP is ignored on displays that are not attached to a controller that supports an enhanced interface for nonprogrammable workstations.
2. WRDWRAP can be specified on a field that is contained on a single row. Although wrapping will not occur, the character insert function of the field will still change.
3. Subfiles do not support WRDWRAP.

Example

The following example shows how to specify the WRDWRAP keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A          R RECORD1
     A            FIELD1       100A  O  1 17
     A            FIELD2       100A  I  4 17WRDWRAP
     A            FIELD3       100A  B  7 17WRDWRAP
     A            FIELD4       100A  B 10 17
     A            FIELD5       100A  O 13 17
     A
```

In this example, RECORD1 is defined with input, output, and both fields. FIELD2 and FIELD3 will have the benefit of word wrap when the display is attached to a controller that supports an enhanced interface for nonprogrammable workstation. FIELD4 will not have the benefits of word wrap.

Related information

Application Display Programming PDF

DDS for 3270 remote attachment

The 3270 remote attachment feature allows a 3270 Systems Network Architecture (SNA) controller or a 3274 emulating device to be attached to a IBM i model. Some applications for the 3270 that use data description specifications (DDS) might require programming changes.

An operator at a 3277, 3278, or 3279 data entry keyboard can use most features and functions as a similarly configured and authorized 5250 workstation. The attached unit has the same capabilities and limitations as any remote 5251, with the following exceptions:

- No display attributes are seen on either a 3278 or a 3279 when the field is defined as a nondisplay field.
- Numeric-only fields that are used for negative numbers act differently on the 3270 than on the 5250. On the 3270, if the operator enters a negative number in the field, the sign occupies the first position of the field followed by the number. This causes the maximum size of the field to decrease by 1 and should be considered when you design the displays and fields.
- Results that cannot be predicted can occur when data is entered in an input field that is involved in a page command, if any lines involved in the page are not of the same type field attributes and location.
- The following DDS keywords are ignored by 3270 support:

- AUTO (RA) and CHECK (ER)
- BLINK (cursor blink is controlled by keyboard for 3270)
- CHRID
- CHANGE
- LOWER or CHECK (LC) (lowercase is controlled by turning on the display for 3270)
- CHECK (RL and RLTB)
- DSPSIZ (other than 24 x 80)
- LOCK
- MSGLOC (set to row 24)

  Note: For display devices configured as a 3278 Model 4, the MSGLOC keyword is set to display messages on row 43.

- The following DDS display attributes are the only ones valid for 3270 Remote Attachment (except for 3277):

  DSPATR
  RI used for 3278 and 3279 CS used for 3278 and 3279 (changed to UL) UL used for 3278 and 3279 BL used for 3278 (determines color on 3279)

- During a write operation to the error line, the Enter key is defined as a Reset key and cannot be mapped to any other function.

  In addition, if a write operation is requested by the user application to display an error message, a read operation should immediately follow the write operation for the remote 3270 display to allow resetting of the error message. A read operation following a write operation of an error message should be used by call applications regardless of the type of target display. If a read operation does not immediately follow the write error message requested by the user application, remote 3270 displays might overlay the error message before it can be read by the user.

System/36 environment considerations for display files

The User Display Management (USRDSPMGT) keyword causes the record formats in the display file to function similarly to System/36 SFGR display formats.

The SFGR-to-DDS conversion utility always generates the USRDSPMGT keyword. If you are defining a DDS display file to be used in the System/36 environment, you should specify the USRDSPMGT keyword.

For a description of operational differences, see the Application Display Programming book.

Related reference

USRDSPMGT (User Display Management) keyword for display files
You use this file-level keyword to specify that all data written to the display is held until the data is overwritten or cleared by using the CLRL keyword.

Keyword considerations for display files used in the System/36 environment

You cannot specify some keywords in display files that contain the USRDSPMGT keyword.

You cannot specify the following keywords in this kind of display file:

ASSUME
ERASE
ERRSFL
HLPCMDKEY
IGCCNV
KEEP
MNUBAR
PULLDOWN
PUTRETAIN
SFL
SFLCTL
SNGCHCFLD

In a file containing the USRDSPMGT keyword, the OVERLAY keyword is ignored.

The following keywords have a response indicator parameter. Because System/36 environment applications do not support response indicators, you should not specify a response indicator on any of these keywords in a file to be used in the System/36 environment. If you specify a response indicator for any of these keywords in a file that contains the USRDSPMGT keyword, a warning message is issued.

BLANKS	ERRMSGID	PRINT
CAnn	HELP	ROLLDOWN
CFnn	HLPRTN	ROLLUP
CHANGE	HOME	SETOF
CLEAR	PAGEDOWN	SETOFF
DUP	PAGEUP	VLDCMDKEY
ERRMSG		

If you specify a response indicator on the HELP keyword in a file containing the USRDSPMGT keyword, an error message is issued.

Related tasks

Defining a display file for DDS

When you specify positional entries for display files, you need to follow some specific rules for filling in positions 1 through 44 of the data description specifications (DDS) form.

ALTNAME (Alternative Record Name) keyword

You use this record-level keyword to specify an alternative name for a record.

The alternative name can be specified for I/O operations when using program-described files. The syntax of the alternative record name must be valid for the high-level language compiler in use.

The format of the keyword is:

```
ALTNAME('alternative-name')
```

The length of the alternative name is 1 to 8 characters. The first character of the name must not be an asterisk.

The alternative name must be different from all other names and from all DDS record names (positions 19 through 28) in the file. When a duplicate name is found, an error is issued on the record name or alternative record name.

ALTNAME is not allowed on subfile records (SFL keyword).

Option indicators are not valid for this keyword.

The following example shows how to specify the ALTNAME keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A          R RECORD1                  ALTNAME('R( 2).a')
     A
```

The alternative name for RECORD1 is 'R(2).a'.

CHANGE record-level keyword

You use this record-level keyword to indicate that on an input operation, the record is to be returned to the application program only if the user has changed the record.

If the user enters data into any input-capable field, all of the input-capable fields in the record will be returned. If the user does not enter data into any field, the input-capable fields will be returned initialized by the compiler.

This keyword has no parameters.

This format of the CHANGE keyword is allowed only in files containing the USRDSPMGT keyword.

Option indicators are not allowed with this keyword.

For files that are used by the IBM i operating system applications (with or without the USRDSPMGT keyword), use the format of the CHANGE keyword.

Related reference

CHANGE (Change) keyword for display files
You use this record-level or field-level keyword to set on the specified response indicator for an input operation.

HELP and HLPRTN keyword

In a file with the USRDSPMGT keyword, the HELP keyword alone will not return control to the application program. HLPRTN must be specified to return control to the application program.

If you specify a response indicator on the HELP keyword in a file containing the USRDSPMGT keyword, an error message is issued.

MSGID keyword

You use this field-level keyword to allow an application program to identify, at program run time, the message description that contains text for a named field.

You can specify the MSGID keyword in either of the following formats:

```
MSGID(message-identifier [library-name/]message-file)
MSGID(*NONE)
```

You can specify the message-file parameter in one of the following forms:

- `&field3`

  where the field3 length is two (2).

  The field name must exist in the same record as the MSGID field, and the field must be defined as a character field with usage H, P, B, or O.

  You should, for this form only, specify only special values for the file parameter. You cannot specify a library.

  The special values are: U1, U2, P1, P2, M1, and M2. If the specified value is not one of these special values, U1 is used. See Table 15 on page 249 for more information about these values.
- Special values for message-file:
  - *USR1
  - *USR2
  - *PGM1
  - *PGM2
  - *SYS1
  - *SYS2

  When you code a special value for the message-file, the library parameter is not allowed and the library is set to *LIBL as default. See Table 15 on page 249 for more information about these special values.

The following table describes the meaning of the special file values for the MSGID keyword.

Table 15. Special values on MSGID keyword

DDS special value	Length 2 field value	Message text retrieved
*USR1	U1	First level text from message file USR1

Table 15. Special values on MSGID keyword (continued)		
DDS special value	**Length 2 field value**	**Message text retrieved**
*USR2	U2	Second level text from message file USR2
*PGM1	P1	First level text from message file PGM1
*PGM2	P2	Second level text from message file PGM2
*SYS1	M1	First level text from message file SYS1
*SYS2	M2	Second level text from message file SYS2

For information about using message retrieval, see the Application Display Programming book  .

The *NONE parameter indicates that no message text is displayed.

The following example shows how to specify the MSGID keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A          R RECORD1
     A            MSGFIELD1     40A  B 02 10MSGID(CPD0001 *USR1)
     A            MSGFIELD2     10A  O 02 60MSGID(&MSGIDNUM &MSGFILENM);
     A            MSGFIELD3     80A  B 02 60
     A  99                                  MSGID(USR &MSGNBR +
     A                                      &MSGFILENM);
     A                                      MSGID(*NONE)
     A            MSGIDNUM       7A  P      TEXT('Message id')
     A            MSGFILENM      2A  P      TEXT('Message file name')
     A            MSGNBR         4A  P 07 01TEXT('Message number')
     A
```

When RECORD1 is displayed:

- MSGFIELD1 contains the first 40 characters of the message CPD0001 from the message file USR1. Because the field is input/output (usage B), the value of the field can be changed by the user.
- MSGFIELD2 contains the first 10 characters of the message identified by the fields MSGIDNUM and MSGFILENM. Values for MSGIDNUM (the message identifier) and MSGFILENM (the message file) must be set in the program before the display of RECORD1. Because MSGFIELD2 is an output-only field (usage O), it cannot be used in the program.
- If option indicator 99 is on, MSGFIELD3 contains the first 80 characters of the message identified by the prefix USR, the message number set in field MSGNBR, and the message file set in field MSGFILENM. If option indicator 99 is off, MSGFIELD3 does not contain any message text.

Related reference

MSGID (Message Identifier) keyword for display files

You use this field-level keyword to allow an application program to identify, at program run time, the message description that contains text for a named field.

PRINT(*PGM) keyword

How the PRINT key is handled for a display file with the PRINT(*PGM) and USRDSPMGT keywords specified is determined by how the program that is reading data from the display file is compiled and coded.

If the program is compiled with a System/36-compatible compiler (RPGII or COBOL) and the program is coded to handle the Print key exception, the program is given control when the Print key is pressed. If the program is coded not to handle the Print key exception, the screen image is printed.

If the program is compiled with an IBM i compiler (RPG III, RPG IV, or COBOL), the program is always given control when the Print key is pressed. If the program does not handle the Print key exception, it acts as if the Enter key was pressed.

RETKEY (Retain Function Keys) and RETCMDKEY (Retain Command Keys) keywords

You use these record-level keywords to indicate that function keys, command function (CF*nn*) keys, or command attention (CA*nn*) keys, which are enabled on a display, should be retained when the record you are defining is displayed.

In most cases, the keys enabled on the display are those which were specified on the last output operation. Additionally, the IBM i operating system automatically retains valid keys when a record format sends no data to the display.

Note that if the record previously displayed is defined in another display file, the keys enabled for that record will not be retained when the record you are defining is displayed.

These keywords have no parameters.

Related reference

CAnn (Command Attention) keyword for display files
You use this file-level or record-level keyword to specify that the function key specified in the keyword (CA01 through CA24) is available for use.

CFnn (Command Function) keyword for display files
You use this file-level or record-level keyword to specify that the function key specified in the keyword (CF01 through CF24) is available for use.

RETKEY keyword

RETKEY indicates to retain any CLEAR, HELP, HLPRTN, HOME, PAGEDOWN, PAGEUP, PRINT, ROLLDOWN, and ROLLUP keywords when the record is displayed.

You cannot specify RETKEY with a CLEAR, HELP, HOME, PAGEUP, PAGEDOWN, ROLLDOWN, or ROLLUP keyword on the file level or on this record format. PRINT is not allowed on the same record format with RETKEY.

The HLPRTN and PRINT keywords are allowed at the file level with RETKEY. If option indicators are specified on either HLPRTN or PRINT, the state of the indicators when the current record is displayed determines whether the keyword is active.

If you specify HLPRTN on the same record format with RETKEY, the HLPRTN function will not be retained from the previous record.

RETCMDKEY keyword

RETCMDKEY indicates whether to retain CA*nn* or CF*nn* keys when the record is displayed.

You cannot specify the CA*nn* or CF*nn* keywords with the RETCMDKEY on the file level or on this record format. You cannot specify any CA*nn*, CF*nn*, SFLDROP, SFLENTER, or SFLFOLD keywords on the record being defined.

Considerations for specifying RETKEY and RETCMDKEY keywords

These rules must be considered when you specify the RETKEY and RETCMDKEY keywords.

- The file must specify a separate indicator area (INDARA keyword).
- RETKEY and RETCMDKEY are ignored on the first output operation after a file is opened. The retain function is valid only between record formats in the same display file.
- The response indicator of the VLDCMDKEY keyword is set by the IBM i program according to the current valid command keys, including the keys retained when you specify the RETKEY and RETCMDKEY keywords.

- Neither keyword is allowed on a subfile format (SFL keyword) or on a user-defined record (USRDFN keyword).
- You cannot specify either RETKEY or RETCMDKEY in a file that contains the ALTHELP, ALTPAGEUP, or ALTPAGEDWN keyword.

Option indicators are not valid for these keywords.

Example

The following example shows how to specify the RETKEY and RETCMDKEY keywords.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A                                      INDARA
     A          R RECORD1                   CF01(01 'EXIT SCREEN')
     A                                      CF02(02 'SET ON IN02')
     A                                      ROLLUP(03)
     A  08                                  CLEAR(03 'CLEAR KEY')
     A                                   1  3'COMPANY NAME'
     A                                   1 25'CF01 TO EXIT'
     A          R RECORD2                   RETKEY
     A                                      RETCMDKEY
     A                                      OVERLAY
     A            FIELD1         4A  B  5  5
     A          R RECORD3                   RETKEY
     A                                      CF01(90 'ALTERNATE CF01')
     A            FIELD1        10A  B  7  5
     A
```

The records are displayed in the following order: RECORD1, RECORD2, RECORD3.

When RECORD1 is displayed, CF01, CF02, Clear, and Page Up keys are activated. The same keys are valid for RECORD2, because RETKEY and RETCMDKEY are specified. Because RECORD3 specifies RETKEY, the Clear and Page Up keys are valid. CF01 has been redefined for this record. CF02 is not valid for this record.

Note: The retain function does not require the record format to be displayed. (RECORD3 uses function keys defined in RECORD1, but because no OVERLAY keyword is specified in RECORD3, the display is erased before RECORD3 is displayed.)

USRDSPMGT (User Display Management) keyword

You use this file-level keyword to indicate that this display file should be processed with System/36 environment functions.

This keyword has no parameters.

You cannot use USRDSPMGT in display files containing any of the following keywords:

ASSUME
ERASE
HLPCMDKEY
IGCCNV
KEEP
PUTRETAIN
SFL
SFLCTL

In a file containing the USRDSPMGT keyword, the OVERLAY keyword is ignored.

Option indicators are not valid for this keyword.

The following example shows how to specify the USRDSPMGT keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A                                      USRDSPMGT
     A          R RECORD
     A            FIELD1
     A
```

Unicode considerations for display files

Unicode is a universal encoding scheme for written characters and text that enables the exchange of data internationally. Two transformation formats, UTF_16 and UCS_2, of Unicode are supported with DDS.

A Unicode field in a display file can contain UCS-2 or UTF-16 data. Unicode data is composed of *code units*, which represent the minimal byte combination that can represent a unit of text.

There are two transformation formats (encoding forms) of Unicode that are supported with DDS:

- **UTF-16** is a 16-bit encoding form designed to provide code values for over a million characters and a superset of Unicode. UTF-16 data is stored in graphic data types. The CCSID value for data in UTF-16 format is 1200.

  A UTF-16 code unit is 2 bytes in length. A UTF-16 character can be 1 or 2 code units (2 or 4 bytes) in length. A UTF-16 data string can contain any character including UTF-16 surrogates and combining characters.
- **UCS-2** is the Universal Character Set coded in 2 octets, which means that characters are represented in 16 bits per character. One code unit is used in this topic to describe the size of a UCS-2 character. UCS-2 data is stored in graphic data types. The CCSID value for data in UCS-2 format is 13488.

  UCS-2 is a subset of UTF-16 and can no longer support all of the characters defined by Unicode. UCS-2 is identical to UTF-16 except that UTF-16 also supports the combining of characters and surrogates. If you do not need support for the combining of characters and surrogates, you can choose to continue to use the UCS-2 format.

Unicode data is not supported on display devices that currently support the 5250 data stream. Therefore, conversions between the Unicode data and EBCDIC are necessary during input and output. On output, the Unicode data is converted to the CCSID of the device. On input, the data is converted from the device CCSID to the Unicode CCSID.

Because the device CCSID, which is determined from the device configuration, determines what the Unicode data is converted to, the converted data can appear differently on different devices. For example, a Unicode code unit that maps to an SBCS character is displayed as a DBCS replacement character on a graphic-DBCS-capable device. On a DBCS-capable or an SBCS-capable device, the code unit appears as an SBCS character. A Unicode code unit that maps to a DBCS character is displayed as a graphic-DBCS character on a graphic-DBCS-capable device. On a DBCS device, a DBCS character is displayed and bracketed (enclosed in shift-out and shift-in characters). An SBCS replacement character is displayed on an SBCS device.

It is also suggested that all fields that are capable of Unicode are initialized in the output buffer before writing the fields to the screen. Unpredictable results might occur if default initialization is allowed to take place.

Positional entry considerations for display files that use Unicode data

Be aware of these positional entry considerations for display files that use Unicode data. Positions not mentioned have no special considerations for Unicode.

Length (positions 30 through 34)

Specify the length of the field in these positions. The length of a field containing Unicode data can range from 1 through 16 381 code units.

When determining the program length of a field containing Unicode data, consider the following rules:

- Each Unicode code unit is 2 bytes long.
- The program length of the field is specified in number of Unicode code units. For example, a field containing 3 Unicode code units has 6 bytes of data.
- The field's default display length is equal to the field's program length or 2 times the number of Unicode code units.

- After converting between Unicode data and EBCDIC, the resulting data can be equal to, longer or shorter than the original length data before the conversion, depending upon the CCSID of the device. For example, 1 Unicode code unit is composed of 2 bytes of data. That code unit might convert to 1 SBCS character composed of 1 byte of data, 1 graphic-DBCS character composed of 2 bytes of data, or 1 bracketed DBCS character composed of 4 bytes of data.
- The field's display length can be specified separately from the program length by using the alternate-field-length parm on the CCSID keyword.

Data type (position 35)

The only valid data type for Unicode data is the G data type.

G (Graphic)
Type G in combination with the CCSID keyword to specify this field contains Unicode data.

Normally, by specifying G, the field contains graphic-DBCS data. In combination with the CCSID keyword, the field now contains Unicode data. On output, the data is mapped to corresponding characters in the CCSID that the device is configured as. On input, the data is mapped to corresponding Unicode code units.

Decimal positions (positions 36 and 37)

Leave these positions blank when using Unicode data.

Related reference
Valid entries for display files
These entries are valid for display files.

Keyword considerations for display files that use Unicode data (positions 45 through 80)

The DFT keyword can contain SBCS, bracketed-DBCS, or bracketed-DBCS-graphic character strings when specified on a Unicode-capable field.

No validity checking is allowed for a field capable of Unicode.

The CCSID keyword specifies that a G-type field supports Unicode data instead of DBCS-graphical data.

CCSID (Coded Character Set Identifier) keyword

You use this file-, record-, or field-level keyword to specify that a G-type field supports Unicode data instead of DBCS-graphical data. Like DBCS-graphic characters, Unicode code units are two bytes long.

The format of the keyword is:

```
CCSID(Unicode-CCSID | &Unicode-CCSID-field | *REFC
      [field-display-length | *MIN | *LEN display-positions])
```

The Unicode-CCSID parameter is required. Use the Unicode-CCSID parameter to specify a CCSID that uses a Unicode encoding scheme for this field. The Unicode-CCSID parameter can be specified either as a number up to 5 digits long or as a program-to-system field. You must define the program-to-system field with a length of 5 and with the S data type.

A special value of *REFC can be specified instead of a Unicode-CCSID value. It is valid only on reference fields, and the referenced field must be coded with a CCSID keyword specifying a Unicode-CCSID value. Normally, the display file CCSID keyword will override any CCSID keyword attributes taken from the referenced field. If *REFC is specified, the Unicode-CCSID value is taken from the referenced field.

The field-display-length parameter is optional and is valid only when the CCSID keyword you specify it at the field level. You specify the field-display-length as the number of Unicode code units.

When Unicode data is involved in an output operation, the data is converted from the associated Unicode CCSID to the CCSID of the device. Generally, the length of the data will change when this conversion occurs. Therefore, you can use the field-display-length value to specify a displayed field length that is different from the default displayed field length. The default displayed field length of a 'G' data type field is twice the number of code units that are specified for the field length.

The field-display-length value can help avoid truncation of field data when the data length will be longer after conversion than the default displayed field length. The field-display-length value can also help increase the available line space by limiting the displayed field length when the data length will be shorter after conversion. The field length will still be used to define the field's output buffer length.

For example, a display file contains the following line:

```
FLD1         10G  B  2  2 CCSID(X Y)
```

- X is the Unicode-CCSID associated with the field data. Y is the field-display-length of this field. If you do not specify Y, then FLD1's length on the screen is 20 single-byte characters (twice the number of code units specified on the field length).
- If you know that the Unicode data is constructed from single byte data, then you can specify the field-display-length, Y, as 5 Unicode code units. FLD1 will have a length of 10 single byte characters on the screen (twice the number of code units that are specified on the field-display-length).
- If you know that the Unicode data is constructed from double-byte data, then you can specify the field-display-length, Y, as 11 Unicode code units. FLD1 will have a length of 22 single byte characters on the screen. This will allow space for the shift-out and shift-in characters.

A special value, *MIN, can be specified instead of a field-display-length. This value will be used to specify a field length defined in terms of display positions. This value causes the field length on the screen to be equal to the number of Unicode code units defined in the DDS.

A special value, *LEN, along with a display-positions value can be specified instead of a field-display-length. This value is used to specify a field length defined in terms of display positions. This value causes the field length on the screen to be equal to the display-positions value.

If the CCSID keyword is specified at both the field-level and the record- or file-level, the field-level keyword takes precedence.

On output, field data that is longer than the specified field length is truncated. On input, if too many characters were entered into the Unicode field, then the field is reverse imaged and an error appears on the error line stating too many characters were entered. You need to press reset and correct the field. The maximum number of characters to enter is conveyed in the error message.

The CCSID keyword can be specified with all of the following field level keywords:

ALIAS	DFTVAL	ERRMSG
AUTO(RA)	DSPATR(BL)	ERRMSGID
BLANKS	DSPATR(CS)	FLDCSRPRG
CHANGE	DSPATR(HI)	INDTXT
COLOR	DSPATR(MDT)	OVRATR
DFT	DSPATR(ND)	OVRDTA
DLTCHK	DSPATR(PC)	PUTRETAIN
DSPRL	DSPATR(PR)	REFFLD
DUP	DSPATR(RI)	SFLCSRPRG
CHECK(FE)	DSPATR(UL)	TEXT
CHGINPDFT	ENTFLDATR	

Option indicators are not valid for this keyword.

Example

The following example shows how to specify the CCSID keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A                                     CCSID(13488)
00020A          R RECORD1
00030A            FIELD1        30G
00040A            FIELD2        10G         CCSID(61952 6)
00050A          R RECORD2                   CCSID(1200)
00060A            FIELD3        20G
00070A          R RECORD3
00080A            FIELD4        10G         CCSID(61952 *MIN)
00090A            FIELD6        10G         CCSID(1200 *LEN 8)
```

FIELD1 is assigned a CCSID value of 13488. FIELD2 is assigned a CCSID value of 61952 and has a field length of 6 Unicode code units long (12 SBCS characters). FIELD3 is assigned a CCSID value of 1200. FIELD4 will use 10 display positions on the screen. FIELD6 will use 8 display positions on the screen.

Double-byte character set considerations for DDS

Be aware of these double-byte character set (DBCS) considerations for the positional entries and keyword entries for display files, as well as general considerations for using DBCS data in display files.

The functions described in these topics are supported on both DBCS and non-DBCS systems.

Related concepts

General considerations for using DBCS text with DDS files

Positional entry considerations for display files that use DBCS

When you create display files that use double-byte character set (DBCS) characters, be aware of DBCS considerations for the length, data type, and decimal positional entries on display files.

Positions that are not mentioned have no special considerations for DBCS.

Length (positions 30 through 34)

The length of a field containing bracketed-DBCS data can range from 4 through 32 763 bytes. The length of a DBCS-graphic field can range from 1 through 16 381 characters.

When determining the length of a DBCS field, consider the following rules:

- Each DBCS character is 2 bytes long.
- For DBCS-graphic fields, the length of the field is specified in number of DBCS characters.
- Include both shift-control characters in the length of the field for fields with a data type of J, E, or O. Together, these characters are 2 bytes long.
- Fields specified with the J or E data type or keyboard shift must have an even length.

For example, a bracketed-DBCS field that contains up to 3 DBCS characters, 1 shift-in character, and 1 shift-out character, has 8 bytes of data:

```
(3 characters x 2 bytes) + (shift-out + shift-in) = 8
```

A DBCS-graphic field that contains up to 3 DBCS characters has 6 bytes of data:

```
(3 characters x 2 bytes) = 6
```

Data type (position 35)

You can specify the data type in this position by typing J, E, O, G.

J (Only)

Type J to specify this field as a DBCS-only field.

The display station automatically inserts shift-control characters in fields specified with this data type.

If you specify J, you must specify an even number for the field length (positions 30 through 34).

E (Either)
Type E to specify this field as a DBCS-either field.

You can type either DBCS or alphanumeric characters in the field. The type of data typed in the first position of the field determines the type of data that can be typed in the remainder of the field. If the field is empty, the system assumes that alphanumeric data will be typed in. To change the field so DBCS data can be typed, position the cursor in the field and put the keyboard in DBCS mode.

If the field contains DBCS data, the display station automatically inserts shift-control characters.

If you specify E, you must specify an even number for the field length (positions 30 through 34).

O (Open)
Type O to specify this field as a DBCS-open field. You can type both DBCS and alphanumeric characters in this field. Use shift-control characters to distinguish DBCS data from alphanumeric data.

If the field contains DBCS data, the system does not ensure that the data is enclosed between shift-control characters.

If you specify O, you can specify either an even or an odd number for the field length (positions 30 through 34).

G (Graphic)
Type G to specify this field as a DBCS-graphic field. Data typed in this field does not contain shift-control characters.

If you specify G, you must specify the number of DBCS characters for the field length (positions 30 through 34).

Related reference
Valid entries for display files
These entries are valid for display files.

Decimal positions (positions 36 and 37)

Leave these positions blank when using DBCS data.

Keyword considerations for display files that use DBCS

Some DDS keywords should be avoided in double-byte character set (DBCS) data fields, and others should be used with caution.

Do not use the following DDS keywords with DBCS data fields (the data type specified in position 35 is J, E, O, or G):

AUTO(RAZ)	CHECK(VN)	FLTFIXDEC
BLKFOLD	CHECK(VNE)	FLTPCN
CHECK(M10)	CHRID	MSGCON
CHECK(M10F)	DATE	REFSHIFT
CHECK(M11)	DLTEDT	SFLMSGKEY
CHECK(M11F)	DSPATR(OID)	SFLPGMQ
CHECK(RL)	DSPATR(SP)	SFLRCDNBR
CHECK(RLTB)	EDTCDE	SFLROLVAL
CHECK(RZ)	EDTWRD	TIME

Do not use the CHECK(LC) and LOWER keywords on DBCS-only fields (J specified in position 35).

Do not use the IGCALTTYP, IGCANKCNV, CHECK(LC), and LOWER keywords on DBCS-graphic fields (G specified in position 35).

The following DDS keywords can be used in files containing DBCS data only when the function indicated by the keyword is available on the display device or with the type of data used. However, DDS does not apply record- and file-level keywords to DBCS fields.

CHECK(RL)	DSPSIZ(*DS4)	ERASEINP(*ALL)
CHECK(RLTB)	DSPATR(SP)	MDTOFF(*ALL)
COLOR	DSPMOD	

Related reference

DDS keyword entries for display files (positions 45 through 80)

You type the keyword entries that define display files in positions 45 through 80 (functions).

CNTFLD (Continued-Entry Field) keyword

You use this field-level keyword to define a field as a continued-entry field.

Continued-entry fields are sets of associated entry fields that are treated by the workstation controller as a single field during field-data entry and editing. If the display device is not attached to a controller that supports an enhanced interface for nonprogrammable workstations, each segment of the continued entry field is treated separately when editing is performed on the field.

The format of the keyword is:

```
CNTFLD(width of column)
```

One parameter must be specified.

DBCS considerations

DBCS data types have the following restrictions:

J
The width of each continued-entry field segment must be an even number of at least 4 bytes.

E
The width of each continued-entry field segment must be an even number of at least 4 bytes.

O
The width of each continued-entry field segment must be at least 4 bytes wide.

G
The width of each continued-entry field segment must be an even number of at least 4 bytes.

Special consideration must be taken when defining the length of the DBCS continued-entry field to account for the SO/SI character pairs that must bracket the DBCS data on each segment of the continued-entry field. The following total field lengths are required to ensure that the field data fits into DBCS continued-entry fields:

J or E (with DBCS data)
Data Length + (Number of segments - 1) * 2

O
Data Length + (Number of segments - 1) * 3

G or E (with SBCS data)
Data Length

Note: The (Number of segments - 1) * 2 portion of the calculation in the first equation allows for the SO/SI sets that must bracket the DBCS data on the segments of the continued-entry field after the first segment.

The (Number of segments - 1) * 3 portion of the calculation in the second equation allows for the SO/SI sets that must bracket the DBCS data on the segments of the continued-entry field after the first segment. Additional consideration is made for the possibility that a NULL must be placed at the end of a segment wherever a DBCS character is split.

Note: WRDWRAP cannot be used on DBCS continued-entry fields.

GRDATR (Grid Attribute) keyword

You use this file-level or record-level keyword to define the default color and line type attributes for the grid structure.

The format of the keyword is:

```
GRDATR([(*COLOR grid-line-color | &Color-field)]
[(*LINTYP grid-line-attribute | &Lintype-field)])
```

P-fields can be used to define or change the attributes at run time when this keyword is used at the record-level.

Valid parameter and p-fields values are:

Table 16. Valid color values

COLOR	Meaning	Program field value
BLU	Blue	X'01'
GRN	Green	X'02'
CYAN	Cyan	X'03'
RED	Red	X'04'
VLT	Violet	X'05'
YLW	Yellow	X'06'
WHT	White	X'07'
GRY	Gray	X'08'
LBLU	Light blue	X'09'
LGRN	Light green	X'0A'
LTRQ	Light Turquoise	X'0B'
LRED	Light red	X'0C'
LVLT	Light violet	X'0D'
LYLW	Light yellow	X'0E'
HWHT	High-intensity white	X'0F'
BLK	Black	X'10'
NONE	Default value of the display	X'FF'

Note: The default color is white.

Table 17. Valid line types

Line type	Meaning	Program field value
SLD	Solid	X'00'
THK	Thick	X'01'
DBL	Double	X'02'
DOT	Dot	X'03'
DSH	Dash	X'08'
THKDSH	Thick dash	X'09'

Table 17. Valid line types (continued)		
Line type	**Meaning**	**Program field value**
DBLDSH	Double dash	X'0A'
NONE	Default value of the display	X'FF'

Note: The default line type is solid.

If a p-field is specified for either the COLOR or LINTYP parameter, the field must exist in the record format. The field is defined as data type A, usage P, and length of 1.

Grid line support requires DBCS equipment. This equipment should have the capability of calling Japanese DOS.

Option indicators are valid for this keyword.

Example

The following example shows how to specify the GRDATR keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A                                  GRDATR((*COLOR WHT) (LINTYP SLD))
     A          R GRDREC1               GRDRCD
     A                                  GRDATR((*COLOR BLU) (LINTYP DSH))
     A                                  GRDBOX((*POS (2 2 10 70 )) +
     A                                  (*TYPE PLAIN)
     A
     A          R GRDREC2               GRDRCD
     A                                  GRDBOX((*POS (4 4 5 45)) +
     A                                  (*TYPE PLAIN)
     A                                  GRDLIN((*POS (6 4 20)) +
     A                                  (*TYPE LOWER) +
     A                                  (*COLOR RED) (*LINTYP DBL))
     A
```

When the GRDREC1 record is written, the TYPE PLAIN box defined by the GRDBOX keyword within the GRDREC2 record will be displayed with a blue dash lines. These attributes are defined on the GRDATR keyword on the GRDREC1 record.

When the GRDREC2 record is written, the TYPE PLAIN box defined by the GRDBOX keyword within the GRDREC2 record will be displayed with a white solid line. These attributes are defined on the GRDATR keyword at the file level. The GRDLIN defined within GRDREC2 will be a red double line. The attributes defined on the GRDBOX or GRDLIN keyword override any GRDATR keyword on the file- or record-level.

GRDBOX (Grid Box) keyword

You use this record-level keyword to define the shape, positioning, and attributes for the box structure. This keyword defines whether the box is erased, displayed, or not processed.

The format of the keyword is:

```
GRDBOX((*POS ([*DS3] [*DS4]
start-row | &start-row-field
start-column | &start-column-field
depth | &depth-field
width | &width-field));
[(*TYPE type of box
[horizontal rule | &hrule-field]
[vertical rule | &vrule-field])]
[(*COLOR color of box | &color-field)]
[(*LINTYP line type of box | &lintyp-field)]
[(*CONTROL | &control-field)]
```

The *POS parameter is a required parameter. This parameter describes the position and size of the box. When coding *DS3 or *DS4 within the *POS parameter, you can have 2 different start row, start column, and length values depending on the display size being used. DSPSIZ keyword must be coded on the file level.

The *TYPE parameter is a required parameter. The horizontal and vertical rule values define the number of character spaces between each rule. For example, if a *TYPE VRT box is defined with a width of 21 columns and a rule value of 3 columns, there will be 6 vertical lines within the box. If the rule value is not an even multiple of the width or depth, the odd space rule will occur at the right side or the bottom of the box. The default for this parameter is PLAIN.

The horizontal or vertical rule values can be defined using program-to-system fields. If a field name is specified, the field must exist in the record format. The field is defined as a data type S, usage P, field length of 3, and zero decimal positions.

The *COLOR and *LINTYP parameters define the color and attributes of the box. P-fields can be used to define or change the attributes at run time.

For more information about the *COLOR and *LINTYP parameters, see “GRDATR (Grid Attribute) keyword” on page 259.

If *NONE is defined by the GRDBOX keyword, the color set by the GRDATR keyword will be used.

If a p-field is specified for either the COLOR or LINTYP parameter, the field must exist in the record format. The field is defined as data type A, usage P, and length of 1.

The *CONTROL parameter specifies whether this GRDBOX is to be displayed, erased from the screen, or ignored (similar to optioning off the keyword). The field must exist in the record format and must be defined as data type S, usage P, and length of 1. If the p-field is set to 0, the grid line will be displayed. If the p-field is set to 1, the GRDBOX keyword will not be processed. If the p-field is set to -1, the grid line record currently shown will be cleared. If the p-field is set to something other than the defined values, then the default 0 will be used.

Grid line support requires DBCS equipment. This equipment should have the capability of calling Japanese DOS.

Option indicators are valid for this keyword.

Example

The following example shows how to specify the GRDBOX keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A                                     DSPSIZ(*DS3 *DS4)
     A                                     GDRATR((*COLOR WHT) (*LINTYP SLD))
     A          R GRDREC1                  GRDRCD
     A                                     GRDATR((*COLOR BLU) (LINTYP DSH))
     A                                     GRDBOX(*POS (2 5 10 70 ) (*TYPE PLAIN))
     A
     A          R GRDREC2                  GRDRCD
     A 90                                  DSPMOD(*DS4)
     A                                     GRDBOX((*POS (*DS3 5 5 18 70) +
     A                                     (*DS4 5 5 19 120)) (*TYPE PLAIN))
     A                                     GRDBOX((*POS (*DS3 5 5 18 70) +
     A                                     (*DS4 7 7 3 103)) (*TYPE VTR 10) +
     A                                     (*CONTROL &CNTL1));
     A
     A                                     GRDBOX((*POS (*DS3 12 7 6 53) +
     A                                     (*DS4 127 6 103)) +
     A                                     (*TYPE HRZ 2) +
     A                                     (*COLOR RED)  (*LINTYP &LNTP1); +
     A                                     (*CONTROL &CNTL2));
     A
     A
     A 95                                  GRDBOX((*POS (&SCROW1 &SCOL1 &DPTH1 +
     A                                     &WDTH1)); +
     A                                     (*TYPE HRZVRT &HRUL1 &VRUL1); +
     A                                     (*COLOR &CLR1); +
     A                                     (*CONTROL &CNTL3));
     A            CNTL1          1S 0P
     A            CNTL2          1S 0P
     A            CNTL3          1S 0P
     A            LNTP1          1A  P
     A            CLR1           1A  P
     A            SROW1          3S 0P
     A            SCOL1          3S 0P
     A            DPTH1          3S 0P
```

```
      A            WDTH1          3S 0P
      A            HRUL1          3S 0P
      A            VRUL1          3S 0P
      A
```

When the GRDREC1 record is written, the plain box defined at position row 2, column 4, depth of 10 rows, and width of 70 columns will be displayed. The box will have a color of blue with dash lines.

When the GRDREC2 record is written, the following output appears:

- If the record is written to a 24-by-80 display or DSPMOD is turned off, then:
  1. A plain box displays starting at row 5, column 5, depth of 18 rows, and width of 70 columns. The lines of the grid are white in color and have a solid line type defined by the file-level GRDATR keyword.
  2. If the value in the p-field CNTL1 equals 0, a vertical ruled box is drawn starting at row 7, column 7, depth of 3 rows, and width of 70 columns. The box has a vertical line every 10 character spaces. The lines of the grid are white and have a solid line type defined by the file-level GRDATR keyword. If the p-field CNTL1 value is -1, the box is erased. If the p-field CNTL1 value is 1, no action is taken by the GRDBOX keyword.
  3. If the value in the p-field CNTL2 equals 0, a horizontal ruled box displays. The box starts at row 12, column 7, depth of 6 rows, and width of 60 columns. The box has a horizontal line every 2 character spaces. The lines are red and the line type depends on the value in the p-field LNTP1. If the value in LNTP1 is not valid or is NONE (X'FF'), the line type is set to the line type from the file-level GDRATR keyword (solid) as default. If the p-field CNTL2 value is a -1, the box is erased. If the p-field CNTL2 value is 1, no action is taken by the GRDBOX keyword.
  4. If the option indicator 95 is turned on and the value in p-field CNTL3 equals 0, the horizontal and vertical ruled box is processed. The row, column, width, and depth are determined at run time from the appropriate p-fields. The color is determined from the p-field value in CLR1. The line type is set to the GRDATR keyword at the file-level as default. If option indicator 95 is turned off, the box is not be processed. If the p-field CNTL3 value is a -1, no action is taken by the GRDBOX keyword.
- If record is written to a 27-by-132 display and DSPMOD is turned on, then:
  1. A plain box displays starting at row 5, column 5, depth of 19 rows, and width of 120 columns. The lines of the grid are white in color and have a solid line type defined by the file-level GRDATR keyword.
  2. If the value in the p-field CNTL1 equals 0, a vertical ruled box is drawn starting at row 7, column 7, depth of 3 rows, and width of 110 columns. The box has a vertical line every 10 character spaces. The lines of the grid are white and have a solid line If the p-field CNTL1 value is a -1, the box is erased. If the p-field CNTL1 value is a 1, no action is taken by the GRDBOX keyword.
  3. If the value in the p-field CNTL2 equals 0, a horizontal ruled box displays. The box starts at row 12, column 7, depth of 6 rows, and width of 110 columns. The box has a horizontal line every 2 character spaces. The lines are red and the line type depends on the value in the p-field LNTP1. If the value in LNTP1 is not valid or is NONE (X'FF'), the line type is set to the line type from the file-level GDRATR keyword (solid) as default. If the p-field CNTL2 value is a -1, the box is erased. If the p-field CNTL2 value is 1, no action is taken by the GRDBOX keyword.
  4. If the option indicator 95 is turned on and the value in p-field CNTL3 equals 0, the horizontal and vertical ruled box is processed. The row, column, width, and depth are determined at run time from the appropriate p-fields. The color is determined from the p-field value in CLR1. The line type is set to the GRDATR keyword at the file-level as default. If option indicator 95 is turned off, the box is not processed. If the p-field CNTL3 value is a -1, no action is taken by the GRDBOX keyword.

GRDCLR (Grid Clear) keyword

You use this record-level keyword to define the rectangle on a screen in which all grid structures are cleared.

The format of the keyword is:

```
GRDCLR[(*POS ([*DS3][*DS4]
start row | &start-row-field
start column | &start-column-field
depth | &depth-field
width | &width-field))]
```

If no parameters are defined, GRDCLR keyword will clear all grid lines.

The *POS parameter is an optional parameter. This parameter display size conditioning on the GRDCLR keyword. When coding *DS3 or *DS4 within the *POS parameter, you can have 2 different start row, start column, and length values depending on the display size being used. DSPSIZ keyword must be coded on the file level.

If a field name is specified, the field must exist in the record format, data type S, usage P, length of 3, and zero decimal positions.

Grid line support requires DBCS equipment. This equipment should have the capability of calling Japanese DOS.

Option indicators are valid for this keyword.

Example

The following example shows how to specify the GRDCLR keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A                                      DSPSIZ(*DS3 *DS4)
     A
     A          R GRDREC1                   GRDRCD
     A                                      GRDCLR
     A
     A          R GRDREC2                   GRDRCD
     A  90                                  DSPMOD(*DS4)
     A
     A  95                                  GRDCLR((*POS (*DS3 4 4 10 60) +
     A                                      (*DS4 4 4 10 120)))
     A
     A                                      GRDLIN((*POS (*DS3 6 4 20) +
     A                                      (*DS4 6 4 110)) (*TYPE LOWER)  +
     A                                      (*COLOR RED) (*LINTYP DBL))
     A
     A          R GRDREC3                   GRDRCD
     A
     A  95                                  GRDCLR((*POS (&SCROW &SCOL &DPTH &WDTH)));
     A
     A                                      GRDLIN((*POS (6 4 20)) +
     A                                      (*TYPE LOWER)  +
     A                                      (*COLOR RED) (*LINTYP DBL))
     A
     A            SROW           3S 0P
     A            SCOL           3S 0P
     A            DPTH           3S 0P
     A            WDTH           3S 0P
     A
```

When the GRDREC1 record is written, the display screen is cleared of all grid structures.

When the GRDREC2 record is written to a 24 by 80 display or the DSPMOD keyword is turned off and option indicator 95 is turned on, a rectangle starting at row 4, column 4, depth of 10 rows, and width of 60 columns will be cleared. If the GRDREC2 record is written to a 27 by 132 display and the DSPMOD keyword is turned on and option indicator 95 is turned on, a rectangle starting a at row 4, column 4, depth of 10 rows, and width of 120 columns will be cleared. The GRDCLR keyword will be processed before the GRDLIN keyword so existing grids will be cleared before any new grids are drawn.

When the GRDREC3 record is written and option indicator 95 is turned on, the GRDCLR keyword is processed. The position and size of the rectangle for the GRDCLR keyword will be determined at run time from the appropriate p-field. The GRDCLR keyword will be processed before the GRDLIN keyword so existing grids will be cleared before any new grids are drawn.

GRDLIN (Grid Line) keyword

You use this record-level keyword to define the shape, positioning, and attributes for the line structure. This keyword defines whether the line is erased, displayed, or not processed.

The format of the keyword is:

```
GRDLIN((*POS([*DS3] [*DS4]
start line | &start-line-field
start column | &start-column-field
length | &length-field
[(*TYPE type of line
[repeat | &repeat-field]
[interval rule | &interval-field])]
[(*COLOR color of line | &color-field)]
[(*LINTYP type of line | &lintyp-field)]
[(*CONTROL | &control-field)]
```

The *POS parameter is a required parameter. This parameter allows display size and conditioning of the GRDLIN keyword. Coding *DS3 or *DS4 with the *POS parameter, you can have 2 different start row, start column, and length values depending on the display size being used. DSPSIZ keyword must be coded on the file level.

If a field name is specified, the field must exist in the record format, data type S, usage P, length of 3, and zero decimal positions.

The type parameter is a required parameter. The valid values for the type parameter are:

Value
Meaning

UPPER
Horizontal line on the upper character border

LOWER
Horizontal line on the lower character border

RIGHT
Vertical line on the right character border

LEFT
Vertical line on the left character border

The repeat parameter specifies the number of times the line is to be repeated. The interval parameter specifies the number of character spaces between the repeated lines.

The default for the type parameter is upper. If neither the repeat value nor the interval value is coded, a single grid line is drawn. The repeat and interval defaults are 1.

If a field name is specified, the field must exist in the record format and must be defined as data type S, usage P, and length greater than 3.

The *COLOR and *LINTYP parameter defines the color and attributes of the box. P-fields can be used to define or change the attributes at run time.

For more information about the *COLOR and *LINTYP parameters, see “GRDATR (Grid Attribute) keyword” on page 259.

If NONE is defined by the GRDLIN keyword, the color set by the GRDATR keyword will be used.

If a p-field is specified for either the COLOR or LINTYP parameter, the field must exist in the record format. The field is defined as data type A, usage P, and length of 1.

The *CONTROL parameter specifies the whether the GRDLIN is to be displayed, erased from the screen, or ignored (similar to optioning off the keyword). The field must exist in the record format and must be defined as data type S, usage P, and length of 1. If the p-field is set to 0, the grid line will be displayed. If the p-field is set to 1, the GRDLIN keyword will not be processed. If the p-field is set to -1, the grid line record currently shown will be cleared. If the p-field is set to something other than the defined values, then the default 0 will be used.

Grid line support requires DBCS equipment. This equipment should have the capability of calling Japanese DOS.

Option indicators are valid for this keyword.

Example

The following example shows how to specify the GRDLIN keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A                                     DSPSIZ(*DS3 *DS4)
     A                                     GDRATR((*COLOR WHT) (*LINTYP SLD))
     A          R GRDREC1                  GRDRCD
     A                                     DSPMOD(*DS4)
     A                                     GRDLIN((*POS (*DS3 2 1 80) +
     A                                     (*DS4 2 1 132)) (*TYPE LOWER))
     A                                     GRDLIN((*POS (*DS3 4 6 20) +
     A                                     (*DS4 4 6 22)) (*TYPE RIGHT 4 15) +
     A                                     (*COLOR RED)  (*LINTYP DBL) +
     A                                     (*CONTROL &CNTL1));
     A
     A                                     GRDLIN((*POS (8 1 &LEN1); +
     A                                     (*TYPE LOWER 3 6) +
     A                                     (*COLOR &CLR1); (*LINTYP &LNTP1); +
     A                                     (*CONTROL &CNTL2));
     A            CNTL1          1S 0P
     A            CNTL2          1S 0P
     A            LEN1           3S 0P
     A            LNTP1          1A  P
     A            CLR1           1S 0P
     A
```

When the GRDREC1 record is written:

- If record is written to a 24 by 80 display or DSPMOD is turned off, then:
  1. A horizontal line is drawn on the bottom character edge starting at row 2 and column 1. The length of the line is 80 columns long. The lines of the grid are white in color and have a solid line type defined by the file-level GRDATR keyword.
  2. If the value in the p-field CNTL1 equals 0, 4 vertical lines are drawn on the right border of characters in column 6, 21, 36, and 51. Each line is 20 rows long. The grid line is red using double lines. If the p-field CNTL1 value is a -1, the box is erased. If the p-field CNTL1 value is a 1, no action is taken by the GRDLIN keyword.
  3. If the value in the p-field CNTL2 equals 0, 3 horizontal lines are drawn at the bottom character edge of rows 8, 14, and 20. The length of the lines is determined at run time from the value in the p-field LEN1. If the value in that p-field is greater than the width of the display, the value is truncated to the display width. The color and line value is determined at run time from the p-field CLR1 and LNTP1. If the p-field CNTL2 value is 1, no action is taken by the GRDLIN keyword.
- If record is written to a 27 by 132 display and DSPMOD is turned on, then:
  1. A horizontal line is drawn on the bottom character edge starting at row 2 and column 1. The length of the line is 132 columns long. The lines of the grid are white in color and have a solid line type defined by the file-level GRDATR keyword.
  2. If the value in the p-field CNTL1 equals 0, 4 vertical lines are drawn on the right border of characters in column 6, 21, 36, and 51. Each line is 22 rows long. The grid line is red using double lines. If the p-field CNTL1 value is a -1, the box is erased. If the p-field CNTL1 value is a 1, no action is taken by the GRDLIN keyword.
  3. If the value in the p-field CNTL2 equals 0, 3 horizontal lines are drawn at the bottom character edge of rows 8, 14, and 20. The length of the lines is determined at run time from the value in the p-field LEN1. If the value in that p-field is greater than the width of the display, the value is truncated to the display width. The color and line value are determined at run time from the p-field CLR1 and LNTP1. If the p-field CNTL1 value is a -1, the box is erased. If the p-field CNTL2 value is 1, no action is taken by the GRDLIN keyword.

GRDRCD (Grid Record) keyword

You use this record-level keyword to define a grid line structure.

A grid line is defined as:

- Upper horizontal line of a character box
- Lower horizontal line of a character box
- Left vertical line of a character box
- Right vertical line of a character box

This keyword has no parameters.

A grid line record can contain one or more GRDBOX or GRDLIN keywords that define the grid structures; otherwise, it can contain the GRDCLR keyword to remove grid line structures from the display. The grid line record can only contain the GRDCLR keyword to clear the grid structure.

A record with the GRDRCD keyword specified must contain only grid related keywords or keywords needed to define a window. It cannot contain any other displayable fields. There can be program-to-system fields on the record that define the allowable parameters on the grid related keywords.

The following keywords are allowed on a record containing the GRDRCD keyword:

DSPMOD	GRDLIN	USRRSTDSP
FRCDTA	RETKEY	WDWBORDER
GRDATR	RETCMDKEY	WDWTITLE
GRDBOX	RMVWDW	WINDOW
GRDCLR		

The grid record can contain a window definition.

Grid line support requires DBCS equipment. This equipment should have the capability of calling Japanese DOS.

Option indicators are not valid for this keyword.

Example

The following example shows how to specify the GRDRCD keyword.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A
     A          R GRDREC1                   GRDRCD
     A
```

IGCALTTYP (Alternative Data Type) keyword

You specify this field-level keyword to change alphanumeric character fields that are capable of input and output to DBCS fields with data type O.

This keyword has no parameters.

Put the keyword into effect by specifying IGCDTA(*YES) on the CRTDSPF, CHGDSPF, and OVRDSPF commands. Fields specified with this keyword are DBCS fields when you specify IGCDTA(*YES) and are alphanumeric character fields when you specify IGCDTA(*NO). For example, you can create the file by specifying IGCDTA(*NO) on the CRTDSPF command. When using the file to display DBCS data, override the file with the OVRDSPF command, specifying IGCDTA(*YES). To override the display file IGCDSPF, type: `OVRDSPF FILE(IGCLIB/IGCDSPF) IGCDTA(*YES)`

Consider the following rules, when using the IGCALTTYP keyword:

- Specify this keyword only for input- and output-capable fields whose keyboard shift type is A, N, X, W, or I. Do not specify this keyword for DBCS fields.

Note: Fields specified with IGCDTA(*YES) are recognized as alphanumeric OPEN data type fields (O) when keyboard shift type is defined as N, X, W, or I.

- The following keywords are not allowed with the IGCALTTYP keyword:

  AUTO(RAZ)
  BLKFOLD
  CHECK(M10 M11 M10F M11F RL RZ VN VNE)
  CMP(EQ GE GT LE LT NE NG NL)
  COMP(EQ GE GT LE LT NE NG NL)
  DUP
  RANGE
  VALUES

Option indicators are not allowed with IGCALTTYP.

Example

The following example shows how to specify the IGCALTTYP keyword on the DDS coding form.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A          R RECORD
00020A            FLDA          79A  I 23  2IGCALTTYP
     A
```

When the IGCALTTYP keyword is put into effect, FLDA can contain DBCS data.

IGCCNV (DBCS Conversion) keyword

This file-level keyword enables double-byte character set (DBCS) conversion in display files, an alternative to directly typing in DBCS characters from a keyboard.

The format for the keyword is `IGCCNV(CFnn line-number)`.

The first parameter, CFnn, identifies which command function key, when pressed, begins and ends the conversion function. Specify any CF key (CF01 through CF24) as the parameter value. Do not specify a CF key that has already been assigned a function.

The second parameter, line-number, identifies where (on the display) the system should place the conversion prompt line. When you type the alphanumeric word to be converted on the conversion prompt line, the system displays related DBCS words. The prompt line requires an entire display line and looks like this:

```
_ ______  _ XXXXXXXXXXXXXXXXXXXXXX
```

The underlined fields (_) are input fields, in which you type the word to be converted and specify the type of conversion to be performed.

The field indicated with XXXX is an output field, in which the system displays DBCS words related to the alphanumeric entry to be converted.

The prompt line can be placed anywhere on the display, as long as it does not overlap other displayed records that contain input fields.

Consider the following rules when using the IGCCNV keyword:

- Use this keyword only with files displayed on DBCS display stations.
- At least one field in the file must be an input-capable DBCS field, or an input-capable field specified with the IGCALTTYP keyword.
- Avoid using the IGCCNV keyword with the CHECK(ME) keyword. Use of DBCS conversion with a mandatory-entry field causes operational problems.
- Avoid using the IGCCNV keyword with field validation keywords (CHECK, CMP, RANGE, and VALUES). Using this keyword causes DBCS conversion to work improperly.

- You must define the file for a 24 x 80 display.
- Do not display the DBCS conversion format over a format that uses the USRDFN (user-defined) keyword.
- Option indicators are not allowed with this keyword.

Example

The following example shows how to specify the IGCCNV keyword on the DDS coding form.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00100A*
00101A*
00102A                                      IGCCNV(CF24 24)
00103A          R MENU
     A
```

A user can press the F24 key to begin and end DBCS conversion. Conversion can be used on all input-capable DBCS fields. The conversion format is displayed on line 24.

Additional considerations for describing display files that contain DBCS data

Be aware of these additional considerations when you describe display files that contain double-byte character set (DBCS) data.

- Specify IGCDTA(*YES) on the CRTDSPF command when DBCS data is present in the file, but not indicated in DDS. For example, specify IGCDTA(*YES) if the file sends messages that are DBCS (DDS keyword MSGCON).
- Prevent users from using display files to insert alphanumeric data into DBCS database files by specifying the keyboard shift for a field in a field reference file rather than in a display file. Users cannot type alphanumeric data in input-capable fields of DBCS display files and, therefore, cannot type alphanumeric data into the database file.

  Specify data type J or G in a database field reference file and R in position 29 of the associated display file.

  Use data type J or G for all fields in a field reference file to reduce the possibility of incorrectly setting the default keyboard shift to O (open).
- Describe fields in the file as DBCS fields to cause the system to consider the file to be DBCS even if you do not specify IGCDTA(*YES) on the CRTDSPF command.
- The system displays the DBCS data that does not fit on one display line onto the next display line with the following effects:
  - DDS sends a warning message stating that it split DBCS characters for constant and initialized fields containing DBCS data.
  - DDS sends a warning message stating that it split DBCS characters if you specified the J, E, or G data type.
  - DDS does not send a warning message stating that it split DBCS characters if you specified the O data type. DDS warns you of the potential for this problem when the file is created.
  - The second display line of a continued field might not make sense if the system must split a DBCS character in order to continue the line.
- Text with bracketed-DBCS characters can be used anywhere that comments and character strings are allowed.
- Consider the following rules when you specify subfiles:
  - Use the SFLMSG keyword to create DBCS messages by typing DBCS data for the character string in the message.

    Check the length of the message. The space available to display it must be long enough to contain the message. DDS warns you when a display field might be truncated. However, the field might be

truncated in the middle of a DBCS character, and the data displayed following the truncated character will not make sense.

- The system ignores the SFLEND keyword when displaying a plus sign (+) to indicate that more records exist in the subfile. When displaying the plus sign, the system writes over a DBCS character.

- Consider the following rules when you specify the MSGID keyword:
  - If the message text contains DBCS characters, and the message length exceeds the MSGID field length, the message text is truncated so that it ends with an alphanumeric character. If the truncation occurs in the middle of a DBCS character, the text truncates after the previous DBCS character and a shift-in character is added to the end of the text.
  - If the message text contains DBCS characters, either define the MSGID field so that it does not wrap to the next line, or make sure the message text does not wrap in the middle of a DBCS character.

Code license and disclaimer information

IBM grants you a nonexclusive copyright license to use all programming code examples from which you can generate similar function tailored to your own specific needs.

SUBJECT TO ANY STATUTORY WARRANTIES WHICH CANNOT BE EXCLUDED, IBM, ITS PROGRAM DEVELOPERS AND SUPPLIERS MAKE NO WARRANTIES OR CONDITIONS EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OR CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT, REGARDING THE PROGRAM OR TECHNICAL SUPPORT, IF ANY.

UNDER NO CIRCUMSTANCES IS IBM, ITS PROGRAM DEVELOPERS OR SUPPLIERS LIABLE FOR ANY OF THE FOLLOWING, EVEN IF INFORMED OF THEIR POSSIBILITY:

1. LOSS OF, OR DAMAGE TO, DATA;
2. DIRECT, SPECIAL, INCIDENTAL, OR INDIRECT DAMAGES, OR FOR ANY ECONOMIC CONSEQUENTIAL DAMAGES; OR
3. LOST PROFITS, BUSINESS, REVENUE, GOODWILL, OR ANTICIPATED SAVINGS.

SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF DIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, SO SOME OR ALL OF THE ABOVE LIMITATIONS OR EXCLUSIONS MAY NOT APPLY TO YOU.

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
1623-14, Shimotsuruma, Yamato-shi
Kanagawa 242-8502 Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Software Interoperability Coordinator, Department YBWA
3605 Highway 52 N
Rochester, MN 55901
U.S.A.

© Copyright IBM Corp. 1999, 2013

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs.

© Copyright IBM Corp. _enter the year or years_.

Programming interface information

This SQL call level interface publication documents intended Programming Interfaces that allow the customer to write programs to obtain the services of IBM i.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Linux® is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other product and service names might be trademarks of IBM or other companies.

Terms and conditions

Permissions for the use of these publications is granted subject to the following terms and conditions.

Personal Use: You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative works of these publications, or any portion thereof, without the express consent of IBM.

Commercial Use: You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

Index

Numerics

3270 remote attachment 246

A

ALARM (Audible Alarm) keyword 29
ALIAS (Alternative Name) keyword 29
Allow Graphics (ALWGPH) keyword 33
Allow Roll (ALWROL) keyword 34
Alternative Help (ALTHELP) keyword 30
Alternative Name (ALIAS) keyword 29
Alternative Page Down/Alternative Page Up (ALTPAGEDWN/ALTPAGEUP) keywords 31
Alternative Record Name (ALTNAME) keyword 31
ALTHELP (Alternative Help) keyword 30
ALTNAME (Alternative Record Name) keyword
    in the System/36 environment 248
ALTPAGEDWN/ALTPAGEUP (Alternative Page Down/Alternative Page Up) keywords 31
ALWGPH (Allow Graphics) keyword 33
ALWROL (Allow Roll) keyword 34
assigning
    menu-bar switch key 155
    menu-cancel key 156
ASSUME (Assume) keyword 35
asterisk fill, specifying 100
Audible Alarm (ALARM) keyword 29
AUTO (Auto) keyword 36

B

Blank Fold (BLKFOLD) keyword 39
BLANKS (Blanks) keyword 37
BLINK (Blink) keyword 39
BLKFOLD (Blank Fold) keyword 39

C

CAnn (Command Attention) keyword 39
CCSID (Coded Character Set Identifier) keyword
    UCS-2 level 1 254
CFnn (Command Function) keyword 41
CHANGE (Change) keyword
    in the System/36 environment 248
Change Display File (CHGDSPF) command 114, 118
Change Input Default (CHGINPDFT) keyword 61
Character Identifier (CHRID) keyword 66
CHCACCEL (Choice Accelerator Text) keyword 44
CHCAVAIL (Choice Color/Display Attributes when Available) keyword 45
CHCCTL (Choice Control) keyword 47
CHCSLT (Choice Color/Display Attribute when Selected) 49
CHCUNAVAIL (Choice Color/Display Attribute When Unavailable) keyword 50
CHECK (Check) keyword 52
Check Message Identifier (CHKMSGID) keyword 63
CHGDSPF (Change Display File) command 114, 118
CHGINPDFT (Change Input Default) keyword 61
CHKMSGID (Check Message Identifier) keyword 63
CHOICE (Selection Field Choice) keyword 64
Choice Accelerator Text (CHCACCEL) keyword 44
Choice Color/Display Attribute when Selected (CHCSLT) keyword 49
Choice Color/Display Attribute when Unavailable (CHCUNAVAIL) keyword 50
Choice Color/Display Attributes when Available (CHCAVAIL) keyword 45
Choice Control (CHCCTL) keyword 47
choice field
    controlling 47
CHRID (Character Identifier) keyword 66
CLEAR (Clear) keyword 66
Clear Line (CLRL) keyword 67
CLRL (Clear Line) keyword 67
CMP (Comparison) keyword 69
CNTFLD (Continued-Entry Field) keyword 69, 258
COLOR (Color) keyword 70
color/display attribute
    selected 49
    unavailable 50
Command Attention (CAnn) keyword 39
Command Function (CFnn) keyword 41
COMP (Comparison) keyword 74
Comparison (CMP) keyword 69
conditioning positional entry 3
constant fields positional entry 7
continued-entry field
    creating 69
    DBCS consideration 258
    menu-bar separator 152
Continued-Entry Field (CNTFLD) keyword 69, 258
controlling
    choice field 47
Create Display File (CRTDSPF) command 114, 118
Create Edit Description (CRTEDTD) command 102
creating
    continued-entry field 69
    cursor-progression field 116
    edit mask 104
    entry field attribute 109
    menu bar 147
    menu-bar choice 148
    menu-bar separator 152
    multiple-choice selection field 145
    pull-down menu 175
    subfile cursor progression 191
    validate numeric 234
    window 242
CRTDSPF (Create Display File) command 114, 118
CRTEDTD (Create Edit Description) command 102
CSRINPONLY (Cursor Movement to Input-Capable Positions Only) keyword 75

CSRLOC (Cursor Location) keyword 76
cursor control 59
Cursor Location (CSRLOC) keyword 76
Cursor Movement to Input-Capable Positions Only (CSRINPONLY) keyword 75
cursor progression
  field 116
Cursor Progression Field (FLDCSRPRG) keyword 116

D

data type positional entry 11
data types that are valid for display files 12
DATE (Date) keyword 77
Date Format (DATFMT) keyword 78
Date Separator (DATSEP) keyword 79
DATFMT (Date Format) keyword 78
DATSEP (Date Separator) keyword 79
DBCS Alternative Data Type (IGCALTTYP) keyword 266
DBCS considerations for keyword entries 257
DBCS considerations for positional entries 256
DBCS Conversion (IGCCNV) keyword 267
DDS file considerations
  Unicode 253
decimal positional entry 24
Default (DFT) keyword 80
Default Value (DFTVAL) keyword 81
defining
  help area 120
  selection field choice 64
  single-choice selection field 225
defining a display file 1
Delete Check (DLTCHK) keyword 82
Delete Edit (DLTEDT) keyword 82
DFT (Default) keyword 80
DFTVAL (Default Value) keyword 81
Display Attribute (DSPATR) keyword 83
display files
  defining with DDS 1
  keyword entries 29
  record specification 1
  specifying file name 2
  Unicode 253
Display Mode (DSPMOD) keyword 90
Display Right to Left (DSPRL) keyword 91
Display Size (DSPSIZ) keyword 92
display sizes
  primary and secondary 92
  special cases 93
displaying
  menu bar 151
DLTCHK (Delete Check) keyword 82
DLTEDT (Delete Edit) keyword 82
DSPATR (Display Attribute) keyword 83
DSPMOD (Display Mode) keyword 90
DSPRL (Display Right to Left) keyword 91
DSPSIZ (Display Size) keyword 92

E

Edit Code (EDTCDE) keyword 99
edit codes, user-defined 102
edit description
  creating 102
edit mask
  creating 104
Edit mask (EDTMSK) keyword 104
Edit Word (EDTWRD) keyword 104
edit word parts 105
EDTCDE (Edit Code) keyword 99
EDTMSK (Edit mask) keyword 104
EDTWRD (Edit Word) keyword 104
ENTFLDATR (Entry Field Attribute) keyword 109
entry field attribute
  creating 109
Entry Field Attribute (ENTFLDATR) keyword 109
ERASE (Erase) keyword 110
Erase Input (ERASEINP) keyword 111
ERASEINP (Erase Input) keyword 111
ERRMSG (Error Message) keyword 112
ERRMSGID (Error Message Identifier) keyword 112
Error Message (ERRMSG) keyword 112
Error Message Identifier (ERRMSGID) keyword 112
Error Subfile (ERRSFL) keyword 115
ERRSFL (Error Subfile) keyword 115

F

field name positional entry 7
field specification, incorrect 28
FLDCSRPRG (Cursor Progression Field) keyword 116
floating-currency symbol, specifying 100
Floating-Point Precision (FLTPCN) keyword 117
Floating-Point to Fixed Decimal (FLTFIXDEC) keyword 116
FLTFIXDEC (Floating-Point to Fixed Decimal) keyword 116
FLTPCN (Floating-Point Precision) keyword 117
Force Data (FRCDTA) keyword 118
FRCDTA (Force Data) keyword 118

G

GETRETAIN (Get Retain) keyword 119
GRDATR (Grid Attribute) keyword 259
GRDBOX (Grid Box) keyword 260
GRDCLR (Grid Clear) keyword 262
GRDLIN (Grid Line) keyword 264
GRDRCD (Grid Record) keyword 266
Grid Attribute (GRDATR) keyword 259
Grid Box (GRDBOX) keyword 260
Grid Clear (GRDCLR) keyword 262
Grid Line (GRDLIN) keyword 264
Grid Record (GRDRCD) keyword 266

H

HELP (Help) keyword
  in the System/36 environment 249
help area
  defining 120
Help Area (HLPARA) keyword 120
Help Boundary (HLPBDY) keyword 123
Help Cleared (HLPCLR) keyword 124
Help Command Key (HLPCMDKEY) keyword 124
Help Document (HLPDOC) keyword 125
Help Excluded (HLPEXCLD) keyword 127

Help Full (HLPFULL) keyword 127
help identifier
  specifying 128
Help Identifier (HLPID) keyword 128
Help Panel Group (HLPPNLGRP) keyword 128
Help Record (HLPRCD) keyword 129
Help Return (HLPRTN) keyword 130
Help Search Index (HLPSCHIDX) keyword 131
Help Sequencing (HLPSEQ) keyword 132
Help Title (HLPTITLE) keyword 132
HLPARA (Help Area) keyword 120
HLPBDY (Help Boundary) keyword 123
HLPCLR (Help Cleared) keyword 124
HLPCMDKEY (Help Command Key) keyword 124
HLPDOC (Help Document) keyword 125
HLPEXCLD (Help Excluded) keyword 127
HLPFULL (Help Full) keyword 127
HLPID (Help Identifier) keyword 128
HLPPNLGRP (Help Panel Group) keyword 128
HLPRCD (Help Record) keyword 129
HLPRTN (Help Return) keyword
  in the System/36 environment 249
HLPSCHIDX (Help Search Index) keyword 131
HLPSEQ (Help Sequencing) keyword 132
HLPTITLE (Help Title) keyword 132
HOME (Home) keyword 133
HTML (Hyper Text Markup Language) keyword 134
Hyper Text Markup Language (HTML) keyword 134

I

IGCALTTYP (DBCS Alternative Data Type) keyword 266
IGCCNV (DBCS Conversion) keyword 267
INDARA (Indicator Area) keyword 135
Indicator Area (INDARA) keyword 135
Indicator Text (INDTXT) keyword 135
INDTXT (Indicator Text) keyword 135
Initialize Record (INZRCD) keyword 140
INVITE (Invite) keyword 136
INZRCD (Initialize Record) keyword 140

K

KEEP (Keep) keyword 141
keyboard control 56
keyboard shift positional entry 11
keyword entries
  DBCS considerations 257
keyword entries for display files 29

L

length positional entry 10
location positional entry 26
LOCK (Lock) keyword 141
Log Input (LOGINP) keyword 142
Log Output (LOGOUT) keyword 142
LOGINP (Log Input) keyword 142
LOGOUT (Log Output) keyword 142
LOWER (Lower) keyword 143

M

MAPVAL (Map Values) keyword 143
MDTOFF (Modified Data Tag Off) keyword 144
menu bar
  creating 147
  displaying 151
Menu Bar (MNUBAR) keyword 147
menu-bar choice
  creating 148
Menu-Bar Choice (MNUBARCHC) keyword 148
Menu-Bar Display (MNUBARDSP) keyword 151
menu-bar separator
  keyword 152
Menu-Bar Separator (MNUBARSEP) keyword 152
menu-bar switch key
  assigning 155
Menu-Bar Switch Key (MNUBARSW) keyword 155
menu-cancel key
  assigning 156
Menu-Cancel Key (MNUCNL) keyword 156
Message Constant (MSGCON) keyword 160
Message Identifier (MSGID) keyword 161
Message Location (MSGLOC) keyword 163
MLTCHCFLD (Multiple-Choice Selection Field) keyword 145
MNUBAR (Menu Bar) keyword 147
MNUBARCHC (Menu-Bar Choice) keyword 148
MNUBARDSP (Menu-Bar Display) keyword 151
MNUBARSEP (Menu-Bar Separator) keyword 152
MNUBARSW (Menu-Bar Switch Key) keyword 155
MNUCNL (Menu-Cancel Key) keyword 156
Modified Data Tag Off (MDTOFF) keyword 144
MOUBTN (Mouse Buttons) keyword 157
Mouse Buttons (MOUBTN) keyword 157
MSGCON (Message Constant) keyword 160
MSGID (Message Identifier) keyword
  in the System/36 environment 249
MSGLOC (Message Location) keyword 163
multiple-choice selection field
  creating 145
  keyword 145
Multiple-Choice Selection Field (MLTCHCFLD) keyword 145

N

name positional entry 7
name type positional entry 7
No Coded Character Set Identifier (NOCCSID) keyword 164
NOCCSID (No Coded Character Set Identifier) keyword 164

O

Open Printer File (OPENPRT) keyword 164
OPENPRT (Open Printer File) keyword 164
OVERLAY (Overlay) keyword 165
Override Attribute (OVRATR) keyword 166
Override Data (OVRDTA) keyword 167
OVRATR (Override Attribute) keyword 166
OVRDTA (Override Data) keyword 167

P

PAGEDOWN/PAGEUP (Page down/Page up) keywords 167

parts of edit word 105
Passed Record (PASSRCD) keyword 168
PASSRCD (Passed Record) keyword 168
positional entries
    DBCS considerations 256
positional entries for display files 3
primary and secondary display sizes 92
PRINT (Print) keyword 169
PRINT(*PGM) keyword
    in the System/36 environment 250
programming for the Dup key 98
PROTECT (Protect) keyword 171
PSHBTNCHC (Push Button Field Choice) keyword 172
PSHBTNFLD (Push Button Field) keyword 173
pull-down menu
    creating 175
Pull-Down Menu (PULLDOWN) keyword 175
PULLDOWN (Pull-Down Menu) keyword 175
Push Button Field (PSHBTNFLD) keyword 173
Push Button Field Choice (PSHBTNCHC) keyword 172
Put with Explicit Override (PUTOVR) keyword 176
PUTOVR (Put with Explicit Override) keyword 176
PUTRETAIN (Put-Retain) keyword 178

R

RANGE (Range) keyword 179
record format name positional entry 7
record specification 1
REF (Reference) keyword 180
reference positional entry 8
Referenced Field (REFFLD) keyword 181
REFFLD (Referenced Field) keyword 181
Remove Window (RMVWDW) keyword 182
reserved positional entry 7
restoring reversed-image fields 114, 207
restrictions on validity checking with DUP keyword 98
Retain Command Keys (RETCMDKEY) keyword 182
Retain Function Keys (RETKEY) keyword 182
Retain Lock Status (RETLCKSTS) keyword 182
RETCMDKEY (Retain Command Keys) keyword 182
RETKEY (Retain Function Keys) and RETCMDKEY (Retain Command Keys) keywords
    in the System/36 environment 251
RETKEY (Retain Function Keys) keyword 182
RETLCKSTS (Retain Lock Status) keyword 182
Return Cursor Location (RTNCSRLOC) keyword 184
Return Data (RTNDTA) keyword 186
reversed-image fields, restoring 114
RMVWDW (Remove Window) keyword 182
ROLLUP/ROLLDOWN (Roll up/Roll down) keyword 183
RTNCSRLOC (Return Cursor Location) keyword 184
RTNDTA (Return Data) keyword 186

S

selection field choice
    defining 64
Selection Field Choice (CHOICE) keyword 64
sequence number positional entry 3
SETOF (Set Off) keyword 187
SETOFF (Set Off) keyword 187
SFL (Subfile) keyword 188
SFLCHCCTL (Subfile Choice Control) keyword 189
SFLCLR (Subfile Clear) keyword 190
SFLCSRPRG (Subfile Cursor Progression) keyword 191
SFLCSRRRN (Subfile Cursor Relative Record Number) keyword 191
SFLCTL (Subfile Control) keyword 191
SFLDLT (Subfile Delete) keyword 193
SFLDROP (Subfile Drop) keyword 194
SFLDSP (Subfile Display) keyword 195
SFLDSPCTL (Subfile Display Control) keyword 195
SFLEND (Subfile End) keyword 196
SFLENTER (Subfile Enter) keyword 199
SFLFOLD (Subfile Fold) keyword 200
SFLINZ (Subfile Initialize) keyword 201
SFLLIN (Subfile Line) keyword 202
SFLMLTCHC (Subfile Multiple Choice Selection List) keyword 203
SFLMODE (Subfile Mode) keyword 205
SFLMSG (Subfile Message) keyword 206
SFLMSGID (Subfile Message Identifier) keyword 206
SFLMSGKEY (Subfile Message Key) keyword 208
SFLMSGRCD (Subfile Message Record) keyword 209
SFLNXTCHG (Subfile Next Changed) keyword 210
SFLPAG (Subfile Page) keyword 211
SFLPGMQ (Subfile Program Message Queue) keyword 213
SFLRCDNBR (Subfile Record Number) keyword 215
SFLRNA (Subfile Records Not Active) keyword 216
SFLROLVAL (Subfile Roll Value) keyword 217
SFLRTNSEL (Subfile Return Selected Choices) keyword 219
SFLSCROLL (Subfile Scroll) keyword 219
SFLSIZ (Subfile Size) keyword 220
SFLSNGCHC (Subfile Single Choice Selection List) keyword 222
single-choice selection field
    defining 225
Single-Choice Selection Field (SNGCHCFLD) keyword 225
SLNO (Starting Line Number) keyword 224
SNGCHCFLD (Single-Choice Selection Field) keyword 225
specification type positional entry 7
specifying
    color/display attribute unavailable 50
    color/display attributes available 45
    color/display selected 49
    help identifier 128
specifying display file name 2
Starting Line Number (SLNO) keyword 224
Subfile (SFL) keyword 188
Subfile Choice Control (SFLCHCCTL) keyword 189
Subfile Clear (SFLCLR) keyword 190
Subfile Control (SFLCTL) keyword 191
subfile cursor progression
    creating 191
Subfile Cursor Progression (SFLCSRPRG) keyword 191
Subfile Cursor Relative Record Number (SFLCSRRRN) keyword 191
Subfile Delete (SFLDLT) keyword 193
Subfile Display (SFLDSP) keyword 195
Subfile Display Control (SFLDSPCTL) keyword 195
Subfile Drop (SFLDROP) keyword 194
Subfile End (SFLEND) keyword 196
Subfile Enter (SFLENTER) keyword 199
Subfile Fold (SFLFOLD) keyword 200
Subfile Initialize (SFLINZ) keyword 201

Subfile Line (SFLLIN) keyword 202
Subfile Message (SFLMSG) keyword 206
Subfile Message Identifier (SFLMSGID) keyword 206
Subfile Message Key (SFLMSGKEY) keyword 208
Subfile Mode (SFLMODE) keyword 205
Subfile Multiple Choice Selection (SFLMLTCHC) keyword 203
Subfile Next Changed (SFLNXTCHG) keyword 210
Subfile Page (SFLPAG) keyword 211
Subfile Program Message Queue (SFLPGMQ) keyword 213
Subfile Record Number (SFLRCDNBR) keyword 215
Subfile Records Not Active (SFLRNA) keyword 216
Subfile Return Selected Choices (SFLRTNSEL) keyword 219
Subfile Roll Value (SFLROLVAL) keyword 217
Subfile Scroll (SFLSCROLL) keyword 219
Subfile Single Choice Selection List (SFLSNGCHC) keyword 222
Subfile Size (SFLSIZ) keyword 220
SYSNAME (System Name) keyword 227
System Name (SYSNAME) keyword 227
System/36 environment
    ALTNAME (Alternative Record Name) keyword 248
    CHANGE (Change) keyword 248
    HELP (Help) keyword 249
    HLPRTN (Help Return) keyword 249
    MSGID (Message Identifier) keyword 249
    PRINT(*PGM) keyword 250
    RETKEY (Retain Function Keys) and RETCMDKEY (Retain Command Keys) keywords 251
    USRDSPMGT (User Display Management) keyword 252

T

TEXT (Text) keyword 228
TIME (Time) keyword 228
Time Format (TIMFMT) keyword 229
Time Separator (TIMSEP) keyword 230
TIMFMT (Time Format) keyword 229
TIMSEP (Time Separator) keyword 230
title
    window 239

U

UCS-2 level 1
    CCSID (Coded Character Set Identifier) keyword 254
Unicode
    DDS file considerations 253
    display files 253
    positional entries 253
UNLOCK (Unlock) keyword 231
usage positional entry 24
USER (User) keyword 232
User Display Management (USRDSPMGT) keyword 233
User Restore Display (USRRSTDSP) keyword 233
User-Defined (USRDFN) keyword 232
user-defined edit codes 102
USRDFN (User-Defined) keyword 232
USRDSPMGT (User Display Management) keyword
    in the System/36 environment 252
USRRSTDSP (User Restore Display) keyword 233

V

Valid Command Key (VLDCMDKEY) keyword 235
valid data types for display files 12
validate numeric
    creating 234
Validate Numeric (VALNUM) keyword 234
validity checking 52
VALNUM (Validate Numeric) keyword 234
VALUES (Values) keyword 234
VLDCMDKEY (Valid Command Key) keyword 235

W

WDWBORDER (Window Border) keyword 236
WDWTITLE (Window Title) keyword 239
window
    creating 242
    title 239
WINDOW (Window) keyword 242
Window Border (WDWBORDER) keyword 236
Window Title (WDWTITLE) keyword 239
word
    wrap 245
Word Wrap (WRDWRAP) keyword 245
wrap
    word 245
WRDWRAP (Word Wrap) keyword 245



Product Number: 5770-SS1