

The GnuCOBOL 3.2-early-dev Grammar

FOR R3407

Edward Hart

edward.dan.hart@gmail.com

December 27, 2019

COBOL is an industry language and is not the property of any company or group of companies, or of any organisation or group of organisations.

No warranty, expressed or implied, is made by any contributor, or by the CODASYL COBOL Committee,¹ as to the accuracy and functioning of the programming system and language. Moreover, no responsibility is assumed by any contributor, or by the committee, in connection therewith.

The authors and copyright holders of the copyrighted materials used herein are:

- *FLOW-MATIC* [trademark of Sperry Rand Corporation] *Programming for the UNIVAC @ I and II Data Automation Systems*, copyrighted 1958, 1959 by Sperry Rand Corporation,²
- IBM Commercial Translator, Form No. F28-8013, copyrighted 1959 by IBM, and
- FACT, DSI 27A5260-2760, copyrighted 1960 by Minneapolis-Honeywell.

They have specially authorised the use of this material, in whole or in part, in the COBOL specifications. Such authorisation extends to the reproduction and use of COBOL specifications in programming manuals or similar publications.

This work is typeset in Roboto.

Copyright © 2019 Edward Hart

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. Your attention is drawn to the copy of the license in Appendix A.

The moral rights of the author have been asserted.

¹The CODASYL COBOL committee was dissolved in 1992. Its work was continued by ANSI X3J4 and then INCITS PL22.4, which was itself dissolved in 2015.

²Sperry Rand's computer business is now part of Unisys.

Contents

Foreword	xiv
Changelog	xvi
1. GnuCOBOL 3.2-early-dev	xvi
2. GnuCOBOL 3.1-dev	xvii
3. GnuCOBOL 2.2	xxvii
1. Key	3
2. Language fundamentals	5
2.1. Lexical elements	5
2.1.1. COBOL words	5
2.1.2. User-defined words	5
2.1.3. Reserved words	5
2.1.4. Literals	5
2.2. References	11
2.3. Expressions	12
2.3.1. Arithmetic expressions	12
2.3.2. Concatenation expressions	13
2.3.3. Conditional expressions	13
2.4. Concepts	14
2.4.1. Files	14
2.4.2. Locales	15
2.4.3. Screens	15
2.4.4. User-defined functions	15
3. Compiler directives	17
3.1. CALL-CONVENTION directive	17
3.2. *CONTROL statement	18
3.3. COPY statement	19
3.4. D directive	20
3.5. DEFINE directive	21
3.6. DISPLAY directive	22
3.7. EJECT statement	23
3.8. IF directive	24
3.9. LEAP-SECOND directive	25
3.10. LISTING directive	26

Contents

3.11.	@OPTIONS directive	27
3.12.	PAGE directive	28
3.13.	PROCESS statement	29
3.14.	REPLACE statement	30
3.15.	SET directive	31
3.16.	SKIP1 statement	32
3.17.	SKIP2 statement	33
3.18.	SKIP3 statement	34
3.19.	SOURCE directive	35
3.20.	TITLE statement	36
3.21.	TURN directive	37
3.22.	Micro Focus directives	38
3.22.1.	ADDRSV directive	38
3.22.2.	ADDSYN directive	39
3.22.3.	CALLFH directive	40
3.22.4.	COMP1 directive	41
3.22.5.	FOLDCOPYNAME directive	42
3.22.6.	MAKESYN directive	43
3.22.7.	OVERRIDE directive	44
3.22.8.	REMOVE directive	45
3.22.9.	SOURCEFORMAT directive	46
3.23.	Predefined compilation variables	47
4.	Compilation group	50
5.	Identification division	53
5.1.	PROGRAM-ID paragraph	53
5.2.	FUNCTION-ID paragraph	54
5.3.	OPTIONS paragraph	55
6.	Environment division	57
6.1.	Configuration section	57
6.1.1.	SOURCE-COMPUTER paragraph	58
6.1.2.	OBJECT-COMPUTER paragraph	59
6.1.3.	SPECIAL-NAMES paragraph	60
6.1.4.	REPOSITORY paragraph	62
6.2.	Input-output section	63
6.2.1.	FILE-CONTROL paragraph	63
6.2.2.	I-O-CONTROL paragraph	68
7.	Data division	70
7.1.	File section	70
7.1.1.	File description entry	71
7.2.	Working-storage section	72

Contents

7.3.	Communication section	73
7.3.1.	Communication description entry	73
7.4.	Local-storage section	75
7.5.	Linkage section	76
7.6.	Report section	77
7.6.1.	Report description	77
7.7.	Screen section	80
7.7.1.	Screen description	82
7.8.	Record description	86
7.9.	Constant definition	88
7.10.	Data division clauses	89
7.10.1.	ANY LENGTH clause	89
7.10.2.	AUTO clause	90
7.10.3.	BACKGROUND-COLOR clause	91
7.10.4.	BACKGROUND-HIGH clause	92
7.10.5.	BACKGROUND-LOW clause	93
7.10.6.	BACKGROUND-STANDARD clause	94
7.10.7.	BELL clause	95
7.10.8.	BLANK clause	96
7.10.9.	BLANK WHEN ZERO clause	97
7.10.10.	BLINK clause	98
7.10.11.	BLOCK clause	99
7.10.12.	COLOR clause	100
7.10.13.	COLUMN clause	101
7.10.14.	CCOL clause	102
7.10.15.	CLINE clause	103
7.10.16.	CSIZE clause	104
7.10.17.	DATA RECORDS clause	105
7.10.18.	DESTINATION clause	106
7.10.19.	DESTINATION COUNT clause	107
7.10.20.	DESTINATION TABLE OCCURS clause	108
7.10.21.	END KEY clause	109
7.10.22.	Entry name	110
7.10.23.	ERASE clause	111
7.10.24.	ERROR KEY clause	112
7.10.25.	EXTERNAL clause	113
7.10.26.	EXTERNAL-FORM clause	114
7.10.27.	FOREGROUND-COLOR clause	115
7.10.28.	FROM clause	116
7.10.29.	FULL clause	117
7.10.30.	GLOBAL clause	118
7.10.31.	GRID clause	119
7.10.32.	HIGHLIGHT clause	120
7.10.33.	INITIAL clause	121

Contents

7.10.34. JUSTIFIED clause	122
7.10.35. LABEL RECORDS clause	123
7.10.36. LEFTLINE clause	124
7.10.37. Level-number	125
7.10.38. LINAGE clause	126
7.10.39. LINE clause	127
7.10.40. LOWLIGHT clause	128
7.10.41. MESSAGE COUNT clause	129
7.10.42. MESSAGE DATE clause	130
7.10.43. MESSAGE TIME clause	131
7.10.44. NEXT GROUP clause	132
7.10.45. NO ECHO clause	133
7.10.46. OCCURS clause	134
7.10.47. OVERLINE clause	136
7.10.48. PICTURE clause	137
7.10.49. PRESENT WHEN clause	138
7.10.50. PROMPT clause	139
7.10.51. RECORD clause	140
7.10.52. RECORDING MODE clause	141
7.10.53. REDEFINES clause	142
7.10.54. REPORT clause	143
7.10.55. REQUIRED clause	144
7.10.56. REVERSE-VIDEO clause	145
7.10.57. SECURE clause	146
7.10.58. STATUS KEY clause	147
7.10.59. SIGN clause	148
7.10.60. SIZE clause	149
7.10.61. SOURCE clause	150
7.10.62. STANDARD clause	151
7.10.63. SUM clause	152
7.10.64. SYMBOLIC DESTINATION clause	153
7.10.65. SYMBOLIC QUEUE clause	154
7.10.66. SYMBOLIC SOURCE clause	155
7.10.67. SYMBOLIC SUB-QUEUE-1 clause	156
7.10.68. SYMBOLIC SUB-QUEUE-2 clause	157
7.10.69. SYMBOLIC SUB-QUEUE-3 clause	158
7.10.70. SYMBOLIC TERMINAL clause	159
7.10.71. SYNCHRONIZED clause	160
7.10.72. TEXT LENGTH clause	161
7.10.73. TO clause	162
7.10.74. TYPE clause	163
7.10.75. UNDERLINE clause	164
7.10.76. USAGE clause	165
7.10.77. USING clause	168

Contents

7.10.78. VALUE clause	169
7.10.79. VALUE OF clause	170
7.10.80. VARYING clause	171
7.10.81. VOLATILE clause	172
8. Procedure division	174
8.1. Concepts	174
8.1.1. Exceptions	174
8.2. Procedure division header	179
8.3. Common phrases	180
8.3.1. Call-convention phrase	180
8.3.2. LINKAGE phrase	181
8.3.3. RETRY phrase	182
8.3.4. ROUNDED phrase	183
8.3.5. SIZE phrase	184
8.4. ACCEPT statement	185
8.5. ADD statement	191
8.6. ALLOCATE statement	193
8.7. ALTER statement	194
8.8. CALL statement	195
8.9. CANCEL statement	196
8.10. CLOSE statement	197
8.11. COMMIT statement	198
8.12. COMPUTE statement	199
8.13. CONTINUE statement	200
8.14. DELETE statement	201
8.15. DESTROY statement	202
8.16. DISABLE statement	203
8.17. DISPLAY statement	204
8.18. DIVIDE statement	211
8.19. ENABLE statement	212
8.20. ENTRY statement	213
8.21. EVALUATE statement	214
8.22. EXIT statement	215
8.23. FREE statement	216
8.24. GENERATE statement	217
8.25. GO TO statement	218
8.26. GOBACK statement	219
8.27. IF statement	220
8.28. INITIALIZE statement	221
8.29. INITIATE statement	222
8.30. INQUIRE statement	223
8.31. INSPECT statement	224
8.32. JSON GENERATE statement	225

Contents

8.33.	JSON PARSE statement	226
8.34.	MERGE statement	227
8.35.	MODIFY statement	228
8.36.	MOVE statement	229
8.37.	MULTIPLY statement	230
8.38.	NEXT SENTENCE statement	231
8.39.	OPEN statement	232
8.40.	PERFORM statement	233
8.41.	PURGE statement	235
8.42.	RAISE statement	236
8.43.	READ statement	237
8.44.	READY statement	238
8.45.	RECEIVE statement	239
8.46.	RELEASE statement	240
8.47.	RESET statement	241
8.48.	RETURN statement	242
8.49.	REWRITE statement	243
8.50.	ROLLBACK statement	244
8.51.	SEARCH statement	245
8.52.	SEND statement	246
8.53.	SET statement	247
8.54.	SORT statement	250
8.55.	START statement	251
8.56.	STOP statement	252
8.57.	STRING statement	254
8.58.	SUBTRACT statement	255
8.59.	SUPPRESS statement	257
8.60.	TERMINATE statement	258
8.61.	TRANSFORM statement	259
8.62.	UNLOCK statement	260
8.63.	UNSTRING statement	261
8.64.	USE statement	262
8.65.	VALIDATE statement	264
8.66.	WRITE statement	265
8.67.	XML GENERATE statement	267
8.68.	XML PARSE statement	269
9.	Intrinsic functions	271
9.1.	ABS function	271
9.2.	ACOS function	272
9.3.	ANNUITY function	273
9.4.	ASIN function	274
9.5.	ATAN function	275
9.6.	BOOLEAN-OF-INTEGERS function	276

Contents

9.7.	BYTE-LENGTH function	277
9.8.	CHAR function	278
9.9.	CHAR-NATIONAL function	279
9.10.	COMBINED-DATETIME function	280
9.11.	CONCATENATE function	281
9.12.	CONTENT-LENGTH function	282
9.13.	CONTENT-OF function	283
9.14.	COS function	284
9.15.	CURRENCY-SYMBOL function	285
9.16.	CURRENT-DATE function	286
9.17.	DATE-OF-INTEGERS function	287
9.18.	DATE-TO-YYYYMMDD function	288
9.19.	DAY-OF-INTEGERS function	289
9.20.	DAY-TO-YYYYDDD function	290
9.21.	DISPLAY-OF function	291
9.22.	E function	292
9.23.	EXCEPTION-FILE function	293
9.24.	EXCEPTION-FILE-N function	294
9.25.	EXCEPTION-LOCATION function	295
9.26.	EXCEPTION-LOCATION-N function	296
9.27.	EXCEPTION-STATEMENT function	297
9.28.	EXCEPTION-STATUS function	298
9.29.	EXP function	299
9.30.	EXP10 function	300
9.31.	FACTORIAL function	301
9.32.	FORMATTED-CURRENT-DATE function	302
9.33.	FORMATTED-DATE function	303
9.34.	FORMATTED-DATETIME function	304
9.35.	FORMATTED-TIME function	305
9.36.	FRACTION-PART function	306
9.37.	HIGHEST-ALGEBRAIC function	307
9.38.	INTEGER function	308
9.39.	INTEGER-OF-BOOLEAN function	309
9.40.	INTEGER-OF-DATE function	310
9.41.	INTEGER-OF-DAY function	311
9.42.	INTEGER-OF-FORMATTED-DATE function	312
9.43.	INTEGER-PART function	313
9.44.	LENGTH function	314
9.45.	LENGTH-AN function	315
9.46.	LOCALE-COMPARE function	316
9.47.	LOCALE-DATE function	317
9.48.	LOCALE-TIME function	318
9.49.	LOCALE-TIME-FROM-SECONDS function	319
9.50.	LOG function	320

Contents

9.51.	LOG10 function	321
9.52.	LOWER-CASE function	322
9.53.	LOWEST-ALGEBRAIC function	323
9.54.	MAX function	324
9.55.	MEAN function	325
9.56.	MEDIAN function	326
9.57.	MIDRANGE function	327
9.58.	MIN function	328
9.59.	MOD function	329
9.60.	MODULE-CALLER-ID function	330
9.61.	MODULE-DATE function	331
9.62.	MODULE-FORMATTED-DATE function	332
9.63.	MODULE-ID function	333
9.64.	MODULE-PATH function	334
9.65.	MODULE-SOURCE function	335
9.66.	MODULE-TIME function	336
9.67.	MONETARY-DECIMAL-POINT function	337
9.68.	MONETARY-THOUSANDS-SEPARATOR function	338
9.69.	NATIONAL-OF function	339
9.70.	NUMERIC-DECIMAL-POINT function	340
9.71.	NUMERIC-THOUSANDS-SEPARATOR function	341
9.72.	NUMVAL function	342
9.73.	NUMVAL-C function	343
9.74.	NUMVAL-F function	344
9.75.	ORD function	345
9.76.	ORD-MAX function	346
9.77.	ORD-MIN function	347
9.78.	PI function	348
9.79.	PRESENT-VALUE function	349
9.80.	RANDOM function	350
9.81.	RANGE function	351
9.82.	REM function	352
9.83.	REVERSE function	353
9.84.	SECONDS-FROM-FORMATTED-TIME function	354
9.85.	SECONDS-PAST-MIDNIGHT function	355
9.86.	SIGN function	356
9.87.	SIN function	357
9.88.	SQRT function	358
9.89.	STANDARD-COMPARE function	359
9.90.	STANDARD-DEVIATION function	360
9.91.	STORED-CHAR-LENGTH function	361
9.92.	SUBSTITUTE function	362
9.93.	SUBSTITUTE-CASE function	363
9.94.	SUM function	364

Contents

9.95.	TAN function	365
9.96.	TEST-DATE-YYYYMMDD function	366
9.97.	TEST-DAY-YYYYDDD function	367
9.98.	TEST-FORMATTED-DATETIME function	368
9.99.	TEST-NUMVAL function	369
9.100.	TEST-NUMVAL-C function	370
9.101.	TEST-NUMVAL-F function	371
9.102.	TRIM function	372
9.103.	UPPER-CASE function	373
9.104.	VARIANCE function	374
9.105.	WHEN-COMPILED function	375
9.106.	YEAR-TO-YYYY function	376
Appendices		380
A. GNU Free Documentation License		382
A.1.	Preamble	382
A.2.	Applicability and definitions	383
A.3.	Verbatim copying	385
A.4.	Copying in quantity	386
A.5.	Modifications	387
A.6.	Combining documents	389
A.7.	Collections of documents	390
A.8.	Aggregation with independent works	391
A.9.	Translation	392
A.10.	Termination	393
A.11.	Future revisions of this license	394
A.12.	Relicensing	395
A.13.	Addendum: How to use this License for your documents	396
B. Language element lists		398
B.1.	Implementor-defined language element list	398
B.2.	Optional language element list	409
B.3.	Non-standard extension list	410

Foreword

This document describes the syntax of COBOL as supported by GnuCOBOL. It is hoped it will complement Gary Cutler and Vincent Coen's *GnuCOBOL Programmer's Guide* which (currently) does not document recent features added to GnuCOBOL. This document is also formatted in \LaTeX , so that everything looks a bit prettier.

The syntax diagrams were transcribed from GnuCOBOL's parsers. It thus replicates some unusual syntax rules and misses some syntax rules implemented outside the parser. For example, the obsolete identification division comment paragraphs are allowed in any order and the syntax of a file-control entry does not distinguish between SEQUENTIAL, INDEXED and RELATIVE organisations.

This is a draft and so has many flaws. If people find this document useful, I will try to fix these shortcomings.

Changelog

1. GnuCOBOL 3.2-early-dev

General

- **EXTFH**: added support for the external file handler (EXTFH) interface, letting users provide a custom file handler.
- **EXTFH**: the old OpenCOBOL EXTFH interface (built into GnuCOBOL with the `--with-seqra-extfh` and `--with-index-extfh` build options) has been marked as obsolete.

Configuration options

- **New compiler configurations**: realia for CA Realia II.
- **New compiler configuration options**: `reference-bounds-check` and `report-column-plus`.
- **New runtime configuration options**:
 - `keycheck` (check whether INDEXED file keys match the SELECT specification exactly).
- **New runtime configuration environment variables**: `IO_OPTIONS`, `IX_OPTIONS`, `LA_OPTIONS`, `LS_OPTIONS`, `RL_OPTIONS`, `SQ_OPTIONS` and `IO_x`, where `x` is any ASSIGN EXTERNAL name used in a program.

Compiler directives

- **SET directive**: added XFD phrase.

Environment division

- **ASSIGN clause**: allow "ASSIGN VARYING ...".
- **FILE-CONTROL entries**: added detection of the OS/VS-era phrases ACTUAL KEY, FILE-LIMIT, NOMINAL KEY, TRACK-AREA and TRACK-LIMIT.
- **I-O-CONTROL paragraph**: added detection of the OS/VS-era clauses CORE-INDEX, CYL-INDEX, CYL-OVERFLOW, EXTENDED-SEARCH, LOCK-HOLDING, MASTER-INDEX, PRINT-CONTROL, RECORD OVERFLOW, REORG-CRITERIA, WRITE-ONLY and WRITE-VERIFY.

Changelog

- **RERUN clause:** added detection of.

Procedure division

- **File I-O:** added support for multiple simultaneous file handlers.
- **File I-O:** allow file-handler to be configured for individual files or organizations using environment variables (or dictionary files).
- **File I-O:** added support for INDEXED files based on LMDB.
- **File I-O:** added initial support for ODBC and OCI as file-handlers.
- **File I-O:** changed file status for detecting bad data in line sequential files from 34 to 71.
- **File I-O (BDB):** fixed file status after OPEN OUTPUT on an OPTIONAL file.
- **File I-O (ISAM):** fixed bug involving START ... LESS EQUAL followed by READ PREVIOUS.
- **OPEN statement:** added detection of OS/VS-era input mode phrases.
- **SET statement:** added support for FH--FCD and FH--KEYDEF.
- **SUBTRACT statement:** fixed [bug #603](#), where subtracting an integer from a floating-point variable lead to incorrect results.
- **WRITE statement (indexed):** fixed bug involving file status 21.

2. GnuCOBOL 3.1-dev

General

- **C API (configuration):** added `cob_set_runtime_option` and `cob_get_runtime_option` for specifying the file handle which printer output or trace output should go to.
- **cobc command-line options:** added `-fcallfh` (and COBOL-IT alias `-use-extfh`).
- **cobc command-line options:** added `-fdump` to dump data of all modules onabend of program.
- **cobc command-line options:** disabled `-fif-cutoff` pending removal.
- **cobc command-line options:** added `-fno-remove-unreachable` to disable removal of unreachable code.
- **cobc command-line options:** `-debug` no longer implies `-ftrace` ([bug #449](#)).

Changelog

- **cobc command-line options:** added -O0 for disabling optimisations ([feature request #255](#)).
- **cobc command-line options:** replaced -tsymbols by -f(no)-tsymbols.
- **cobc command-line options:** added -fno-theader, -fno-tmessages and -fno-tsource to suppress parts of listings.
- **cobc command-line options:** added -Wno-dialect to suppress dialect-specific warnings.
- **cobc command-line options:** added -Wother to enable miscellaneous warnings.
- **cobc command-line options:** added -fgen-c-line-directives and -fgen-c-labels to enable generation of labels and directives indicating where generated C code came from in the original COBOL source.
- **cobcrun command-line options:** renamed -runtime-conf to -runtime-config.
- **COBOL-85 NIST testsuite:** tests can now be downloaded from Sourceforge when the NIST website is unavailable ([bug #563](#)).
- **COBOL-85 NIST testsuite:** tests can now be performed individually by running make in the test module directory.
- **Copyfiles:** fixed segfault when GnuCOBOL tried and failed to open a copyfile ([bug #458](#)).
- **Copyfiles:** GnuCOBOL now checks for recursive copyfiles ([bug #467](#)).
- **Debugging:** Added debug logging for compiler developers (--enable-debug-log ./configure option).
- **Dependencies:** GnuCOBOL now (optionally) depends on libxml2 and cJSON (for XML and JSON I-O).
- **DJGPP support:** added minimal support for DJGPP.
- **Error messages:** runtime error messages now use translated versions (depending on a user's language).
- **Error messages:** added checks for version control conflict markers in source code.
- **Error messages:** fixed [bug #586](#), where GnuCOBOL would cause a buffer overflow when outputting an error message with a very long literal.
- **Expressions:** fixed [bug #431](#), where decimal constants were not initialised in INITIAL programs.
- **IBM i support:** fixed segfault involving filepaths
- **Listings:** fixed the substitution of tokens in quotes in copyfiles ([bug #494](#)).

Changelog

- **Listings:** fixed some cases of cross-reference listings containing wrong line numbers.
- **Listings:** now require the size of a listing page to be at least 20 lines.
- **Listings:** fixed segfault in listing programs with large REPLACE arguments ([bug #515](#)).
- **MinGW support:** added script for generating binaries using MinGW.
- **Modules:** enabled running of modules generated with older versions of GnuCOBOL ([feature request #239](#)).
- **National literals:** added Fujitsu NC national character literals ([feature request #352](#)).
- **Nested programs:** fixed [bug #435](#), where identifiers in the containing program where not propagated to the contained program, causing a segfault.
- **Nested programs:** fixed [bug #587](#), where a memory error occurred when a program was nested inside a program with the same name.
- **Numeric literals:** added support for HP COBOL octal literals (%...).
- **OrangeC support:** added minimal support for OrangeC.
- **Recursion:** fixed [bug #222](#), where the return value was lost when returning from a recursive call.
- **Reserved words:** allow users to make default reserved words aliases for other words.
- **Screen section testsuite:** added a manual testsuite for screen section functionality ([patch #26](#)).
- **Signals:** added error message for SIGFPE (see [bug #434](#)).
- **Solaris support:** fixed errors and warnings when compiling in Solaris 10.
- **Tracing:** added new options for controlling tracing (see [feature request #242](#)).
- **User-defined words:** increased default maximum user-defined length to 63 characters (per COBOL 202X draft standard).

Configuration options

- **IBM dialect:** reserved words updated to Enterprise COBOL V6.2.
- **incorrect-conf-sec-order option:** changed to "ok" from "error" in mf and default dialects.
- **MF dialect:** added ARGUMENT-NUMBER and ARGUMENT-VALUE as reserved words.

Changelog

- **New compiler configuration options:** binary-comp-1, call-convention-mnemonic, call-convention-linkage, continue-after, display-special-fig-consts, free-redefines-position ([feature request #211](#)), hp-octal-literals, goto-entry, line-col-zero-default, national-character-literals move-figurative-space-to-numeric, move-non-numeric-lit-to-numeric-is-zero, missing-statement, perform-without-varying-by, record-delimiter, record-delim-with-fixed-recs, records-mismatch-record-clause, screen-section-rules, sequential-delimiters, symbolic-constant, xml-generate-extra-phrases, zero-length-literals.
- **New runtime configuration options:**
 - col_just_lrc (for enabling/disabling the LEFT/RIGHT/CENTER phrases of the report COLUMN clause),
 - printer and display_print_pipe (for specifying what command should be executed before DISPLAY UPON PRINTER; similar to Micro Focus' COBPRINTER option);
 - display_print_file (name of file which DISPLAY UPON PRINTER will append its output to), display_punch_file (name of file to created upon first DISPLAY UPON SYSPUNCH),
 - mouse_flags (to let mouse clicks move the cursor in ACCEPT),
 - mouse_interval (maximum time between mouse press and mouse release events for them to recognised as a click),
 - trace_format,
 - dump_file and
 - dump_width.
- **Registers:** disabling a register also removes the reserved word with the same name, if it exists ([feature request #278](#)).
- **Registers:** added COL and LIN.
- **Runtime configuration options:** use the system's file path separator instead of always using : (so the path separator is now ; on Windows). (See [bug #610](#).)

Compiler directives

- **COPY statement:** added support for copybook and library names containing periods as text-names. For example, "COPY copybook.cpy." is now accepted. See [feature request #344](#).
- **Fujitsu @OPTIONS directive:** added recognition of.
- **New directives:** ADDRSV, ADDSYN, MAKESYN, OVERRIDE and REMOVE ([feature request #210](#)); CALLFH and COMP1.
- **SET directive:** added XFD phrase.

Changelog

Identification division

- **OPTIONS paragraph:** added support for ARITHMETIC clause.

Environment division

- **ALTERNATE RECORD KEY clause:** implemented split keys (SOURCE IS; [feature request #23](#)) and sparse keys (SUPPRESS WHEN; [feature request #281](#)).
- **ALPHABET clause:** added recognition of more pre-defined alphabets.
- **ASSIGN clause:** re-enabled use of linkage section or BASED items ([bug #421](#)).
- **ASSIGN clause:** added SYSPCH and SYSPUNCH device-names for writing to a punch file (this is not related to CARD-PUNCH).
- **ASSIGN clause:** added device-names S01 through to S05.
- **CALL-CONVENTION phrase:** added C and PASCAL (see [feature request #311](#)).
- **CLASS phrase:** added recognition of ALPHANUMERIC/NATIONAL and alphabet phrases.
- **CLASS phrase:** fixed [bug #566](#), where class-names were not inherited by nested subprograms.
- **CRT STATUS phrase:** improved syntax checks.
- **CURRENCY phrase:** changed to emit error when CURRENCY SIGN other than “\$” is entered.
- **CURRENCY phrase:** improved error messages for invalid currency signs.
- **CURSOR phrase:** fixed [bug #579](#), where the CURSOR data item was not used for the initially positioning the cursor in an ACCEPT statement.
- **CURSOR phrase:** fixed for data items of length 4 characters.
- **EXTERN clause:** improved error messages ([bug #446](#)).
- **FILE STATUS clause:** added detection of VSAM secondary status identifier ([feature request #51](#)).
- **I-O-CONTROL paragraph:** added detection of APPLY COMMIT clause (from the COBOL 202X draft standard).
- **OBJECT-COMPUTER paragraph:** added recognition of NATIONAL collating sequences.
- **OCCURS clause:** correctly implemented nested OCCURS DEPENDING tables (ODOSLIDE).
- **PASSWORD clause:** added detection of.

Changelog

- **RECORD clause:** fixed behaviour in the IBM dialect when the maximum record length does not match the length in the RECORD clause.
- **RECORD DELIMITER phrase:** added support for BINARY-SEQUENTIAL and LINE-SEQUENTIAL phrases.
- **RECORD DELIMITER phrase:** improved syntax checks (see [bug #442](#)).
- **RECORD KEY clause:** implemented sparse keys (SOURCE IS; [feature request #23](#)).
- **TOP phrase:** added as mnemonic for ADVANCING PAGE.

Data division

- **88-level items:** improved error messages when used in the wrong places.
- **ANY LENGTH clause:** fixed [bug #487](#), where literals moved to ANY LENGTH items were incorrectly truncated.
- **ANY LENGTH clause:** fixed [bug #511](#), where comparisons treated ANY LENGTH items as having just one character.
- **ANY LENGTH clause:** fixed segfault when referencing linkage-section ANY LENGTH items in the first compiled program.
- **COLUMN clause:** “COL +1” and similar clauses are no longer allowed; there must now be a space between the sign and the number.
- **COLUMN clause (screen):** added POS as synonym for COLUMN.
- **LINE clause:** “LINE +1” and similar clauses are no longer allowed; there must now be a space between the sign and the number.
- **LINE clause (report):** allow LINES instead of LINE.
- **PICTURE clause:** added recognition of PIC 1(n).
- **PICTURE clause:** improved error messages.
- **PICTURE clause:** added recognition of LOCALE phrase.
- **RECORD VARYING clause:** fixed [bug #441](#), where an FD could be declared as RECORD VARYING, but have records of only one size.
- **Screen section:** made some COBOL words context-sensitive to screen section.
- **Screen description:** the rules on which clauses must be specified when now depend on the dialect (see [bug #382](#)).
- **SYNCHRONIZED clause:** deactivated RIGHT phrase, pending correct implementation (previously SYNCHRONIZED RIGHT was the same as SYNCHRONIZED LEFT).

Changelog

- **USAGE clause:** added option to make COMP-1 mean a 16-bit signed integer ([feature request #272](#)).
- **USAGE clause:** added recognition of BIT and COMP-0.
- **USAGE clause:** added COMP-N.
- **VOLATILE clause:** added recognition of.

Procedure division

- **ACCEPT statement (environment):** added recognition of INPUT STATUS and TERMINAL-INFO and SYSTEM-INFO phrases.
- **ACCEPT statement (screen):** fixed ACCEPT WITH UPDATE and not working ([bug #423](#)).
- **ACCEPT statement (screen):** fixed [bug #426](#), where backspacing at the start of a field moved the cursor to the second-to-last character of the preceding field.
- **ACCEPT statement (screen):** added detection of CONTROL KEY clause.
- **ACCEPT statement (screen):** allow numeric-edited fields to contain spaces (see [bug #491](#)).
- **ACCEPT statement (screen):** fixed error when filling in a one character field with insert mode on ([bug #498](#)).
- **ACCEPT statement (screen):** pressing the home key, or end key, now toggles between going to the start of, or end of, the field and the cursor's original position.
- **ACCEPT statement (screen):** added WITH NO BELL clause.
- **ACCEPT statement (screen):** added support for Microsoft COBOL position specifiers.
- **ACCEPT statement (screen):** removed silent minimum of 500 ms from TIME-OUT phrase ([feature request #196](#)).
- **ACCEPT statement (temporal):** fixed [bug #469](#), where ACCEPT FROM DAY was off by -1 .
- **ALLOCATE statement:** added recognition of LOC clause.
- **CALL statement:** fixed ON EXCEPTION not working properly with -fstatic ([bug #462](#)).
- **CALL statement:** added LINKAGE phrase ([feature request #311](#)).
- **CANCEL statement:** fixed segfault in recursive programs ([bug #98](#)).
- **Conditions:** strengthened syntax checks for abbreviated conditions.

Changelog

- **Conditions:** improved error messages ([bug #567](#)).
- **CONTINUE (after):** added support for (from COBOL 202X draft standard). See [feature request #354](#).
- **DISPLAY statement (printer):** allow for redirecting DISPLAY UPON PRINTER(-1) to files based on the runtime configuration.
- **DISPLAY statement (screen):** fixed DISPLAY LOW-VALUE not setting position of cursor for next DISPLAY statement ([bug #423](#)).
- **DISPLAY statement (screen):** added configuration option to disable Micro Focus' special behaviour with some figurative constants (see [bug #423](#)).
- **DISPLAY statement (screen):** fixed [bug #428](#), where DISPLAY ALL "x" WITH SIZE only displayed "x" once.
- **DISPLAY statement (screen):** added support for Microsoft COBOL's position specifiers and ERASE phrase.
- **ENTRY statement:** added LINKAGE phrase ([feature request #311](#)).
- **Error messages:** runtime error messages now include "error: " after the fil before the actual error message.
- **EVALUATE statement:** added partial support for WHEN clauses not containing any statements, which transfer control to the next WHEN clause containing a statement, if any (so the behaviour is like C's switch statement).
- **Exceptions:** the exception status is now only reset by SET LAST EXCEPTION TO OFF.
- **Exceptions:** fixed EC-SIZE-OVERFLOW being raised when EC-SIZE-ZERO-DIVIDE is active ([bug #223](#)).
- **File I-O:** fixed [bug #457](#), where file status 30 was set instead of the correct permanent error status (34, 35 or 37).
- **File I-O:** added support for VBISAM 2.1.1.
- **File I-O:** error messages from libcob now include the file I-O statement that caused them, when known.
- **Floating-point arithmetic:** fixed incorrect SIZE ERROR exceptions (see [bug #470](#)).
- **Floating-point arithmetic:** fixed [bug #478](#), where calculations with the SIZE ERROR phrase have different results to calculations without it.
- **Floating-point arithmetic:** now activate the SIZE ERROR handler when a floating-point variable is set to a non-finite value (see [bug #122](#)).

Changelog

- **GO TO statement:** added non-standard ENTRY phrase extension.
- **INQUIRE statement:** added recognition of.
- **JSON GENERATE statement:** added (patch #45).
- **JSON PARSE statement:** added recognition of.
- **LENGTH OF phrase:** fixed error message when used on group fields (bug #175).
- **LENGTH OF phrase:** allow non-numeric literals.
- **MODIFY statement:** added recognition of.
- **MOVE statement:** added options to interpret moving non-numeric values to numeric items as moving zero to numeric items.
- **MOVE statement:** improved detection of overlapping MOVES when using reference modification.
- **MOVE statement:** fixed remaining cases of bug #220, involving numeric-edited items.
- **MOVE statement:** fixed incorrect truncation of COMP-4 items.
- **PERFORM statement:** added check for non-zero item in BY phrase (feature request #268).
- **PERFORM statement:** allow the BY phrase to be omitted, which is the same as specifying BY 1 (feature request #158).
- **PERFORM statement (inline):** added partial support for inline PERFORM statements not containing any statements.
- **Procedure division header:** added recognition of MEMORY SIZE phrase.
- **Procedure division header:** added LINKAGE phrase (feature request #311).
- **RAISE statement:** added recognition of.
- **READ statement (relative):** fixed file status on end of file.
- **Reference modification:** improved out-of-bounds error message (bug #445).
- **SORT statement:** added sorting of tables without keys (patch #42).
- **SORT statement:** added recognition of more COLLATING SEQUENCE phrases.
- **OPEN statement:** fail with file status 39 when the first record of an indexed file is larger than specified in the FD.

Changelog

- **OPEN statement:** fixed segfault when attempting to open a file in a non-existent folder ([bug #520](#)).
- **OPEN statement:** added ACUCOBOL locking clauses.
- **READ statement (indexed):** fail with file status 43 when the read record is larger than specified in the FD.
- **Reference modification:** added warnings for invalid reference modifications involving variables.
- **Report writer:** implemented, including features from COBOL 2002 and the IBM Report Writer.
- **REWRITE statement (indexed):** now sets file status to 02 when duplicate key is detected.
- **SET statement:** fixed [bug #225](#), where an invalid SET statement caused an error saying “invalid MOVE statement”.
- **SET statement:** fixed [bug #543](#).
- **STOP statement:** fixed the NORMAL phrase causing a compilation error ([bug #433](#)).
- **USE statement (exception):** improved recognition of.
- **VALIDATE statement:** added recognition of.
- **WRITE statement:** fixed compilation error when writing to a GLOBAL file in a sub-program ([bug #87](#)).
- **WRITE statement (indexed):** now sets file status to 02 when duplicate key is detected.
- **XML GENERATE statement:** added ([patch #41](#)).
- **XML PARSE statement:** added recognition of.

Intrinsic functions

- **CONTENT-OF function:** added.
- **CONTENT-LENGTH function:** added.
- **LENGTH, BYTE-LENGTH and LENGTH-AN functions:** now forbid numeric arguments.
- **MOD function:** fixed EC-SIZE-ZERO-DIVIDE being raised instead of EC-ARGUMENT-FUNCTION when zero was provided as an argument.

Changelog

- **REM function:** fixed EC-SIZE-ZERO-DIVIDE being raised instead of EC-ARGUMENT-FUNCTION when zero was provided as an argument.
- **X"91" function:** corrected number of arguments from 2 to 3 ([bug #527](#)).
- **WHEN-COMPILED function:** fixed timezone, which was missing its sign and contained nonsense when negative ([bug #436](#)).

Built-in subprograms

- **C\$GETPID:** fixed wrong process ID being returned after forking ([bug #451](#)).
- **C\$SLEEP:** if the requested time is too large, sleep for the maximum possible time instead of not at all.
- **CBL_GET_CSR_POS:** fixed wrong position being returned after ACCEPT statement ([bug #483](#)).
- **CBL_READ_KBD_CHAR:** this now works ([bug #500](#)).
- **SYSTEM:** added workaround of buggy system() implementation on Windows, which removes leading and trailing quotes.

3. GnuCOBOL 2.2

This list tracks changes made from 23 November 2013. This excludes many changes made in 2009–2013 which would be pertinent to those upgrading from a 2009 build of OpenCOBOL 1.1 found in many package repositories.

General

- **64-bit numbers:** fixed bugs in handling of 64-bit numbers (e.g. [bug #229](#)).
- **ACUCOBOL windows:** added detection of ACUCOBOL's window/message box GUI syntax.
- **C API (data):** added several functions for getting and setting cob_field items.
- **C API (files):** added cob_file_external_addr, cob_file_malloc and cob_file_free.
- **C API (screen):** added several functions from Micro Focus' C to COBOL API: cob_display_text, cob_sys_get_char, cob_get_char, cob_get_text, cob_display_formatted_text, cob_sys_get_csr_pos, cob_sys_set_csr_pos, cobmove, cobaddstrc, cobprintf and cobgetch (feature requests [#148](#) and [#187](#)).
- **C API (signals):** added cob_raise to send signal to signal handlers.
- **C compiler support:** fixed errors in compilers without designated initializers.

Changelog

- **cobc command-line options:** added -O3 to enable more optimisations.
- **cobc command-line options:** added -Wfatal-error to make the compiler abort on the first error.
- **cobc command-line options:** added -Wpossible-overlap to warn items that *may* overlap (-Woverlap only warns if items definitely overlap).
- **cobc command-line options:** added -fmax-errors to set number of errors at which the compiler aborts.
- **cobc command-line options:** added -fwinmain to output WinMain instead of main ([feature request #194](#)).
- **cobc command-line options:** added -t and -T for complete listing support (-t for 80-characters wide listings and -T for 132-characters wide) which includes cross-references (thanks to Dave Pitts).
- **cobc command-line options:** added -vvv (like -vv but passes verbose option to the linker as well) and -### (like -v but commands are not executed).
- **cobc command-line options:** allow -, i.e. stdin, as a source file.
- **COBOL-85 NIST testsuite:** tests now refer to \$COBC, \$COBCRUN and \$COBCRUN_DIRECT environment variables instead of directly calling cobc and cobcrun, allowing the testsuite to run in conjunction with tools like valgrind.
- **COBOL-85 NIST testsuite:** tests for obsolete feature flagging are now executed, if possible.
- **Comments:** added ACUCOBOL comments: \$ as synonym for * in indicator area and | as synonym for floating comment indicator *>.
- **Communication facility:** added detection of communication facility syntax.
- **configure:** added useful error message when help2man, bison and flex are missing when they are needed.
- **curses:** fixed compilation errors when configured without curses ([bug #90](#)).
- **Error messages:** error messages are now lowercase, in line with the GNU Coding Standards ([bug #198](#)).
- **Error messages:** segfaults in the compiler now cause an error message to be displayed.
- **Error messages:** replaced instances of “ODO” by the clearer “OCCURS DEPENDING ON”.

Changelog

- **Expressions:** resolve constant expressions and optimise constant decimals at compile time.
- **Expressions:** added support for IBM OS/VS COBOL's arithmetic.
- **Expressions:** improved error messages for malformed expressions.
- **Indicators:** invalid indicators no longer cause compilation to immediately terminate ([feature request #126](#)).
- **Information:** output compiler version used to build GnuCOBOL and any mathematical libraries used ([feature request #169](#)).
- **Information:** output what a reserved word is an alias for in the --list-reserved output ([feature request #214](#)).
- **Manpage:** added manpage generation and installation.
- **Nested programs:** Nested programs no longer need to have END PROGRAM.
- **National literals:** added basic support for national literals.
- **Numeric literals:** added ACUCOBOL numeric literals: B#... for binary, O#... for octal, and X#... and H#... for hexadecimal.
- **Literals:** fixed heap corruptions caused by uncommon literals ([bug #195](#)).
- **Literals:** allow concatenation of literal and Boolean literals.
- **Memory management:** all memory belonging to the parsers and lexers is freed upon a compiler abend.
- **Memory management:** fixed memory leaks due to recursive CALLs.
- **Microsoft Visual C++:** output when compiling with cl.exe is now filtered and temporary files are deleted.
- **MinGW:** fixed use of wrong directory separator.
- **Signals:** removed error message on SIGPIPE.
- **Signals:** added error message for SIGBUS.
- **Translations:** updated, with new support for German and Italian.
- **User-defined functions:** function definitions must now end with END FUNCTION.
- **User-defined functions:** function definitions may no longer be nested in programs ([bug #255](#)).
- **Windows support:** allow linking with asm files.

Changelog

- **Windows support:** added support for DISAM in the batch file which creates distributables.
- **Windows support:** fixed environment-setting batch files not working with Microsoft Visual Studio 2017.
- **Windows support:** fixed 64-bit environment-setting batch files not checking the correct directories for binaries and libraries.

Configuration options

- **Deleted compiler configuration options:** eject-statement, cobol85-reserved.
- **New compiler configurations:** all dialects have been split into standard and strict dialects, with strict dialects maintaining source compatibility with the dialect's compiler(s).
- **New compiler configurations:** acu for ACUCOBOL, cobol2014 for COBOL 2014, rm for RM-COBOL, xopen for X/Open.
- **New compiler configuration options:** accept-display-extensions, accept-update, accept-auto, acu-literals, arithmetic-osvs, call-overflow, console-is-crt, constant-01, constant-78, constant-folding, define-constant-directive, hexadecimal-boolean, hexadecimal-national-literals, incorrect-conf-sec-order, intrinsic-function, listing-statements, literal-length, move-figurative-constant-to-numeric, move-figurative-quote-to-numeric, move-ibm, national-literals, no-echo-means-secure, not-exception-before-exception, numeric-boolean, numeric-literal-length, numeric-value-for-edited-item, pic-length, program-name-redefinition, program-prototypes, reference-out-of-declaratives ([feature request #179](#)), register, renames-uncommon-levels, reserved, reserved-words, stop-identifier, system-name, title-statement, use-for-debugging, word-length ([feature request #43](#)).
- **Registers:** compiler configurations can now specify all the registers to generate.
- **Registers:** added registers not yet implemented by GnuCOBOL as reserved words.
- **Renamed compiler configuration options:** debugging-line to debugging-mode, relaxed-syntax-check to relax-syntax-checks.
- **Reserved words:** compiler configurations can now specify all the reserved words and context-sensitive words permitted.
- **Reserved words:** compiler configurations can now specify whether a reserved word is an alias for another reserved word.
- **Runtime configuration:** added ability to configure some libcob features at runtime.
- **Support options:** options which specify if a feature is supported can now take a "+" before their argument to indicate it takes effect only if the current level of support is less strict than "ok".

Changelog

Compiler directives

- **\$ indicator character:** added \$ as an indicator for compiler directive lines.
- **>>IF directive:** fixed [bug #263](#), where nested >>IF directives were not handled correctly.
- **New constants:** GCCOMP, GNUCOBOL.
- **New directives:** >>CALL-CONVENTION, >>LISTING, >>PAGE.
- **New directives (detection only):** *CBL, *CONTROL, TITLE.
- **New >>SET phrase:** SOURCEFORMAT.

Identification division

- **Comment paragraphs:** fixed invalid parsing of quote characters inside comment paragraphs ([bug #297](#)).
- **FUNCTION-ID:** added checks for redefinition of function-names.
- **INITIAL phrase:** fixed premature deallocation of INITIAL programs ([bug #52](#)).
- **OPTIONS paragraph:** added with implementation of DEFAULT ROUNDED MODE and ENTRY-CONVENTION phrases and recognition of INTERMEDIATE ROUNDING phrase.
- **PROGRAM-ID:** added checks for redefinition of program-names.
- **PROGRAM-ID phrases:** permit INITIAL or RECURSIVE before COMMON ([bug #244](#)).
- **Program/function-names:** warn if program/function-names contain spaces.

Environment division

- **ASSIGN clause:** missing ASSIGN clauses are now detected at compile-time.
- **ASSIGN clause:** added PRINTER and PRINTER-1 device-names for writing to a printer.
- **ASSIGN clause:** added CARD-PUNCH, CARD-READER, CASSETTE, INPUT, INPUT-OUTPUT, MAGNETIC-TAPE and OUTPUT device-names for line sequential devices.
- **ASSIGN clause:** temporarily prohibit BASED and linkage items in ASSIGN USING due to [bug #421](#).
- **CALL-CONVENTION phrase:** statically calling functions with CALL-CONVENTION 74 no longer causes linker errors ([bug #316](#)).

Changelog

- **CURRENCY phrase:** fixed [bug #182](#), where a preceding SWITCH phrase caused an incorrect duplicate CURRENCY clause error.
- **File-control entry:** fixed [bug #71](#), where referring to a global constant caused an internal error.
- **File-control entry:** fixed [bug #331](#), where using an identifier in a file record qualified with the file's name caused an error.
- **FUNCTION phrase:** added checks for redefinition of function-(prototype-)names.
- **FUNCTION phrase:** compiler will no longer stop when it encounters a syntax error.
- **LOCK MODE clause:** fixed combination of LOCK MODE IS AUTOMATIC/MANUAL with LOCK ON MULTIPLE.
- **PROGRAM phrase:** added support for program-prototype-names.
- **SIGN clause:** improved syntax checks.
- **SWITCH phrase:** added check for duplicate on/off clauses ([bug #136](#)).
- **SWITCH phrase:** added new switch names: SWITCH-16 through to SWITCH-36 ([feature request #65](#)), "SWITCH 1" to "SWITCH 26" (and their aliases "SWITCH A" to "SWITCH Z"), UPSI-0 to UPSI-8 (equivalent to "SWITCH 0" to "SWITCH 8") and USW-0 to USW-31 (equivalent to "SWITCH 0" to "SWITCH 31").

Data division

- **78-level items:** strengthened syntax checks.
- **88-level items:** strengthened syntax checks.
- **ANY NUMERIC clause:** ANY NUMERIC items must now have PIC 9.
- **ANY LENGTH clause:** ANY LENGTH items may no longer be BY VALUE parameters (see [bug #219](#)).
- **ANY LENGTH clause:** ANY LENGTH items must now have PIC X or PIC N.
- **BLANK clause:** fixed [bug #143](#), where BLANK LINE/SCREEN did not colour line/screen.
- **BLANK WHEN ZERO clause:** added checks that BLANK WHEN ZERO is not specified with PICTURE clauses containing S.
- **Constant items:** expressions in VALUE clauses now permitted.
- **Data description:** added a maximum record length.

Changelog

- **Data description:** increased maximum size of non-indexed file record to 64 MiB (maximum size of an indexed file record is 65535 bytes).
- **ERASE clause:** fixed [bug #186](#), where ERASE EOL and ERASE EOS could be specified simultaneously.
- **FULL clause:** added warning for useless FULL clauses on numeric items ([feature request #209](#)).
- **HIGHLIGHT and LOWLIGHT clauses:** added checks that HIGHLIGHT and LOWLIGHT are not specified simultaneously.
- **Local-storage section:** fixed [bug #78](#), where local-storage items were initialised after file section items.
- **LOWLIGHT clause:** implemented.
- **OCCURS clause:** fixed internal compiler when used with SYNC ([bug #155](#)).
- **OCCURS clause:** allow KEY phrase and INDEXED phrase in any order.
- **OCCURS clause:** fixed [bug #167](#), where overly large numeric literals were accepted in the OCCURS clause.
- **OCCURS clause (depending):** require the minimum length to be less than the maximum length ([feature request #99](#)).
- **OCCURS clause (depending):** disabled nested OCCURS DEPENDING tables due to bugs.
- **OCCURS clause (screen-section):** require relative LINE/COLUMN clauses in OCCURS entries ([bug #83](#)).
- **OCCURS clause (unbounded):** added by Frank Swarbrick ([patch #50](#)).
- **PICTURE clause:** restricted number of permitted PICTURE strings ([bug #232](#)).
- **PICTURE clause:** improved checks of constant-names referenced in PICTURE strings.
- **RENAMES items:** strengthened syntax checks.
- **RESERVE clause:** allow the optional word AREAS.
- **Screen description:** permit figurative constants in screen items ([bug #108](#)).
- **TALLY special register:** added.
- **USAGE clause:** added ACUCOBOL's HANDLE phrases (see [feature request #77](#)).
- **VALUE clause:** VALUE clauses in REDEFINES entries now cause warnings, not errors, for compatibility.
- **Variable records:** added checks that the minimum size of a variable record is large enough to contain the record key.

Changelog

Procedure division

- **ACCEPT statement:** added ESCAPE as synonym for EXCEPTION.
- **ACCEPT statement:** permit clauses in any order.
- **ACCEPT statement:** allow WITH before every screen attribute clause.
- **ACCEPT statement:** entering control-C now terminates the program.
- **ACCEPT statement (screen):** fixed failed ACCEPTs caused by a buffer overflow.
- **ACCEPT statement (screen):** enhanced support for special keys (insert, tab, delete, alt-delete, etc.).
- **ACCEPT statement (screen):** fixed [bug #161](#) where screens terminated after entering a few characters in a field.
- **ACCEPT statement (screen):** added DEFAULT as synonym for UPDATE.
- **ACCEPT statement (screen):** ERASE and BLANK clauses in screens are now ignored ([bug #192](#)).
- **ACCEPT statement (screen):** fixed [bug #160](#) where ACCEPT statement LINE/COLUMN clauses did not work.
- **ACCEPT statement (screen):** fixed segfault on ACCEPT OMITTED ([bug #300](#)).
- **ACCEPT statement (screen):** added checks that screen attributes are not specified multiple times or after conflicting attributes.
- **ACCEPT statement (screen):** fixed some phrases not being recognised without being preceded by WITH ([bug #402](#)).
- **ACCEPT statement (screen):** fixed the backspace and delete keys not working and the insert key not toggling between insertion and overwriting.
- **ACCEPT statement (screen):** cursor now changes with insertion/overwrite mode (if supported by the terminal).
- **ACCEPT statement (screen):** a beep is emitted on attempts to *insert* data into a full field.
- **ADD statement (corresponding):** restricted to numeric items ([bug #235](#)).
- **ADD statement (table):** added detection of ADD TABLE.
- **Addition of COMP-3 numbers:** fixed bug where COMP-3 addition failed.
- **Addition of floating-point numbers:** fixed incorrect addition of floating-point numbers.

Changelog

- **CALL statement:** implemented [feature request #101](#), allowing more arguments to be provided.
- **CALL statement:** fixed behaviour when calling cancelled modules.
- **CALL statement:** added RETURNING NOTHING.
- **CALL statement:** the generation of C function declarations for static CALLs can now be disabled.
- **CALL statement:** added checks for static CALLs referring to C macros.
- **CALL statement:** warn if a literal containing the program-name contains spaces.
- **CALL statement:** added detection of NESTED phrase.
- **CANCEL statement:** fixed crash caused by cancelling a cancelled module.
- **Conditions:** restricted use of IS ([bug #321](#)).
- **Conditions:** added warnings for always true/false conditions (including the reason why it is always true/false).
- **DESTROY statement:** added detection of DESTROY.
- **DISPLAY statement:** permit clauses in any order.
- **DISPLAY statement:** allow WITH before every screen attribute clause.
- **DISPLAY statement (screen):** fixed bug where EC-SCREEN exceptions did not trigger ON EXCEPTION handler ([bug #243](#)).
- **DISPLAY statement (screen):** fixed bugs in DISPLAY SPACES/ALL X"02"/ALL X"07".
- **DISPLAY statement (screen):** added checks that screen attributes are not specified multiple times or after conflicting attributes.
- **DISPLAY statement (screen):** DISPLAY OMITTED marked as unfinished; currently equivalent to DISPLAY LOW-VALUE.
- **DISPLAY statement (screen):** fixed some phrases not being recognised without being preceded by WITH ([bug #402](#)).
- **END DECLARATIVES phrase:** fixed [bug #88](#), where an erroneous unreachable code warning was emitted for code without a main procedure.
- **ENTRY statement:** suppress incorrect unreachable code warnings.
- **Exception handlers:** permit NOT ON EXCEPTION/END-OF-PAGE/etc. before ON EXCEPTION/END-OF-PAGE/etc.

Changelog

- **EXIT statement:** added extension RETURNING/GIVING clause for PROGRAM phrase.
- **File I-O:** added detection of and handling for error when no disc space is available for output files.
- **File I-O:** added RETRY and ADVANCING ON LOCK as pending features.
- **File I-O:** fixed detection of DISAM file handler.
- **FREE statement:** NULL addresses no longer cause an exception.
- **GOBACK statement:** added extension RETURNING/GIVING clause.
- **INITIALIZE statement:** fixed [bug #84](#), where literals could be passed to INITIALIZE.
- **INITIALIZE statement:** fixed [bug #287](#), where reference-modified group items were not treated like elementary items.
- **INSPECT statement:** fixed [bug #47](#), where clauses were permitted in invalid orders.
- **LENGTH OF phrase:** fixed [bug #89](#), where the length of REDEFINES item where calculated incorrectly.
- **LENGTH OF phrase:** fixed [bug #110](#), where LENGTH OF was not allowed in the UNTIL phrase of a PERFORM statement or in a VALUE clause.
- **MERGE statement:** added recognition of more COLLATING SEQUENCE phrases.
- **MOVE statement:** added more checks for overlapping MOVE statements.
- **MOVE statement:** fixed truncation of COMP numbers not conforming to the binary-truncate setting ([bug #69](#)).
- **MOVE statement:** fixed [bug #344](#), where trying to MOVE to a procedure-name caused a segfault.
- **MOVE statement:** added support for IBM's character-by-character MOVE.
- **PERFORM statement:** fixed [bug #368](#), where the compiler segfaulted when there was a PERFORM statement with an empty body and DEBUGGING MODE was specified.
- **Procedure division header:** fixed [bug #55](#), where a user-defined function without parameters failed to compile.
- **Procedure division header:** disabled the BY VALUE phrase, pending a working implementation.

Changelog

- **Procedure division header:** fixed [bug #349](#), where BY VALUE pointer parameters lead to code that couldn't be compiled by older versions of Microsoft Visual C++ (patched by Mario Matos).
- **Procedure division header:** RETURNING items must now be declared in the linkage section.
- **Procedure division header:** added RETURNING OMITTED.
- **Procedure division header:** added entry-convention specifiers.
- **Procedure division header:** now mandatory in function definitions (see [bug #271](#)).
- **Procedure division header:** CHAINING programs may no longer be called by other programs ([bug #354](#)), per the ACUCOBOL implementation.
- **Reference modification:** fixed [bug #146](#), where the length of reference-modified item in an OCCURS DEPENDING table was too long because it was assumed the OCCURS DEPENDING table was at its maximum size.
- **READ statement:** a failed second READ of a missing OPTIONAL file now results in a file status of 46, not 23.
- **REWRITE statement:** added REWRITE FILE ([feature request #170](#)).
- **Screen I-O:** added detection of situations which raise EC-SCREEN-LINE-NUMBER, EC-SCREEN-STARTING-COLUMN and EC-SCREEN-ITEM-TRUNCATED.
- **Screen I-O:** added support for the LINE 0 and COL 0 extensions.
- **Screen I-O:** added some ACUCOBOL synonyms (NO ECHO, OFF, REVERSED, REVERSE, etc.).
- **Screen I-O:** added detection of ACUCOBOL's non-standard clauses like TAB, NO-ECHO, STANDARD, BACKGROUND-HIGH, BACKGROUND-LOW, BACKGROUND-STANDARD and SIZE.
- **SEARCH statement (ALL):** fixed [bug #314](#), where SEARCH ALL with an empty OCCURS DEPENDING table did not exit as soon as possible.
- **Segment numbers:** added syntax checks.
- **SET statement (address):** disallowed changing address of non-01/77-level item ([bug #366](#)).
- **SET statement (attribute):** made HIGHLIGHT ON imply LOWLIGHT OFF and vice versa.
- **SET statement (exception):** added.

Changelog

- **SET statement (thread)**: added detection of ACUCOBOL extension.
- **STOP statement (identifier)**: added (see [bug #320](#)).
- **STOP statement (literal)**: fixed segfault.
- **STOP statement (thread)**: added detection of ACUCOBOL extension.
- **STRING statement**: strengthened syntax checks ([bug #259](#)).
- **SUBTRACT statement (corresponding)**: restricted to numeric items ([bug #235](#)).
- **SUBTRACT statement (table)**: added detection of SUBTRACT TABLE.
- **Tracing**: fixed [bug #216](#), where a segfault occurred with a program made from modules some of which had been compiled with tracing and physical CANCEL enabled and some of which hadn't.
- **UNSTRING statement**: fixed [bug #54](#), where the POINTER value was calculated incorrectly when the delimiter was longer than one character.
- **UNSTRING statement**: allow a literal to be the subject of an UNSTRING.
- **WRITE statement**: added WRITE FILE ([feature request #170](#)).

Intrinsic functions

- **New functions (ACUCOBOL)**: ABSOLUTE-VALUE (synonym for ABS).
- **New functions (COBOL 2014)**: FORMATTED-CURRENT-DATE, FORMATTED-DATE, FORMATTED-DATETIME, FORMATTED-TIME, INTEGER-OF-FORMATTED-DATE, TEST-FORMATTED-DATETIME.
- **ISO-8601-date-handling functions**: added extension SYSTEM-OFFSET as replacement for last optional argument.
- **ISO-8601-date-handling functions**: added EC-IMP-UTC-UNKNOWN if a time format ending in Z is provided but the timezone cannot be found.
- **LENGTH function**: added detection of PHYSICAL phrase.
- **RANDOM function**: fixed non-random number generation.

Built-in subprograms

- **CBL_GC_FORK**: added.
- **CBL_GC_PRINTABLE**: renamed from C\$PRINTABLE.
- **CBL_GC_WAITPID**: added.

Changelog

- **CBL_SET_CSR_POS**: added (feature requests #148 and #187).
- **CBL_READ_KBD_CHAR**: added (feature requests #148 and #187).

Changelog

1. Key

Element	Notes
Braces, { }	One element within the braces must be selected.
Brackets, []	One or zero elements within the brackets must be selected.
Vertical lines,	Each element may be selected once and in any order; if within braces, at least one element must be selected.
Ellipsis, ...	The preceding element may be repeated any number of times.
OPTIONAL-RESERVED-WORD <u>MANDATORY-RESERVED-WORD</u>	Mandatory reserved words in brackets are often used instead of optional reserved words to indicate an optional feature.
Deleted element	These elements were previously in the COBOL standard but have since been deleted. Their use is strongly discouraged.
Archaic element	These elements remain in the standard, but their use is considered poor style and is strongly discouraged.
Obsolete element	These elements are slated to be deleted from the standard. Their use is strongly discouraged.
X/Open extension	An extension which may have come from COBOL dialects by Micro Focus, IBM, Acucorp, Ryan-McFarland, Fujitsu or Microsoft.
GnuCOBOL-only extension	
Miscellaneous extension	
Unimplemented element	These elements are recognised by GnuCOBOL, but result in errors.

2. Language fundamentals

2.1. Lexical elements

2.1.1. COBOL words

2.1.2. User-defined words

2.1.3. Reserved words

2.1.4. Literals

Alphanumeric literals

Format 1 (standard)

$$\left\{ \begin{array}{l} \text{'character-1...'} \\ \text{"character-2..."} \end{array} \right\}$$

Format 2 (hexadecimal)

$$\left\{ \begin{array}{l} \text{X'hex-character-1...'} \\ \text{X"hex-character-2..."} \end{array} \right\}$$

Format 3 (null-terminated)

$$\left\{ \begin{array}{l} \text{Z'character-3...'} \\ \text{Z"character-4..."} \end{array} \right\}$$

2. Language fundamentals

Format 4 (raw-C-string)

$\left\{ \begin{array}{l} \text{L}' \text{character-5} \dots \text{' } \\ \text{L}'' \text{character-6} \dots \text{' } \end{array} \right\}$

Numeric literals

Format 5 (integer)

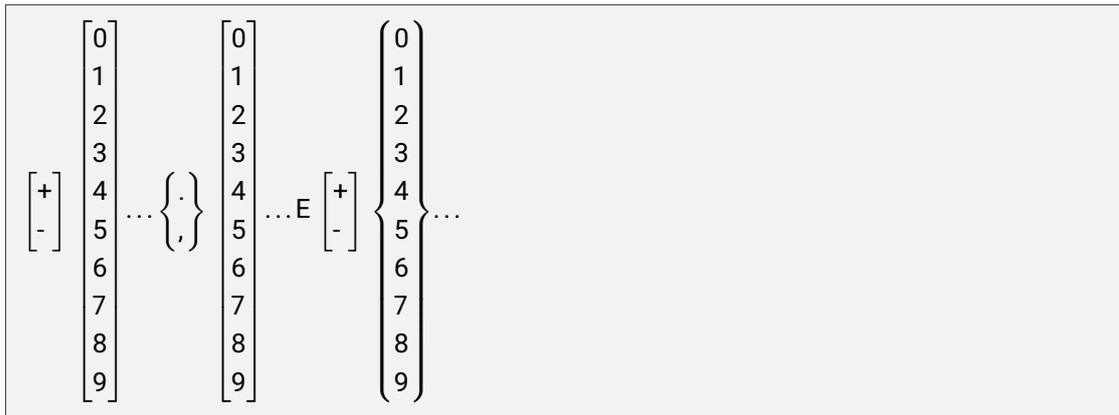
$\left[\begin{array}{l} + \\ - \end{array} \right] \left\{ \begin{array}{l} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ \dots \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{array} \right\}$

Format 6 (fixed-point)

$\left[\begin{array}{l} + \\ - \end{array} \right] \left\{ \begin{array}{l} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ \dots \left\{ \begin{array}{l} . \\ , \end{array} \right\} \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{array} \right\} \left\{ \begin{array}{l} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ \dots \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{array} \right\}$

Format 7 (floating-point)

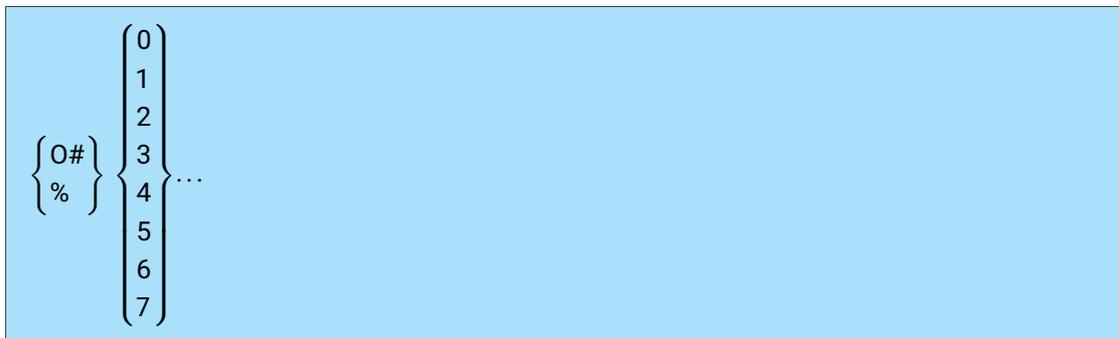
2. Language fundamentals



Format 8 (binary)



Format 9 (octal)



Format 10 (hexadecimal-number)

2. Language fundamentals



Format 11 (hexadecimal-string)



Boolean literals

Format 12 (standard)



Format 13 (hexadecimal)



National literals

Format 14 (standard)

2. Language fundamentals

$$\left\{ \begin{array}{l} N' [character-11] \dots' \\ N'' [character-12] \dots'' \\ NC' [character-13] \dots' \\ NC'' [character-14] \dots'' \end{array} \right\}$$

Format 15 (hexadecimal)

$$\left\{ \begin{array}{l} NX' [hex-character-5] \dots' \\ NX'' [hex-character-6] \dots'' \end{array} \right\}$$

Figurative constants

Format 16 (zero)

$$ALL \left\{ \begin{array}{l} ZERO \\ ZEROES \\ ZEROS \end{array} \right\}$$

Format 17 (space)

$$ALL \left\{ \begin{array}{l} SPACE \\ SPACES \end{array} \right\}$$

Format 18 (high-value)

$$ALL \left\{ \begin{array}{l} HIGH-VALUE \\ HIGH-VALUES \end{array} \right\}$$

2. Language fundamentals

Format 19 (low-value)

ALL {
LOW-VALUE
LOW-VALUES }

Format 20 (quote)

ALL {
QUOTE
QUOTES }

Format 21 (null)

ALL {
NULL
NULLS }

Format 22 (literal)

ALL *literal-1*

Format 23 (symbolic-character)

ALL *symbolic-character-1*

2. *Language fundamentals*

2.2. References

2.3. Expressions

2.3.1. Arithmetic expressions

Arithmetic expressions may contain the following operators:

Binary operators	Purpose	Precedence
+	addition	1
-	subtraction	1
*	multiplication	2
/	division	2
**	exponentiation	3
^	exponentiation	3
Unary operators		
+	no effect	4
-	multiplication by -1	4

Binary operators must have a numeric item or expression to both their left and right. Unary operators must have a numeric item or expression to their right only.

Operators with greatest precedence are evaluated first. If an expression contains multiple operators of equal precedence, they are evaluated from left to right.

Arithmetic expressions may contain arithmetic expressions surrounded by parentheses. These nested expressions are evaluated first, before any of the operators of the outer expression.

First symbol	Second symbol			
	Identifier or literal	Binary operator	Unary operator	()
Identifier or literal		✓		✓
Binary operator	✓		✓	✓
Unary operator	✓			✓
(✓		✓	✓
)		✓		✓

2. Language fundamentals

2.3.2. Concatenation expressions

$$\left\{ \begin{array}{l} \text{literal-1} \\ \text{concatenation-expression-1} \end{array} \right\} \& \text{literal-2}$$

2.3.3. Conditional expressions

Binary operators	Purpose	Precedence
AND	logical and	1
OR	logical or	2

Unary operator		
NOT	logical not	3

2.4. Concepts

2.4.1. Files

I-O status

Successful completion of an operation

- 00 Success
- 02 Success – duplicate
- 04 Success – incomplete
- 05 Success – optional
- 07 Success – no unit

Implementor-defined successful completion

End of file

- 10 End of file
- 14 Out of key range
- 21 Key invalid
- 22 Key exists
- 23 Key does not exist
- 24 Key boundary
- 30 Permanent I-O error
- 31 Inconsistent filename
- 34 Boundary violation
- 35 File does not exist
- 37 Permission denied
- 38 Closed with lock
- 39 Conflict attribute

2. Language fundamentals

- 41 File already open
- 42 File not open
- 43 Read not done
- 44 Record overflow
- 46 Read error
- 47 Input denied
- 48 Output denied
- 49 I-O denied
- 51 Record locked
- 57 I-O lineage
- 61 File sharing
- 71 **Bad character**
- 91 Not available

Organizations

Locking

2.4.2. Locales

2.4.3. Screens

2.4.4. User-defined functions

3. Compiler directives

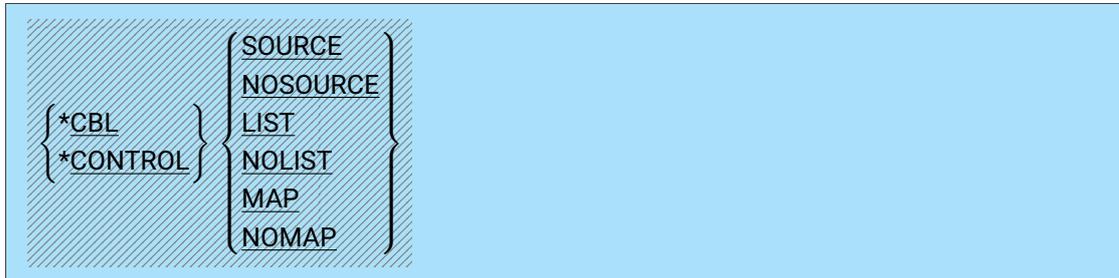
3.1. CALL-CONVENTION directive

<code>>>CALL-CONVENTION</code>	$\left\{ \begin{array}{l} \text{COBOL} \\ \text{EXTERN} \\ \text{STDCALL} \\ \text{STATIC} \end{array} \right.$
--------------------------------------	---

Syntax rules

General rules

3.2. *CONTROL statement

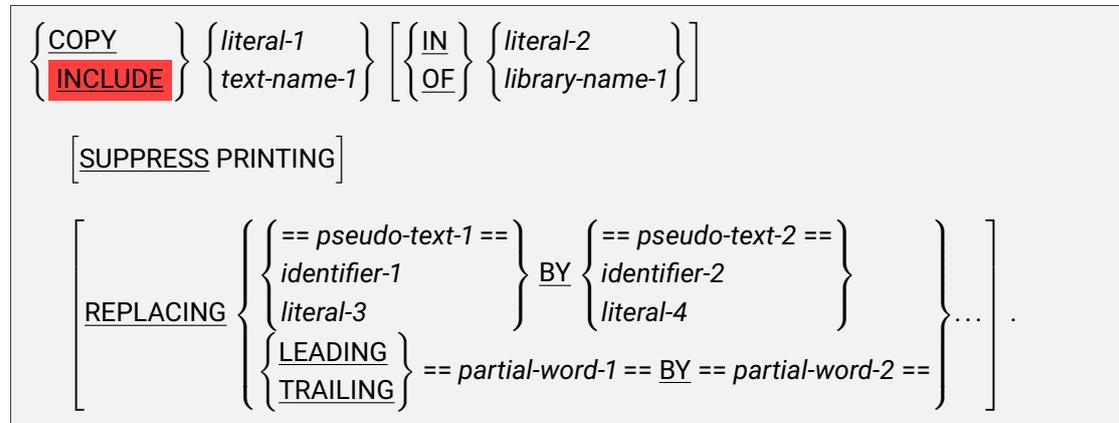


Syntax rules

General rules

3. Compiler directives

3.3. COPY statement



Syntax rules

General rules

3. Compiler directives

3.4. D directive

```
>>D source-text-1
```

Syntax rules

General rules

3. Compiler directives

3.5. DEFINE directive

$\left\{ \begin{array}{l} >> \\ \$ \end{array} \right\}$ DEFINE [CONSTANT] *compilation-variable-1* AS $\left\{ \begin{array}{l} \textit{literal-1} \\ \text{PARAMETER} \\ \text{OFF} \end{array} \right\}$ [OVERRIDE]

Syntax rules

General rules

3.6. DISPLAY directive

Format 1 (general)

$\left. \begin{array}{l} \text{>} \\ \$ \end{array} \right\} \text{DISPLAY source-text-1}$

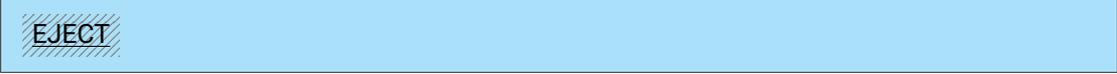
Format 2 (vcs)

$\text{\$DISPLAY VCS = version-string}$

Syntax rules

General rules

3.7. **EJECT statement**



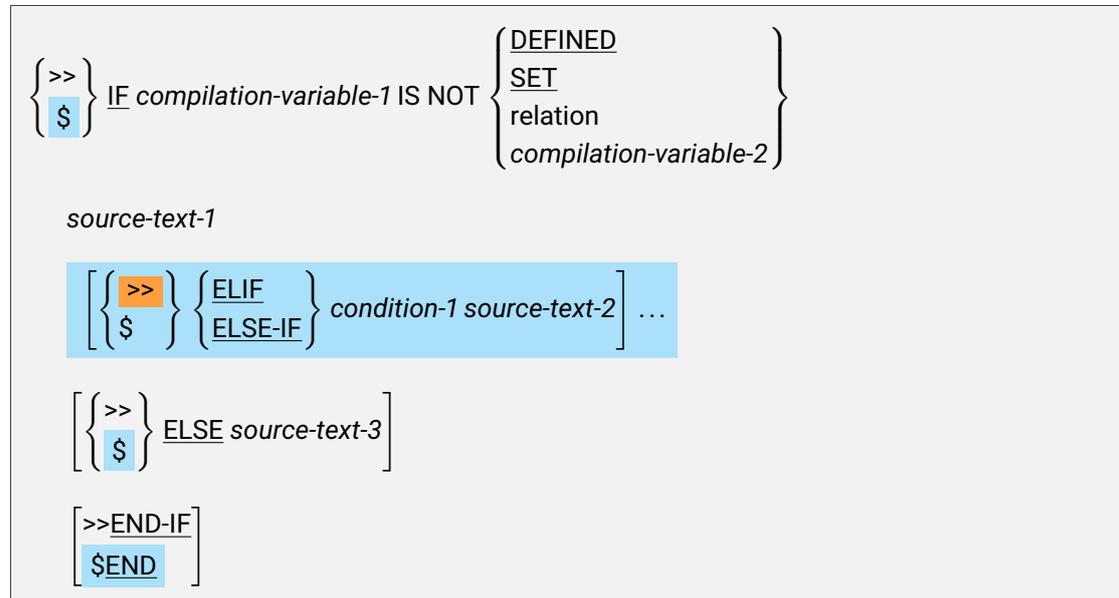
EJECT

Syntax rules

General rules

3. Compiler directives

3.8. IF directive



Syntax rules

General rules

3. Compiler directives

3.9. LEAP-SECOND directive

```
>>LEAP-SECOND
```

Syntax rules

General rules

3. Compiler directives

3.10. LISTING directive

```
>>LISTING { ON }  
          { OFF }
```

Syntax rules

General rules

3.11. @OPTIONS directive

```
@OPTIONS [options-text]
```

Syntax rules

General rules

3. Compiler directives

3.12. PAGE directive

```
>>PAGE [comment-text]
```

Syntax rules

General rules

3.13. **PROCESS** statement



Syntax rules

General rules

3.14. REPLACE statement

Format 1 (on)

```
REPLACE [ALSO] { { == pseudo-text-1 == } BY { == pseudo-text-2 == }  
               { identifier-1 } { identifier-2 }  
               { LEADING } == partial-word-1 == BY == partial-word-2 ==  
               { TRAILING } } .....
```

Format 2 (off)

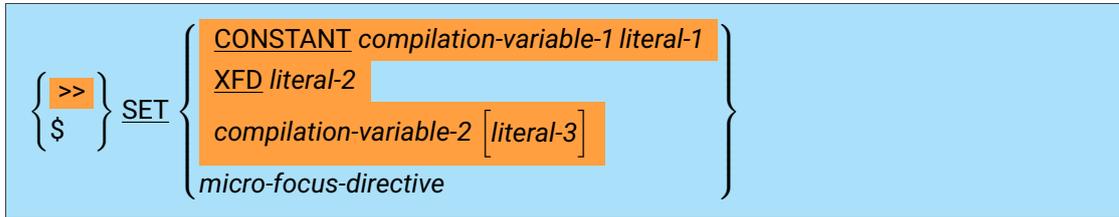
```
REPLACE [LAST] OFF.
```

Syntax rules

General rules

3. Compiler directives

3.15. SET directive



Syntax rules

General rules

3. Compiler directives

3.16. SKIP1 statement



Syntax rules

General rules

3.17. **SKIP2 statement**



Syntax rules

General rules

3. Compiler directives

3.18. SKIP3 statement



SKIP3

Syntax rules

General rules

3. Compiler directives

3.19. SOURCE directive

>>SOURCE FORMAT IS {
 FIXED
 FREE
 VARIABLE }

Syntax rules

General rules

3. Compiler directives

3.20. TITLE statement

TITLE *literal-1*

Syntax rules

General rules

3. Compiler directives

3.21. TURN directive

```
>>TURN {exception-name-1}... [ON  
OFF] [WITH LOCATION]
```

Syntax rules

General rules

3. Compiler directives

3.22. Micro Focus directives

3.22.1. ADDRSV directive

$\left\{ \begin{array}{l} \text{ADDRSV} \\ \text{ADD-RSV} \end{array} \right\} \text{literal-1} \dots$

Syntax rules

General rules

3. Compiler directives

3.22.2. ADDSYN directive

$$\left\{ \begin{array}{l} \text{ADDSYN} \\ \text{ADD-SYN} \end{array} \right\} \text{ literal-1} = \text{literal-2}$$

Syntax rules

General rules

3. Compiler directives

3.22.3. CALLFH directive

```
CALLFH [literal-1]
```

Syntax rules

General rules

3. Compiler directives

3.22.4. COMP1 directive

$\left\{ \begin{array}{l} \text{COMP1} \\ \text{COMP-1} \end{array} \right\}$

Syntax rules

General rules

3. Compiler directives

3.22.5. FOLDCOPYNAME directive

Format 1 (enable)

$\left\{ \begin{array}{l} \text{FOLDCOPYNAME} \\ \text{FOLD-COPY-NAME} \end{array} \right\} \text{literal-1}$

Format 2 (disable)

$\left\{ \begin{array}{l} \text{NOFOLDCOPYNAME} \\ \text{NOFOLD-COPY-NAME} \\ \text{NO-FOLD-COPY-NAME} \end{array} \right\}$

Syntax rules

General rules

3. Compiler directives

3.22.6. MAKESYN directive

$\left\{ \begin{array}{l} \text{MAKESYN} \\ \text{MAKE-SYN} \end{array} \right\} \text{literal-1} = \text{literal-2}$

Syntax rules

General rules

3. Compiler directives

3.22.7. OVERRIDE directive

```
OVERRIDE { literal-1 = literal-2 } ...
```

Syntax rules

General rules

3. Compiler directives

3.22.8. REMOVE directive

```
REMOVE literal-1 ...
```

Syntax rules

General rules

3. Compiler directives

3.22.9. SOURCEFORMAT directive

$\left\{ \begin{array}{l} \text{SOURCEFORMAT} \\ \text{SOURCE-FORMAT} \end{array} \right\} \textit{literal-1}$

Syntax rules

General rules

3.23. Predefined compilation variables

GnuCOBOL defines compilation variables when certain conditions are true. If the condition associated with a variable is false, the variable is not defined.

Name	Condition
DEBUG	The -d debug flag is specified.
EXECUTABLE	The module being compiled contains the main program.
GCCOMP	The size of a COMP item is determined according to the GnuCOBOL scheme, where for a PICTURE of length: <ul style="list-style-type: none"> • 1–2, the item has 1 byte • 3–4, the item has 2 bytes • 5–9, the item has 4 bytes • 10–18, the item has 8 bytes.
GNUCOBOL	GnuCOBOL is compiling the source unit.
HOSTSIGNS	A <i>signed</i> packed-decimal item's value may be considered NUMERIC if the sign has value X"F".
IBMCOMP	The size of a COMP item is determined according to the IBM scheme, where for a PICTURE of length: <ul style="list-style-type: none"> • 1–4, the item has 2 bytes • 5–9, the item has 4 bytes • 10–18, the item has 8 bytes.
MODULE	The module being compiled does not contain the main program.
NOHOSTSIGNS	A <i>signed</i> packed-decimal item's value may not be considered NUMERIC if the sign has value X"F".
NOIBMCOMP	The size of a COMP item is not determined according to the IBM scheme.
NOSTICKY-LINKAGE	Sticky-linkage (linkage-section items remaining allocated between invocations) is not enabled.
NOTRUNC	Numeric data items are truncated according to their internal representation.

3. Compiler directives

OCCOMP	The size of a COMP item is determined according to the GnuCOBOL scheme, where for a PICTURE of length: <ul style="list-style-type: none">• 1–2, the item has 1 byte• 3–4, the item has 2 bytes• 5–9, the item has 4 bytes• 10–18, the item has 8 bytes.
OPENCOBOL	GnuCOBOL is compiling the source unit.
P64	Pointers are greater than 32 bits long.
STICKY-LINKAGE	Sticky-linkage (linkage-section items remaining allocated between invocations) is enabled.
TRUNC	Numeric data items are truncated according to their PICTURE clauses.

4. Compilation group

[*program-definition*]
[*function-definition*] ...

where *program-definition* is

[{ IDENTIFICATION } DIVISION.]
[ID]

PROGRAM-ID. { *program-name-1* } [AS *literal-2*] IS { { COMMON } | { INITIAL } | { RECURSIVE } } PROGRAM .
[EXTERNAL]

[*comment-paragraphs*]

[*environment-division*]
[*data-division*]
[*procedure-division* [*program-definition*] ...]
[END PROGRAM { *program-name-1* }]

where *function-definition* is

[{ IDENTIFICATION } DIVISION.]
[ID]

FUNCTION-ID. { *function-name-1* } [AS *literal-4*] .
[*literal-3*]

[*comment-paragraphs*]

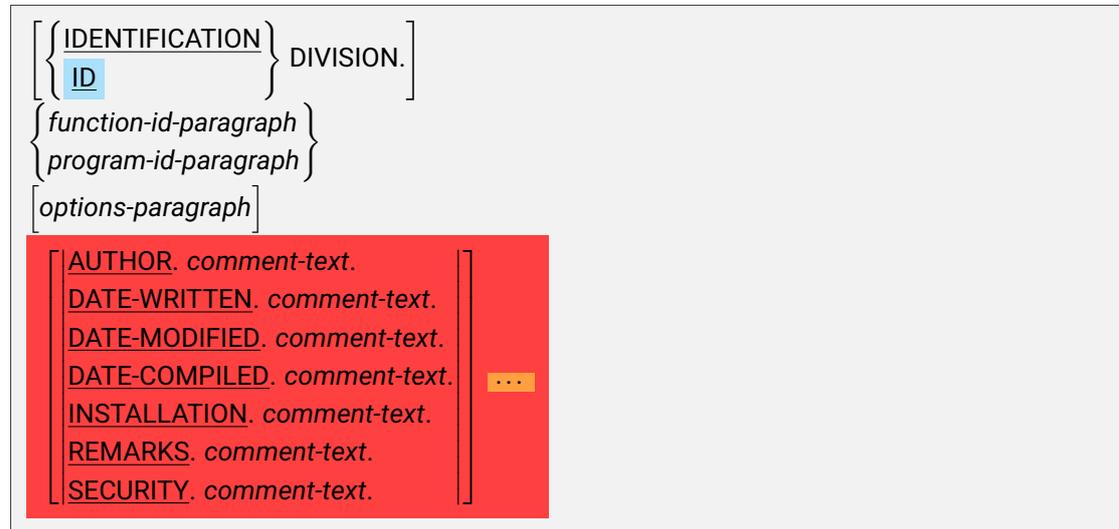
[*environment-division*]
[*data-division*]
[*procedure-division*]
END FUNCTION { *function-name-1* } .
[*literal-3*]

4. *Compilation group*

Syntax rules

General rules

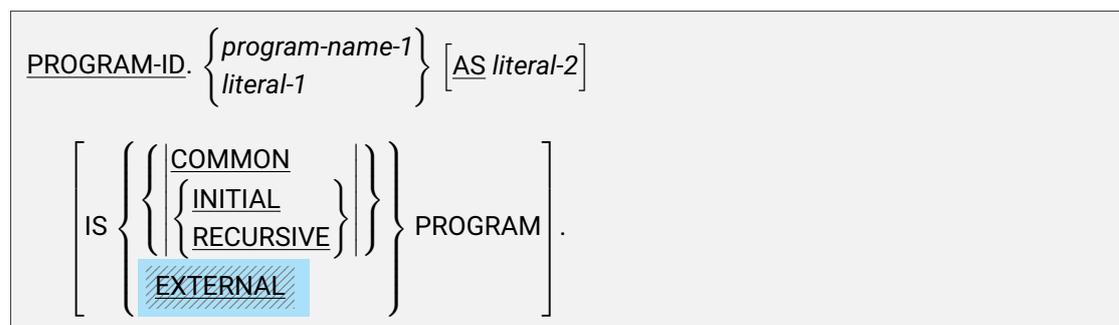
5. Identification division



Syntax rules

General rules

5.1. PROGRAM-ID paragraph



Syntax rules

General rules

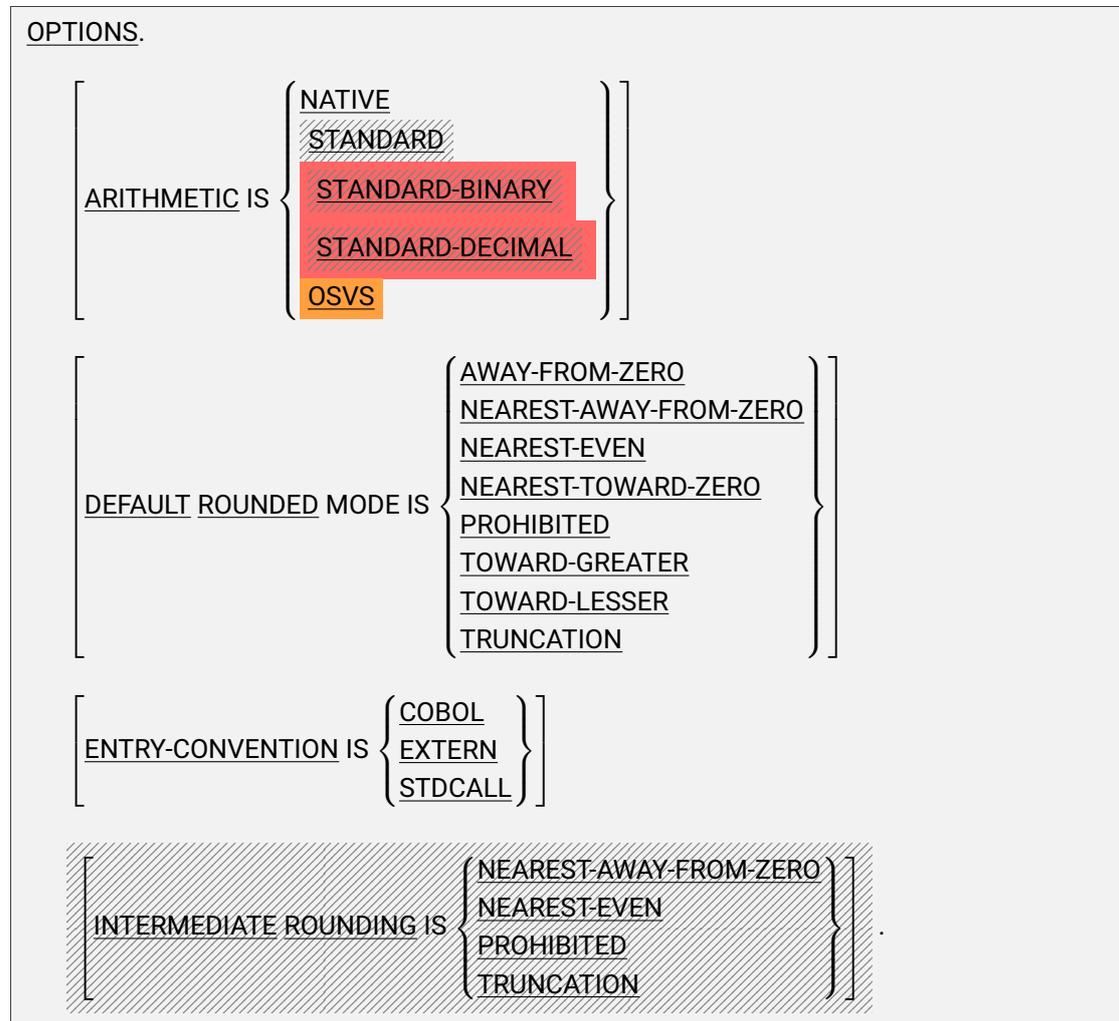
5.2. FUNCTION-ID paragraph

$$\text{FUNCTION-ID.} \left. \begin{array}{l} \text{function-name-1} \\ \text{literal-1} \end{array} \right\} [\text{AS literal-2}].$$

Syntax rules

General rules

5.3. OPTIONS paragraph



Syntax rules

General rules

6. Environment division

```
[ENVIRONMENT DIVISION.]  
[configuration-section]  
[input-output-section]
```

Syntax rules

General rules

6.1. Configuration section

Format 1 (standard)

```
CONFIGURATION SECTION.  
[source-computer-paragraph]  
[object-computer-paragraph]  
[special-names-header [special-names-entry] ...]  
[repository-paragraph]
```

Format 2 (micro-focus-and-gnucobol)

```
[CONFIGURATION SECTION.]  
[source-computer-paragraph  
object-computer-paragraph  
special-names-header  
special-names-entry  
repository-paragraph]
```

6. Environment division

Syntax rules

General rules

6.1.1. SOURCE-COMPUTER paragraph

The SOURCE-COMPUTER paragraph identifies the computer on which the compilation unit should be compiled.

```
SOURCE-COMPUTER. [ { computer-name-1 } ... [ WITH DEBUGGING MODE ] . ]
```

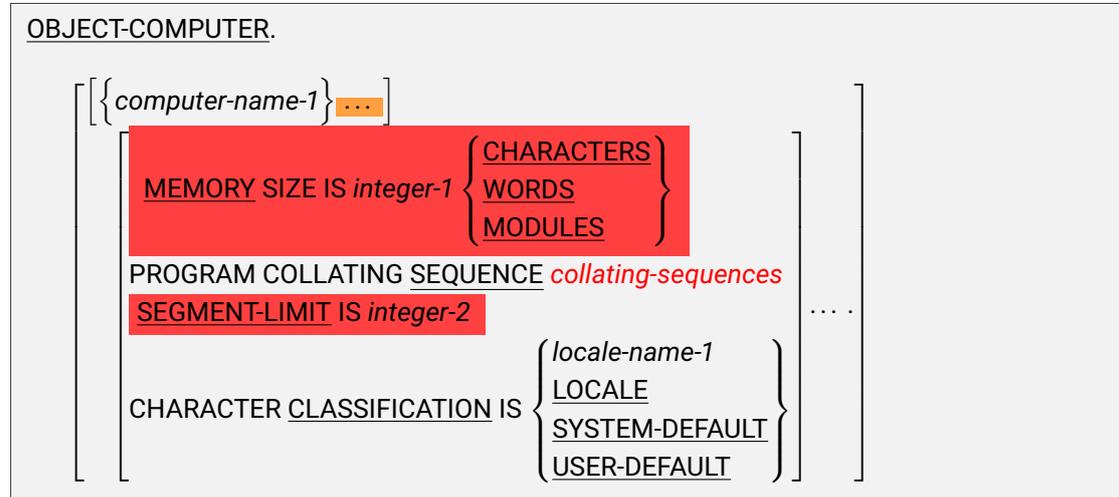
Syntax rules

General rules

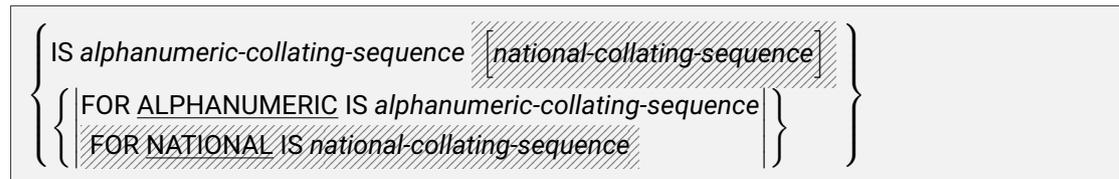
6. Environment division

6.1.2. OBJECT-COMPUTER paragraph

The OBJECT-COMPUTER paragraph identifies the computer on which the runtime module should be run.



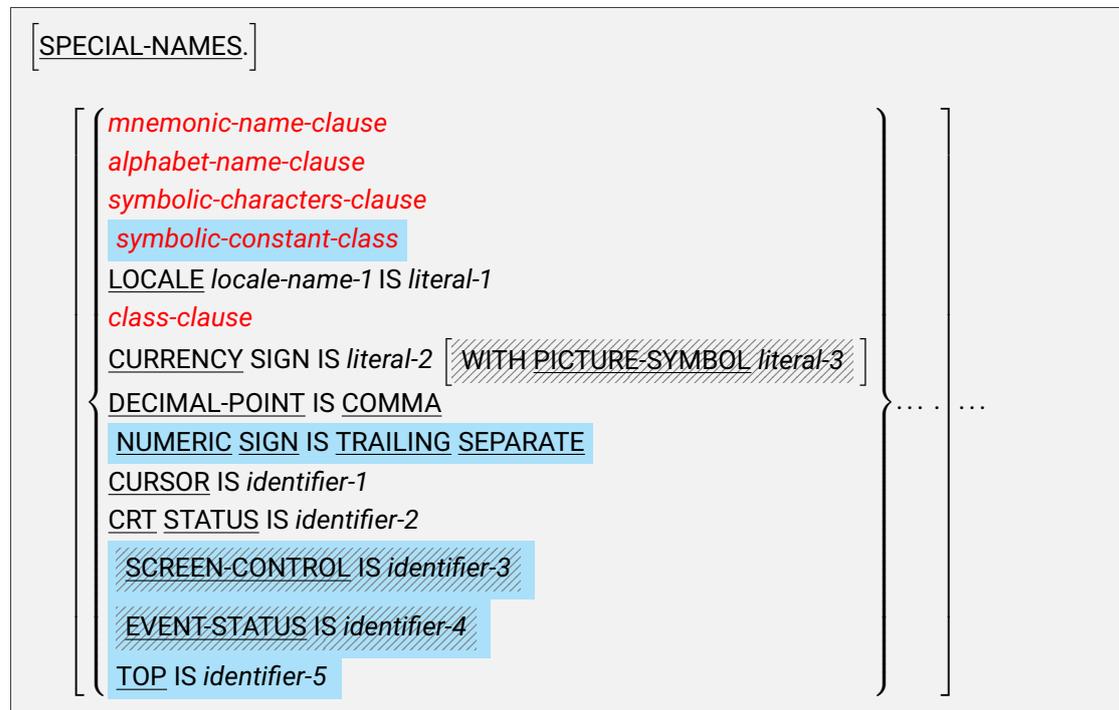
where *collating-sequences* is:



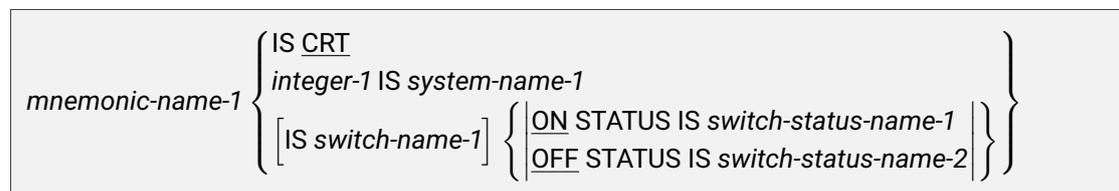
Syntax rules

General rules

6.1.3. SPECIAL-NAMES paragraph

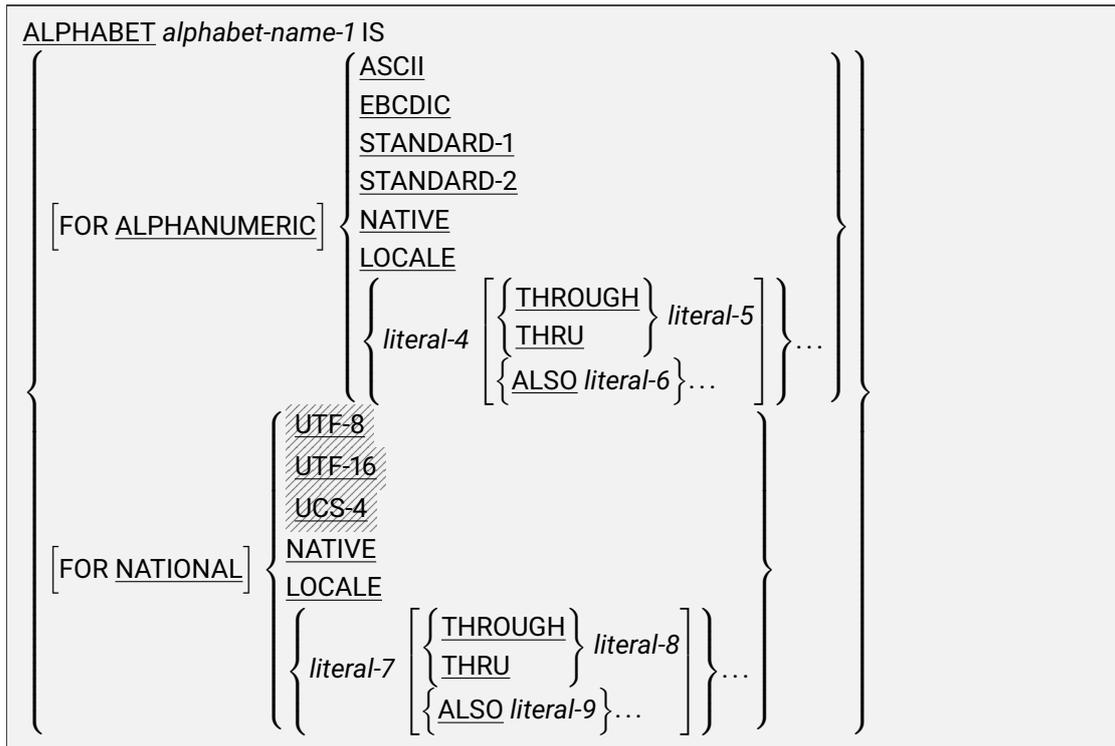


where *mnemonic-name-clause* is

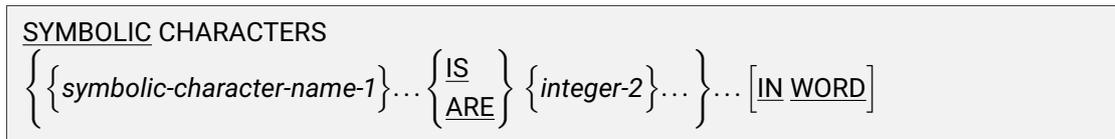


where *alphabet-name-clause* is

6. Environment division



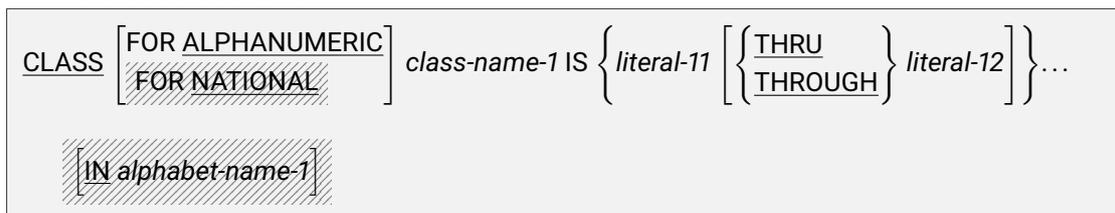
where *symbolic-characters-clause* is



where *symbolic-constant-clause* is



where *class-clause* is



Syntax rules

General rules

6. Environment division

6.1.4. REPOSITORY paragraph

REPOSITORY.

$$\left[\left\{ \begin{array}{l} \underline{\text{FUNCTION}} \left\{ \begin{array}{l} \{function-name-1\} \dots \\ \underline{\text{ALL}} \end{array} \right\} \underline{\text{INTRINSIC}} \\ \underline{\text{FUNCTION}} \text{function-name-2} \left[\underline{\text{AS}} \text{literal-1} \right] \\ \underline{\text{PROGRAM}} \text{program-name-1} \left[\underline{\text{AS}} \text{literal-2} \right] \end{array} \right\} \dots \right]$$

Syntax rules

General rules

6.2. Input-output section

```
[INPUT-OUTPUT SECTION.]
[file-control-paragraph]
[i-o-control-paragraph]
```

Syntax rules

General rules

6.2.1. FILE-CONTROL paragraph

```
[FILE-CONTROL.]
[file-control-entry] ...
```

where *file-control-entry* is

```
SELECT [OPTIONAL  
NOT OPTIONAL] file-name-1
[actual-key-clause
assign-clause
access-mode-clause
alternate-record-key-clause
collating-sequence-clause
collating-sequence-key-clause
file-limit-clause
file-status-clause
lock-mode-clause
organization-clause
padding-character-clause
password-clause
record-delimiter-clause
record-key-clause
relative-key-clause
reserve-clause
sharing-clause
track-area-clause
track-limit-clause
...]
```

where *actual-key-clause* is

6. Environment division

COLLATING SEQUENCE { IS alphanumeric-collating-sequence [national-collating-sequence] }
 { { FOR ALPHANUMERIC IS alphanumeric-collating-sequence } }
 { { FOR NATIONAL IS national-collating-sequence } }

where *collating-sequence-key-clause* is

COLLATING SEQUENCE OF identifier-7 IS alphabet-name-1

where *file-limit-clause* is

{ FILE-LIMIT } { IS } { { identifier-8 } { THROUGH } { identifier-9 } }
 { FILE-LIMITS } { ARE } { { literal-4 } { THRU } { literal-5 } } ...

where *file-status-clause* is

[FILE] STATUS IS identifier-10 [identifier-11]
 [SORT]

where *lock-mode-clause* is

LOCK MODE IS
 { { { MANUAL } } { WITH LOCK ON [MULTIPLE] { RECORD } } }
 { { { AUTOMATIC } } { WITH LOCK ON [MULTIPLE] { RECORDS } } } [WITH ROLLBACK] }
 { EXCLUSIVE [WITH MASS-UPDATE] }

where *nominal-key-clause* is

NOMINAL KEY IS identifier-12

where *organization-clause* is

{ { ORGANIZATION } IS } { INDEXED }
 { { ORGANISATION } } { LINE SEQUENTIAL }
 { RECORD BINARY SEQUENTIAL }
 { RELATIVE }

where *padding-character-clause* is

PADDING CHARACTER IS { identifier-13 }
 { literal-6 }

where *password-clause* is

6. Environment division

PASSWORD IS identifier-14

where *record-delimiter-clause* is

RECORD DELIMITER IS { STANDARD-1
LINE-SEQUENTIAL
BINARY-SEQUENTIAL }

where *record-key-clause* is

RECORD KEY IS identifier-15 [{ =
SOURCE IS } { *identifier-16* } ...]

[PASSWORD IS identifier-17]

[WITH [NO] DUPLICATES]

where *relative-key-clause* is

RELATIVE KEY IS identifier-18

where *reserve-clause* is

RESERVE { NO
integer-1 } [AREA
AREAS]

where *sharing-clause* is

SHARING WITH { ALL OTHER
NO OTHER
READ ONLY }

where *track-area-clause* is

TRACK-AREA IS { *identifier-19*
literal-7 } CHARACTERS

where *track-area-clause* is

TRACK-LIMIT IS integer-2 { TRACK
TRACKS }

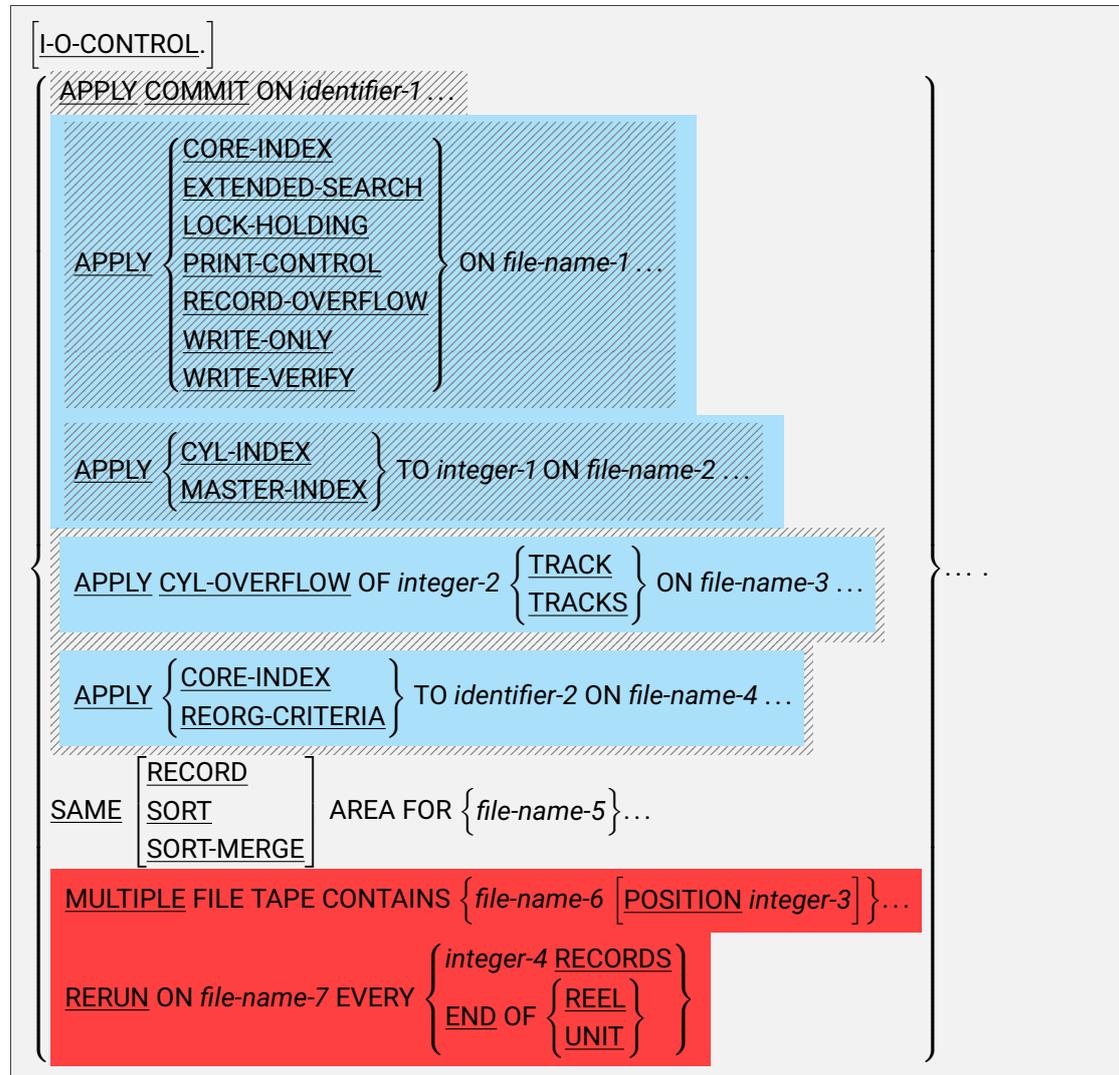
6. *Environment division*

Syntax rules

General rules

6. Environment division

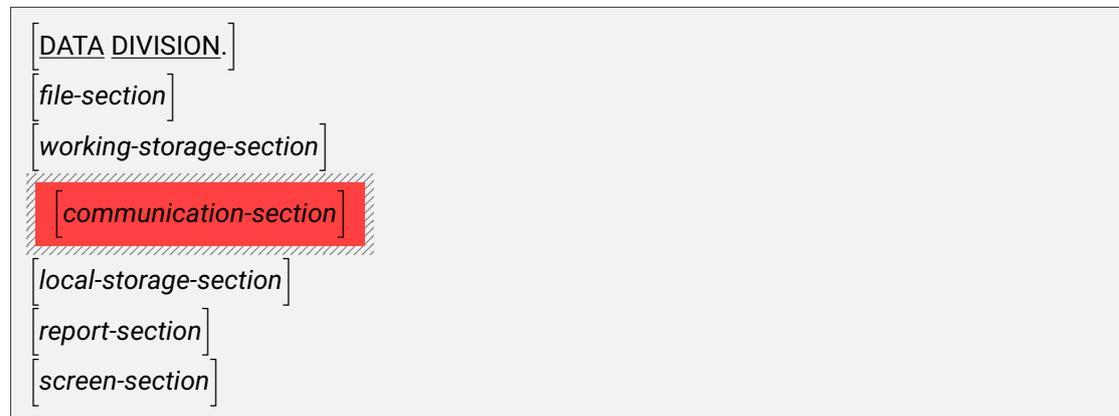
6.2.2. I-O-CONTROL paragraph



Syntax rules

General rules

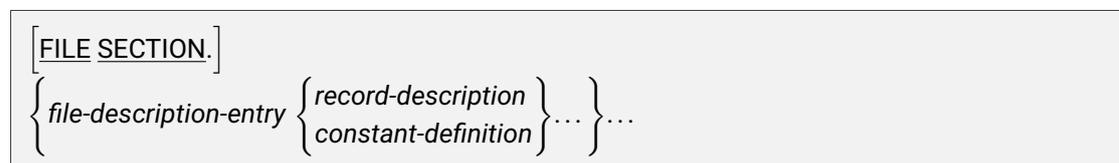
7. Data division



Syntax rules

General rules

7.1. File section

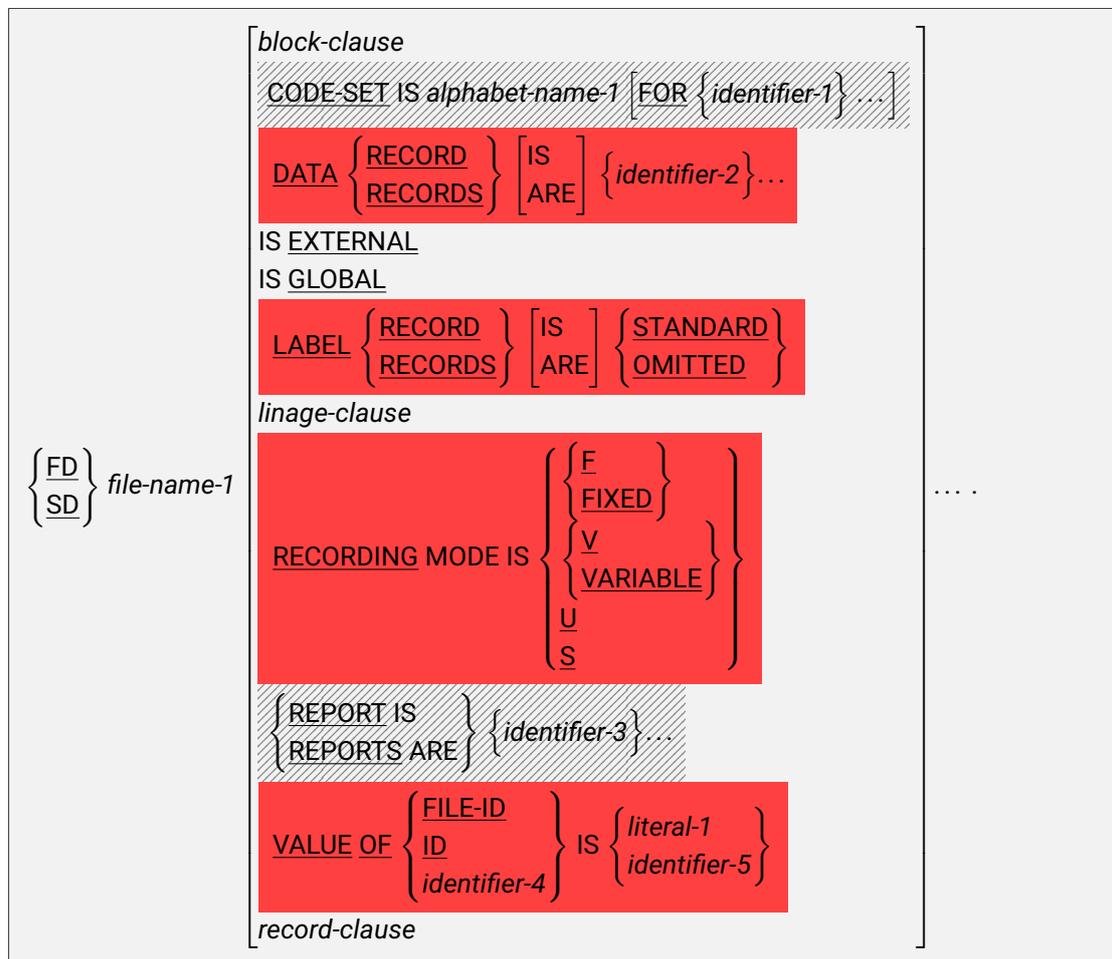


7. Data division

Syntax rules

General rules

7.1.1. File description entry



Syntax rules

General rules

7.2. Working-storage section

WORKING-STORAGE SECTION.

[*constant-definition*
record-description] ...

Syntax rules

General rules

7.3. Communication section

```
COMMUNICATION SECTION. [ communication-description-entry [ record-description
constant-definition ] ... ] ...
```

7.3.1. Communication description entry

Format 1 (input)

```
CD entry-name FOR [ INITIAL ] INPUT
{
  SYMBOLIC QUEUE IS identifier-1
  SYMBOLIC SUB-QUEUE-1 IS identifier-2
  SYMBOLIC SUB-QUEUE-2 IS identifier-3
  SYMBOLIC SUB-QUEUE-3 IS identifier-4
  MESSAGE DATE IS identifier-5
  MESSAGE TIME IS identifier-6
  SYMBOLIC SOURCE IS identifier-7
  TEXT LENGTH IS identifier-8
  END KEY IS identifier-9
  STATUS KEY IS identifier-10
  MESSAGE COUNT IS identifier-11
  identifier-12 identifier-13 identifier-14
  identifier-15 identifier-16 identifier-17
  identifier-18 identifier-19 identifier-20
  identifier-21 identifier-22
}
```

Format 2 (output)

```
CD entry-name FOR OUTPUT
[
  DESTINATION COUNT IS identifier-23
  TEXT LENGTH IS identifier-24
  STATUS KEY IS identifier-25
  DESTINATION TABLE OCCURS integer-1 TIMES [ INDEXED BY { index-name-1 } ... ]
  ERROR KEY IS identifier-26
  DESTINATION IS identifier-27
  SYMBOLIC DESTINATION IS identifier-28
]
```


7.4. Local-storage section

LOCAL-STORAGE SECTION.

[*constant-definition*
record-description] ...

Syntax rules

General rules

7.5. Linkage section

```
LINKAGE SECTION.  
[constant-definition  
record-description] ...
```

Syntax rules

General rules

7.6. Report section

```
REPORT SECTION.
  [constant-definition]
  [report-description] ...
```

Syntax rules

General rules

7.6.1. Report description

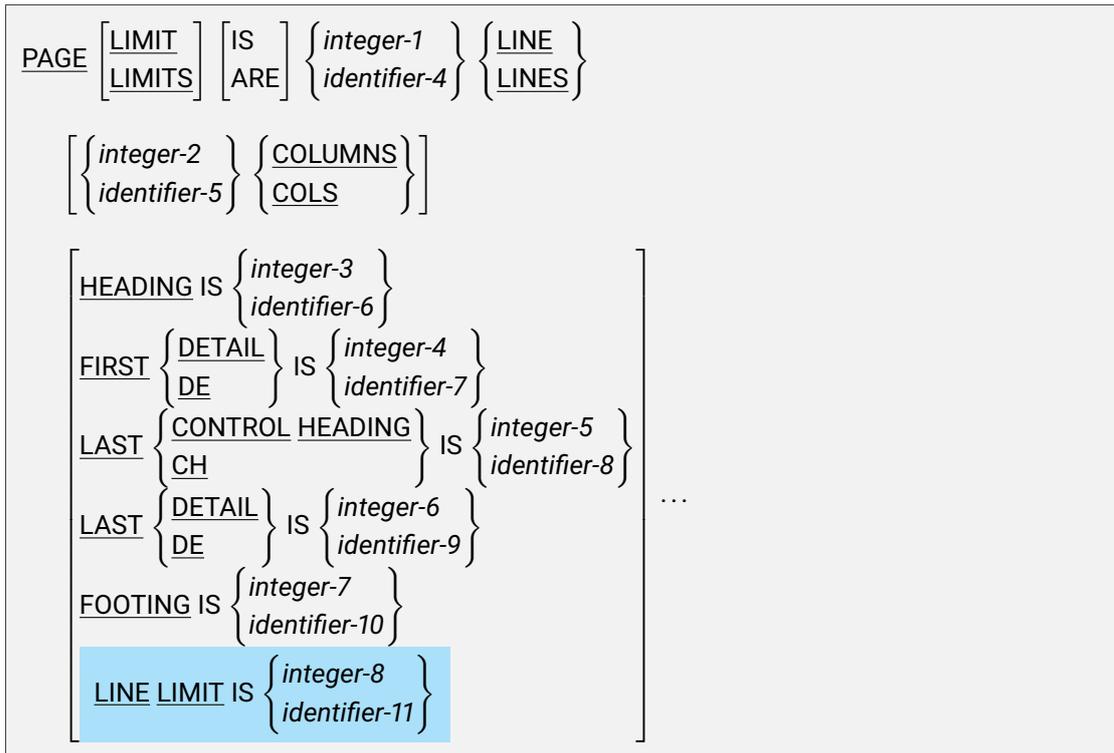
```
RD report-name-1

  [IS GLOBAL
  WITH CODE IS { identifier-1
                literal-1
  { CONTROL } [IS] { { identifier-2 } ...
  { CONTROLS } [ARE] { FINAL [ identifier-3 ] ... } ] ...
  page-limits-clause

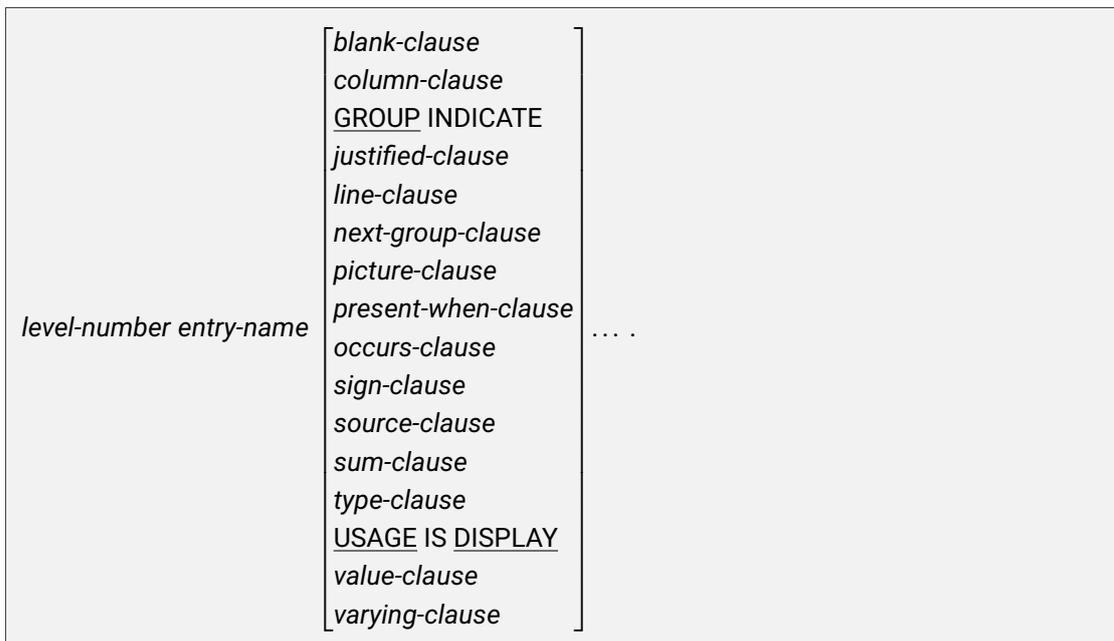
  { report-group-description-1 } ...
```

where *page-limits-clause* is

7. Data division



where *report-group-description* is



7. *Data division*

Syntax rules

General rules

7.7. Screen section

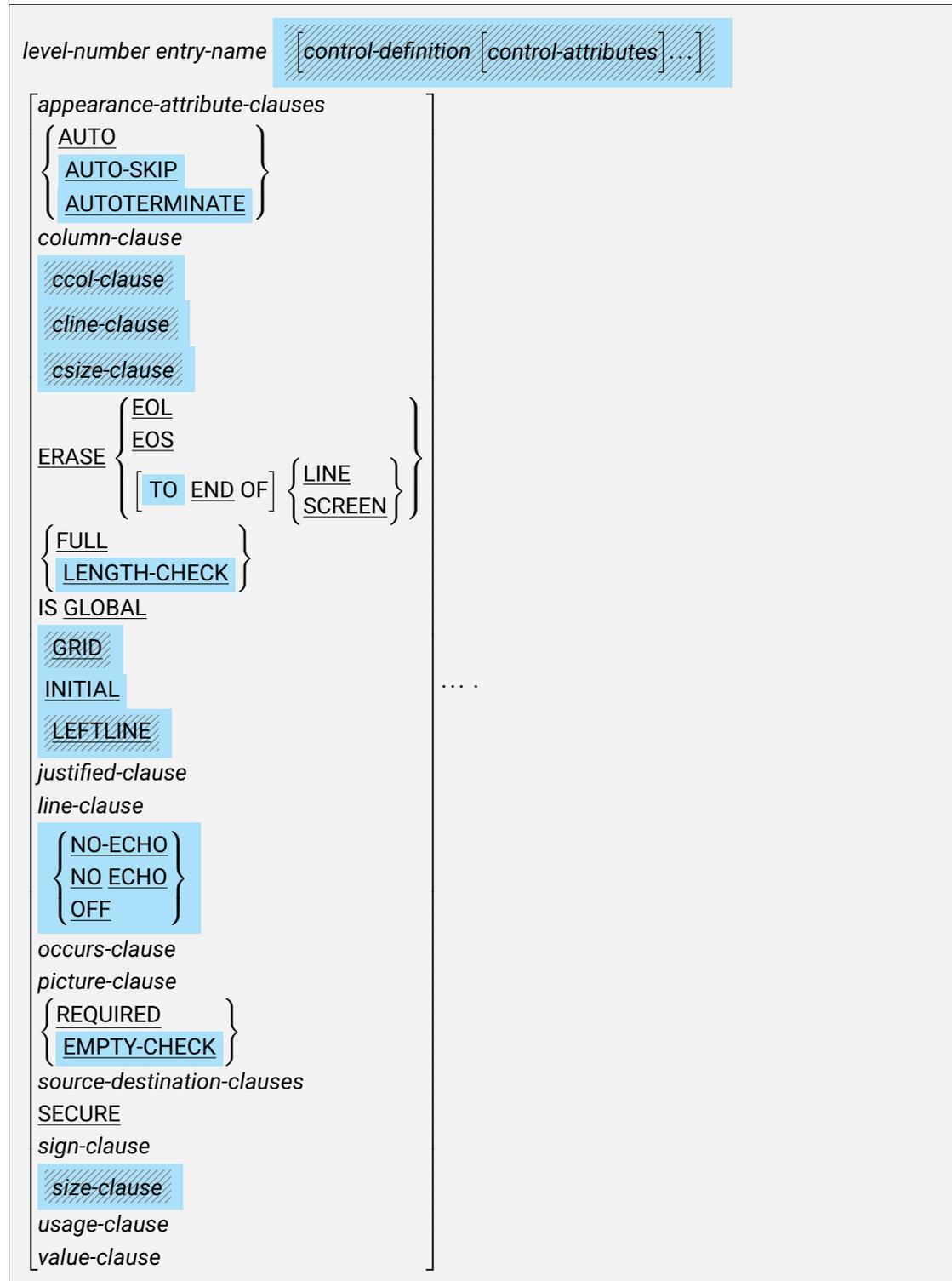
```
SCREEN SECTION.  
  [ constant-definition ]  
  [ screen-description ] ...
```

7. Data division

Syntax rules

General rules

7.7.1. Screen description



7. Data division

where *appearance-attribute-clauses* is

7. Data division

$\left[\left\{ \begin{array}{l} \text{BACKGROUND-COLOR} \\ \text{BACKGROUND-COLOUR} \end{array} \right\} \text{ IS } \left\{ \begin{array}{l} \text{identifier-1} \\ \text{integer-1} \end{array} \right\} \right]$

$\left[\begin{array}{l} \text{BELL} \\ \text{BEEP} \end{array} \right]$

$\text{BLANK} \left\{ \begin{array}{l} \text{LINE} \\ \text{SCREEN} \end{array} \right\}$

$\left\{ \begin{array}{l} \text{BLINK} \\ \text{BLINKING} \end{array} \right\}$

$\text{WITH} \left\{ \begin{array}{l} \text{COLOR} \\ \text{COLOUR} \end{array} \right\} \text{ IS } \left\{ \begin{array}{l} \text{identifier-2} \\ \text{integer-2} \end{array} \right\}$

$\left[\left\{ \begin{array}{l} \text{FOREGROUND-COLOR} \\ \text{FOREGROUND-COLOUR} \end{array} \right\} \text{ IS } \left\{ \begin{array}{l} \text{identifier-3} \\ \text{integer-3} \end{array} \right\} \right]$

$\left[\begin{array}{l} \text{HIGHLIGHT} \\ \text{HIGH} \\ \text{BOLD} \\ \text{LOWLIGHT} \\ \text{LOW} \end{array} \right]$

$\left[\text{WITH STANDARD} \right]$

$\left[\text{WITH BACKGROUND-HIGH} \right]$

$\left[\text{WITH BACKGROUND-STANDARD} \right]$

$\left[\text{WITH BACKGROUND-LOW} \right]$

$\left[\text{OVERLINE} \right]$

$\left[\text{PROMPT} \left[\text{CHARACTER IS } \left\{ \begin{array}{l} \text{identifier-4} \\ \text{literal-1} \end{array} \right\} \right] \right]$

$\left[\begin{array}{l} \text{REVERSE-VIDEO} \\ \text{REVERSED} \\ \text{REVERSE} \end{array} \right]$

$\left[\begin{array}{l} \text{UNDERLINE} \\ \text{UNDERLINED} \end{array} \right]$

7. Data division

where *source-destination-clauses* is

[FROM { *identifier-5*
literal-2 }]

[TO *identifier-6*]

[USING *identifier-7*]

Syntax rules

General rules

7.8. Record description

Format 1 (data-description)

<i>level-number entry-name</i> ANY { <u>LENGTH</u> } <u>NUMERIC</u> <i>blank-when-zero-clause</i> IS EXTERNAL [<u>AS literal-1</u>] <u>IS EXTERNAL-FORM</u> <u>IS IDENTIFIED BY</u> { <i>identifier-1</i> } <i>literal-2</i> IS GLOBAL <i>justified-clause</i> <i>occurs-clause</i> { <u>PICTURE</u> } IS <i>picture-string-1</i> [<u>LOCALE</u> [IS <i>locale-name-1</i>] <u>SIZE IS integer-1</u>] <u>PIC</u> <u>REDEFINES</u> <i>identifier-2</i> <i>sign-clause</i> { <u>SYNCHRONIZED</u> } [<u>LEFT</u>] <u>SYNCHRONISED</u> <u>RIGHT</u> <u>SYNC</u> <i>usage-clause</i> <i>value-clause</i>
---	-------

Format 2 (renames)

66 <i>identifier-3</i> <u>RENAMES</u> <i>identifier-4</i> [{ <u>THROUGH</u> } <i>identifier-5</i>] <u>THRU</u>

Format 3 (condition-name)

7. Data division

88 *identifier-6* { VALUE
VALUES } [IS
ARE] { *literal-3* [{ THROUGH
THRU } *literal-4*] } ...
[WHEN SET TO FALSE IS *literal-5*].

Syntax rules

General rules

7.9. Constant definition

Format 1 (standard)

$$\left. \begin{array}{l} 1 \\ 01 \end{array} \right\} \text{identifier-1 CONSTANT [IS GLOBAL] } \left\{ \begin{array}{l} \text{AS } \left\{ \begin{array}{l} \text{literal-1} \\ \text{BYTE-LENGTH} \\ \text{LENGTH} \end{array} \right\} \text{ OF identifier-2} \\ \text{FROM identifier-3} \end{array} \right\} .$$

Format 2 (micro-focus)

$$78 \text{ identifier-4 } \left[\text{IS GLOBAL} \right] \left\{ \begin{array}{l} \text{VALUE} \\ \text{VALUES} \end{array} \right\} \left[\text{IS} \right] \left[\text{ARE} \right] \left\{ \begin{array}{l} \text{literal-2} \\ \text{START OF identifier-5} \\ \text{NEXT} \end{array} \right\} .$$

Syntax rules

General rules

7.10. Data division clauses

7.10.1. ANY LENGTH clause

The ANY LENGTH clause specifies that the length of the data item will be determined at runtime.

$\text{ANY} \left\{ \begin{array}{l} \text{LENGTH} \\ \text{NUMERIC} \end{array} \right\}$
--

Syntax rules

General rules

7. Data division

7.10.2. AUTO clause

The AUTO clause specifies that the screen cursor will immediately move to the next screen item when the current screen item is full.

```
{  
  AUTO  
  AUTO-SKIP  
  AUTOTERMINATE  
}
```

Syntax rules

General rules

7.10.3. BACKGROUND-COLOR clause

The BACKGROUND-COLOR clause specifies the background-color of the screen item.

$$\left\{ \begin{array}{l} \text{BACKGROUND-COLOR} \\ \text{BACKGROUND-COLOUR} \end{array} \right\} \text{ IS } \left\{ \begin{array}{l} \text{identifier-1} \\ \text{literal-1} \end{array} \right\}$$

Syntax rules

General rules

7. Data division

7.10.4. BACKGROUND-HIGH clause

~~BACKGROUND-HIGH~~

Syntax rules

General rules

7. Data division

7.10.5. BACKGROUND-LOW clause

BACKGROUND-LOW

Syntax rules

General rules

7. Data division

7.10.6. BACKGROUND-STANDARD clause

BACKGROUND-STANDARD

Syntax rules

General rules

7. Data division

7.10.7. BELL clause

{
 BELL
 BEEP
}

Syntax rules

General rules

7.10.8. BLANK clause

BLANK { LINE
SCREEN }

Syntax rules

General rules

7.10.9. BLANK WHEN ZERO clause

The BLANK WHEN ZERO clause causes an item to be blanked when a value of zero is stored in it.



Syntax rules

General rules

7. Data division

7.10.10. BLINK clause

{
BLINK
BLINKING
}

Syntax rules

General rules

7. Data division

7.10.11. BLOCK clause

The BLOCK clause specifies the size of a physical record, that is, how many logical records should be read in one physical I/O operation.

<code>BLOCK CONTAINS <i>integer-1</i> [<u>TO</u> <i>integer-2</i>] [<u>CHARACTERS</u> <u>RECORDS</u>]</code>
--

Syntax rules

General rules

7.10.12. COLOR clause

WITH {COLOR} IS {*identifier-1*}
{COLOUR} {*literal-1*}

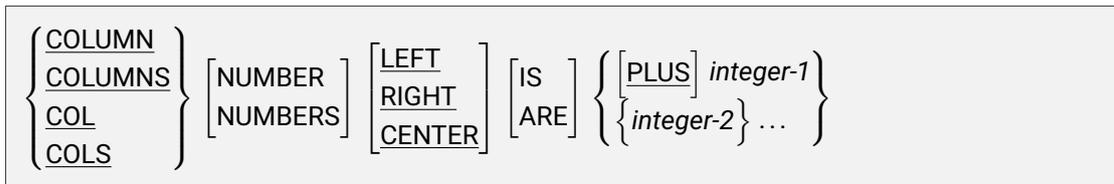
Syntax rules

General rules

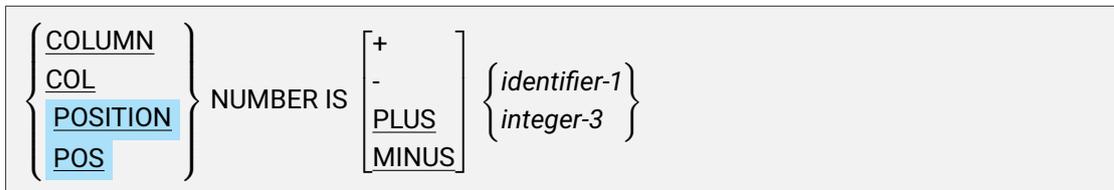
7.10.13. COLUMN clause

The COLUMN clause specifies what column an item should be printed or displayed at.

Format 1 (report-section)



Format 2 (screen-section)



Syntax rules

General rules

7.10.14. CCOL clause



Syntax rules

General rules

7.10.15. CLINE clause



Syntax rules

General rules

7.10.16. CSIZE clause

$$\text{CSIZE} \left[\begin{array}{l} \text{IS} \\ = \end{array} \right] \left\{ \begin{array}{l} \text{identifier-1} \\ \text{integer-1} \end{array} \right\}$$

Syntax rules

General rules

7.10.17. DATA RECORDS clause

DATA { RECORD IS
RECORDS ARE } { *identifier-1* } ...

Syntax rules

General rules

7. Data division

7.10.18. DESTINATION clause

DESTINATION IS *identifier-1*

Syntax rules

General rules

7. Data division

7.10.19. DESTINATION COUNT clause

DESTINATION COUNT IS *identifier-1*

Syntax rules

General rules

7.10.20. DESTINATION TABLE OCCURS clause

DESTINATION TABLE OCCURS *integer-1* TIMES [INDEXED BY {*index-name-1*}...]

Syntax rules

General rules

7. Data division

7.10.21. END KEY clause

END KEY IS *identifier-1*

Syntax rules

General rules

7. Data division

7.10.22. Entry name

The entry name specifies the name of the item being declared.

$\left[\begin{array}{l} \text{FILLER} \\ \text{identifier-1} \end{array} \right]$
--

Syntax rules

General rules

7.10.23. ERASE clause

The ERASE clause indicates part of the screen to be blanked before displaying the item.



Syntax rules

General rules

7. Data division

7.10.24. ERROR KEY clause

ERROR KEY IS *identifier-1*

Syntax rules

General rules

7. Data division

7.10.25. EXTERNAL clause

IS EXTERNAL [AS literal-1]

Syntax rules

General rules

7.10.26. EXTERNAL-FORM clause

$\left\{ \begin{array}{l} \text{IS EXTERNAL-FORM} \\ \text{IS IDENTIFIED BY } \left\{ \begin{array}{l} \text{identifier-1} \\ \text{literal-1} \end{array} \right\} \end{array} \right\}$

Syntax rules

General rules

7.10.27. FOREGROUND-COLOR clause

$\left\{ \begin{array}{l} \text{FOREGROUND-COLOR} \\ \text{FOREGROUND-COLOUR} \end{array} \right\} \text{ IS } \left\{ \begin{array}{l} \text{identifier-1} \\ \text{literal-1} \end{array} \right\}$

Syntax rules

General rules

7.10.28. FROM clause

FROM { *identifier-1* }
 { *literal-1* } }

Syntax rules

General rules

7. Data division

7.10.29. FULL clause

The FULL clause specifies that the item must be filled entirely before the cursor can move to another item.

```
{ FULL  
  LENGTH-CHECK }
```

Syntax rules

General rules

7.10.30. GLOBAL clause

The GLOBAL clause specifies that an item may be accessed from within nested programs.

IS GLOBAL

Syntax rules

General rules

7. Data division

7.10.31. GRID clause



Syntax rules

General rules

7.10.32. HIGHLIGHT clause



Syntax rules

General rules

7. Data division

7.10.33. INITIAL clause

INIITAL

Syntax rules

General rules

7. Data division

7.10.34. JUSTIFIED clause

The JUSTIFIED clause causes data smaller than the data item to be padded by spaces on the left to fill the item.

$\left. \begin{array}{l} \text{JUSTIFIED} \\ \text{JUST} \end{array} \right\} \text{RIGHT}$

Syntax rules

General rules

7.10.35. LABEL RECORDS clause

LABEL { RECORD IS
RECORDS ARE } { STANDARD
OMITTED }

Syntax rules

General rules

7. Data division

7.10.36. LEFTLINE clause

LEFTLINE

Syntax rules

General rules

7. Data division

7.10.37. Level-number

A 1- or 2-digit integer having a value that is either between 1 and 49 or is 66, 77, 78 or 88.

Syntax rules

General rules

7.10.38. LINAGE clause

The LINAGE clause specifies the page limits of a logical page.

$\text{LINAGE IS } \left\{ \begin{array}{l} \text{identifier-1} \\ \text{literal-1} \end{array} \right\} \text{ LINES } \left[\left\{ \begin{array}{l} \text{BOTTOM} \\ \text{TOP} \\ \text{WITH FOOTING AT} \end{array} \right\} \left\{ \begin{array}{l} \text{identifier-2} \\ \text{literal-2} \end{array} \right\} \right] \dots$

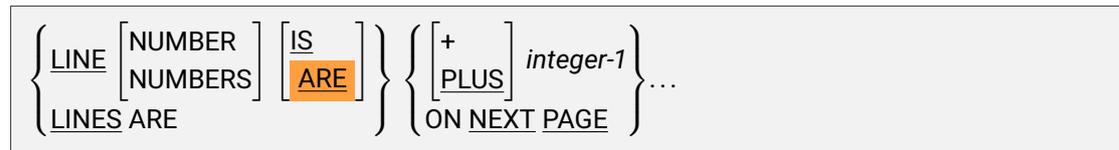
Syntax rules

General rules

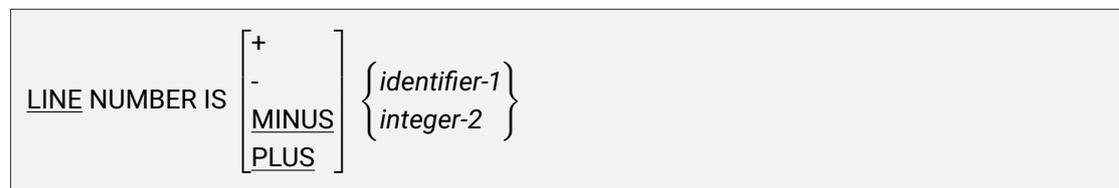
7.10.39. LINE clause

The LINE clause specifies the line an item should be printed or displayed on.

Format 1 (report section)



Format 2 (screen item)



Format 3 (screen control)



Syntax rules

General rules

7.10.40. LOWLIGHT clause

$\left\{ \begin{array}{l} \text{LOWLIGHT} \\ \text{LOW} \end{array} \right\}$

Syntax rules

General rules

7. Data division

7.10.41. MESSAGE COUNT clause

MESSAGE COUNT IS *identifier-1*

Syntax rules

General rules

7. Data division

7.10.42. MESSAGE DATE clause

MESSAGE DATE IS identifier-1

Syntax rules

General rules

7. Data division

7.10.43. MESSAGE TIME clause

MESSAGE TIME IS identifier-1

Syntax rules

General rules

7.10.44. NEXT GROUP clause

The NEXT GROUP clause specifies the number of blank lines that should follow the end of a report group.

$\text{NEXT_GROUP IS } \left\{ \begin{array}{l} \left[\begin{array}{l} + \\ \text{PLUS} \end{array} \right] \text{integer-1} \\ \text{ON NEXT PAGE} \end{array} \right\}$

Syntax rules

General rules

7.10.45. NO ECHO clause

{
NO ECHO
NO-ECHO
OFF
}

Syntax rules

General rules

7.10.46. OCCURS clause

The OCCURS clause describes tables, repeated data items accessible by subscripts.

Format 1 (usual)

```

OCCURS
  { integer-1 [TO integer-2] TIMES [DEPENDING ON identifier-1]
  { DYNAMIC [CAPACITY IN identifier-2] [FROM integer-3] [TO integer-4] [INITIALIZED]
  [ [ { { ASCENDING } KEY IS { identifier-3 } ... } ... ]
  [ INDEXED BY { index-name-1 } ... ] ] ]
    
```

Format 2 (report section)

```

OCCURS integer-5 [TO integer-6] TIMES [DEPENDING ON identifier-4] [STEP integer-7]
    
```

Format 3 (screen section)

```

OCCURS integer-8 TIMES
    
```

Format 4 (unbounded)

```

OCCURS [integer-9 TO] UNBOUNDED TIMES DEPENDING ON identifier-5
  [ [ { { ASCENDING } KEY IS { identifier-6 } ... } ... ]
  [ INDEXED BY { index-name-2 } ... ] ]
    
```

7. *Data division*

Syntax rules

General rules

7. Data division

7.10.47. OVERLINE clause

OVERLINE

Syntax rules

General rules

7.10.48. PICTURE clause

The PICTURE clause describes the general characteristics and editing requirements of an elementary data item.

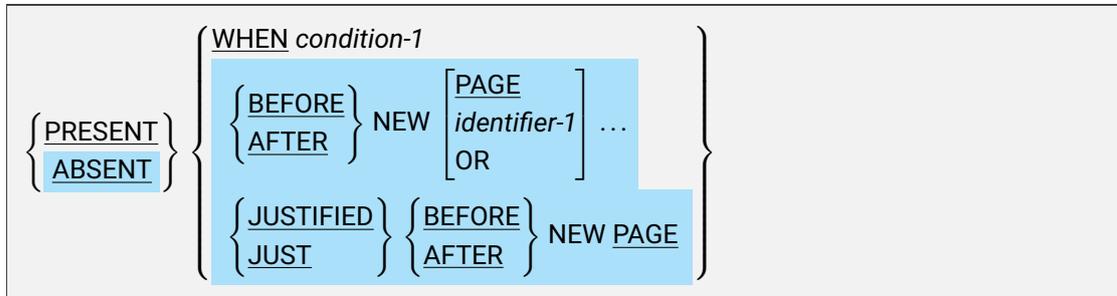
$\left\{ \begin{array}{l} \text{PICTURE} \\ \text{PIC} \end{array} \right\} \text{ IS } \textit{picture-string-1} \left[\text{LOCALE} \left[\text{IS } \textit{locale-name-1} \right] \text{ SIZE IS } \textit{integer-1} \right]$
--

Syntax rules

General rules

7.10.49. PRESENT WHEN clause

The PRESENT WHEN clause specifies a condition under which a report section entry will be processed.



Syntax rules

General rules

7.10.50. PROMPT clause

PROMPT [CHARACTER IS { *identifier-1*
literal-1 }]

Syntax rules

General rules

7.10.51. RECORD clause

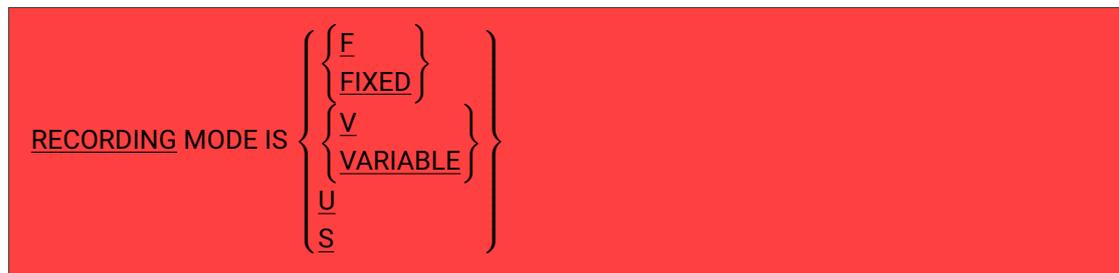
The RECORD clause specifies the number of bytes of a logical record.

$\text{RECORD} \left\{ \begin{array}{l} \text{CONTAINS } \textit{integer-1} \text{ [TO } \textit{integer-2}] \text{ CHARACTERS} \\ \text{IS } \underline{\text{VARYING}} \text{ in size [FROM } \textit{integer-3}] \text{ [TO } \textit{integer-4}] \text{ CHARACTERS} \\ \underline{\text{DEPENDING ON}} \textit{ identifier-1} \end{array} \right\}$

Syntax rules

General rules

7.10.52. RECORDING MODE clause



Syntax rules

General rules

7.10.53. REDEFINES clause

The REDEFINES clause indicates the data shares the same memory as an item with a different description.

```
REDEFINES identifier-1
```

Syntax rules

General rules

7.10.54. REPORT clause

$\left. \begin{array}{l} \text{REPORT IS} \\ \text{REPORTS ARE} \end{array} \right\} \{ \text{identifier-1} \} \dots$

Syntax rules

General rules

7.10.55. REQUIRED clause



Syntax rules

General rules

7.10.56. REVERSE-VIDEO clause



Syntax rules

General rules

7. Data division

7.10.57. SECURE clause

SECURE

Syntax rules

General rules

7. Data division

7.10.58. STATUS KEY clause

STATUS KEY IS *identifier-1*

Syntax rules

General rules

7.10.59. SIGN clause

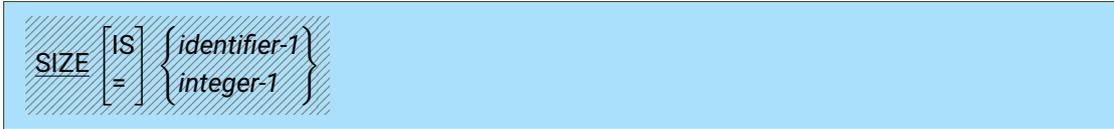
The SIGN clause defines how to store item's sign.

SIGN IS { LEADING TRAILING } [SEPARATE CHARACTER]
--

Syntax rules

General rules

7.10.60. SIZE clause



Syntax rules

General rules

7. Data division

7.10.61. SOURCE clause

The SOURCE clause identifies data to be used in processing a report section entry.

`SOURCE IS number-1 [rounded-phrase]`

Syntax rules

General rules

7. Data division

7.10.62. STANDARD clause

WITH STANDARD

Syntax rules

General rules

7.10.63. SUM clause

The SUM clause provides a list of data items to be summed for use in an elementary report item.

$$\text{SUM OF } \{ \textit{number-1} \} \dots \left[\left\{ \begin{array}{l} \text{RESET ON } \{ \textit{identifier-1} \} \\ \text{FINAL} \\ \text{UPON } \textit{identifier-2} \end{array} \right\} \right]$$

Syntax rules

General rules

7. Data division

7.10.64. SYMBOLIC DESTINATION clause

SYMBOLIC DESTINATION IS *identifier-1*

Syntax rules

General rules

7. Data division

7.10.65. SYMBOLIC QUEUE clause

SYMBOLIC QUEUE IS *identifier-1*

Syntax rules

General rules

7. Data division

7.10.66. SYMBOLIC SOURCE clause

`SYMBOLIC SOURCE IS identifier-1`

Syntax rules

General rules

7. Data division

7.10.67. SYMBOLIC SUB-QUEUE-1 clause

SYMBOLIC SUB-QUEUE-1 IS *identifier-1*

Syntax rules

General rules

7.10.68. SYMBOLIC SUB-QUEUE-2 clause

SYMBOLIC SUB-QUEUE-2 IS *identifier-1*

Syntax rules

General rules

7. Data division

7.10.69. SYMBOLIC SUB-QUEUE-3 clause

SYMBOLIC SUB-QUEUE-3 IS *identifier-1*

Syntax rules

General rules

7. Data division

7.10.70. SYMBOLIC TERMINAL clause

SYMBOLIC TERMINAL IS *identifier-1*

Syntax rules

General rules

7.10.71. SYNCHRONIZED clause

The SYNCHRONIZED clause specifies an item should be aligned in a byte boundary and in what way.



Syntax rules

General rules

7. Data division

7.10.72. TEXT LENGTH clause

TEXT LENGTH IS *identifier-1*

Syntax rules

General rules

7. Data division

7.10.73. TO clause

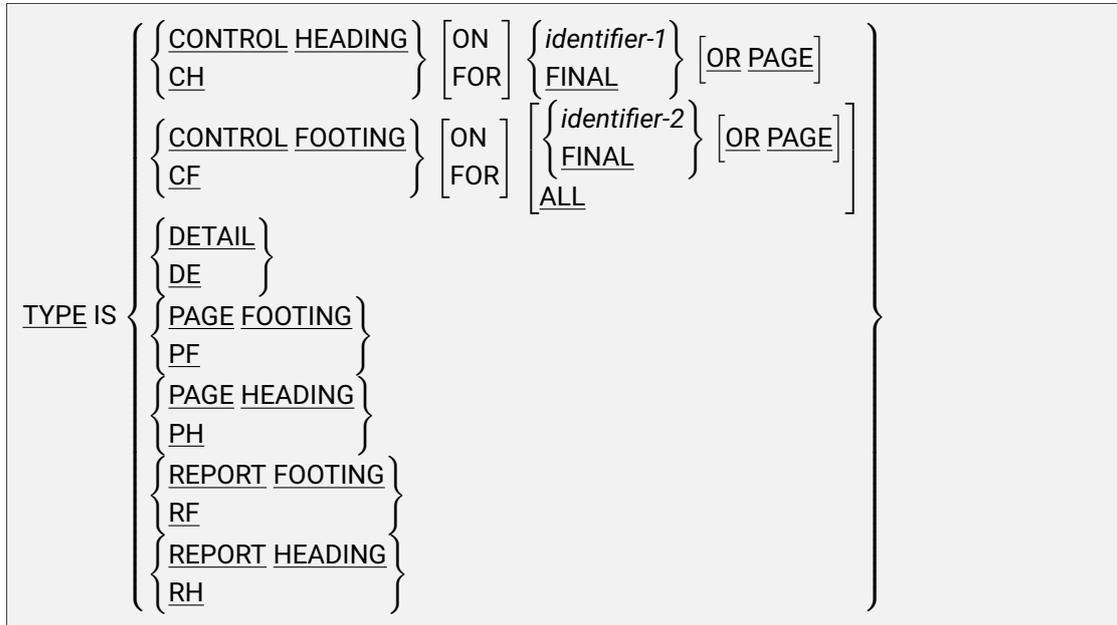
TO *identifier-1*

Syntax rules

General rules

7.10.74. TYPE clause

The TYPE clause specifies when to print a report group.



Syntax rules

General rules

7.10.75. UNDERLINE clause

The UNDERLINE clause specifies that each character of a field is to be displayed with an underline.

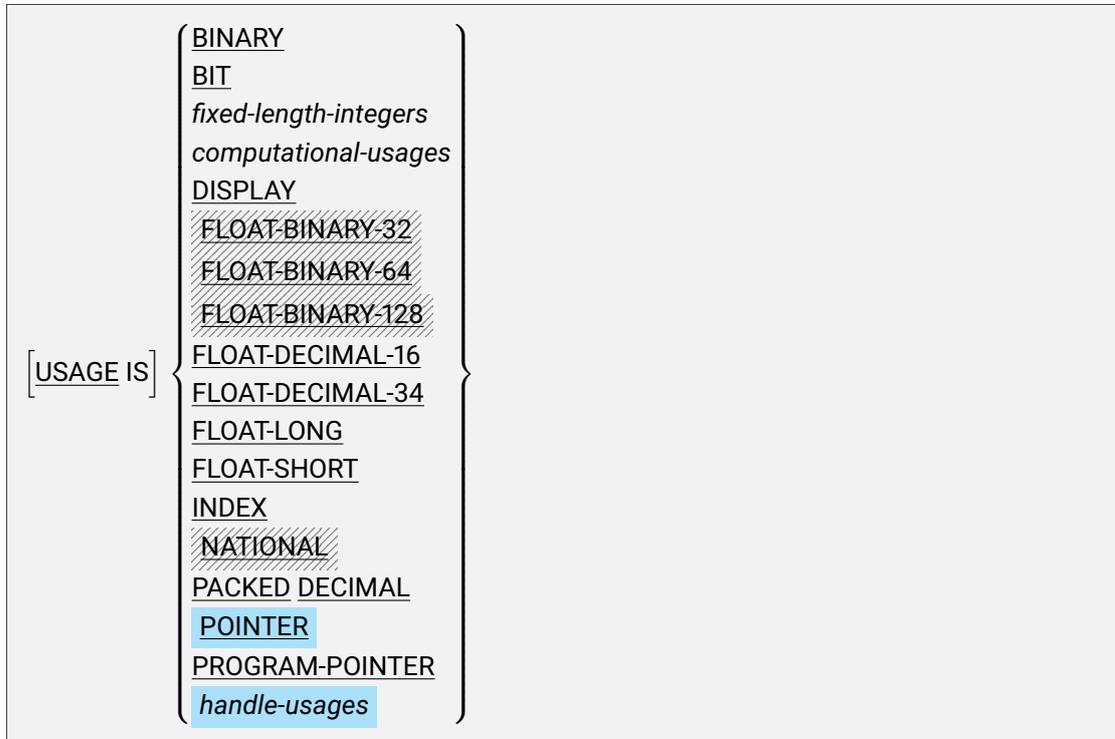
$\left\{ \begin{array}{l} \text{UNDERLINE} \\ \text{UNDERLINED} \end{array} \right\}$

Syntax rules

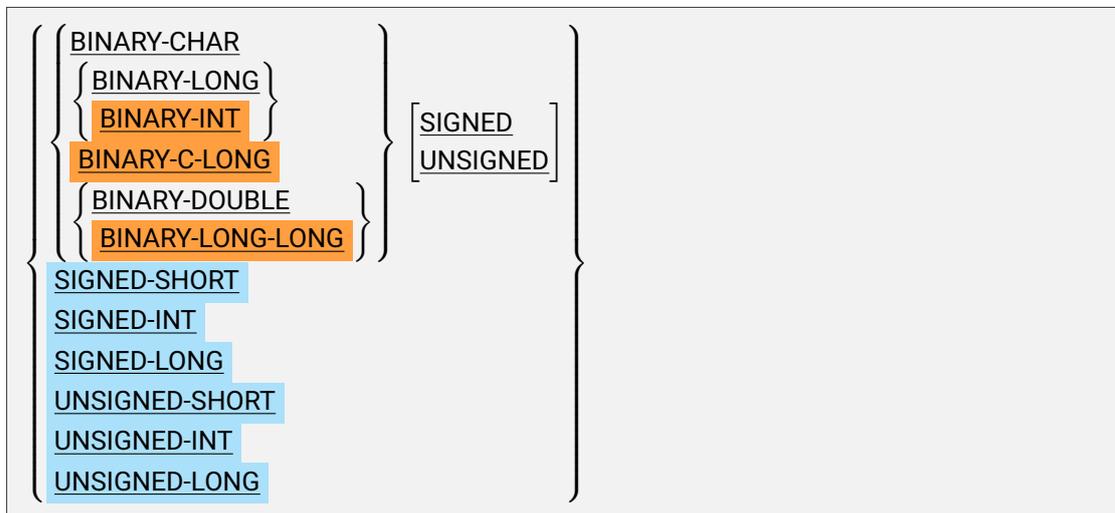
General rules

7.10.76. USAGE clause

The USAGE clause specifies the representation of a data item in memory.

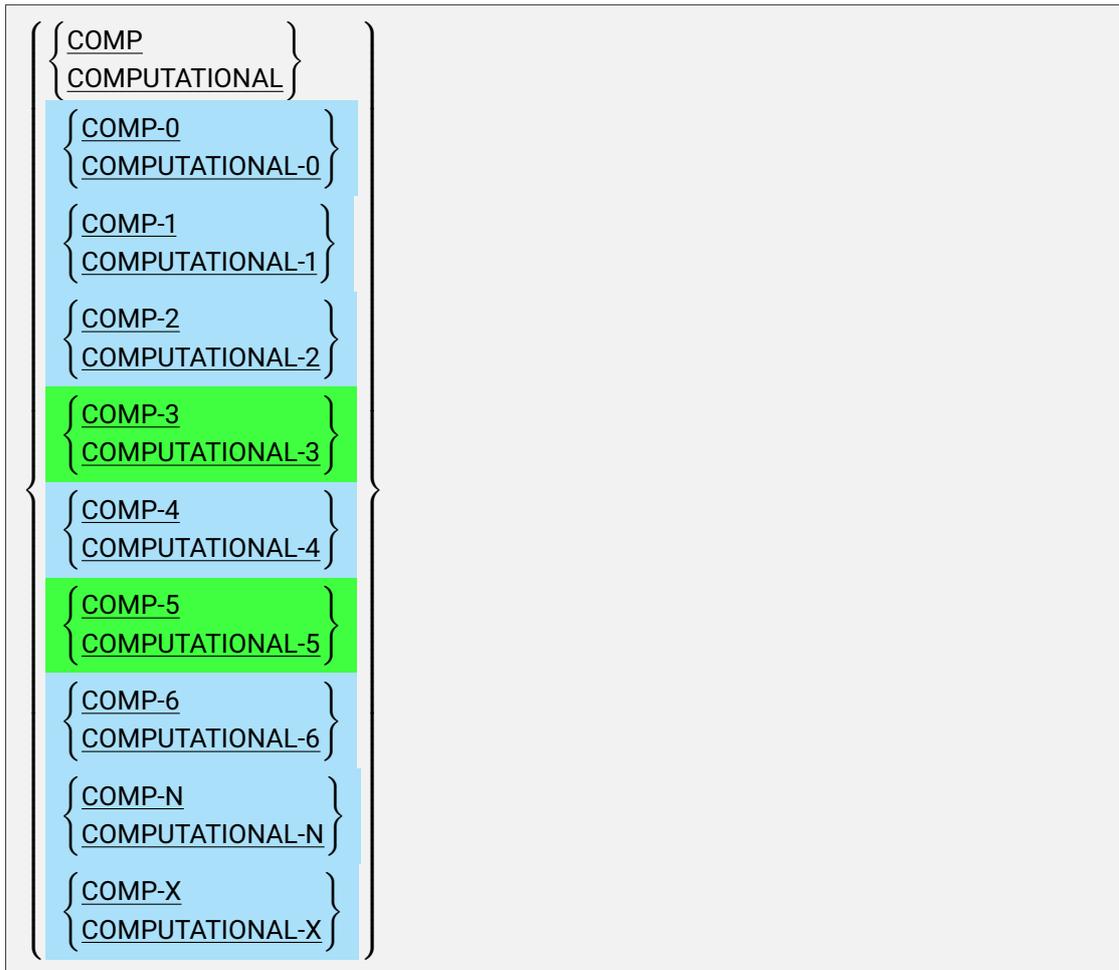


where *fixed-length-integers* is

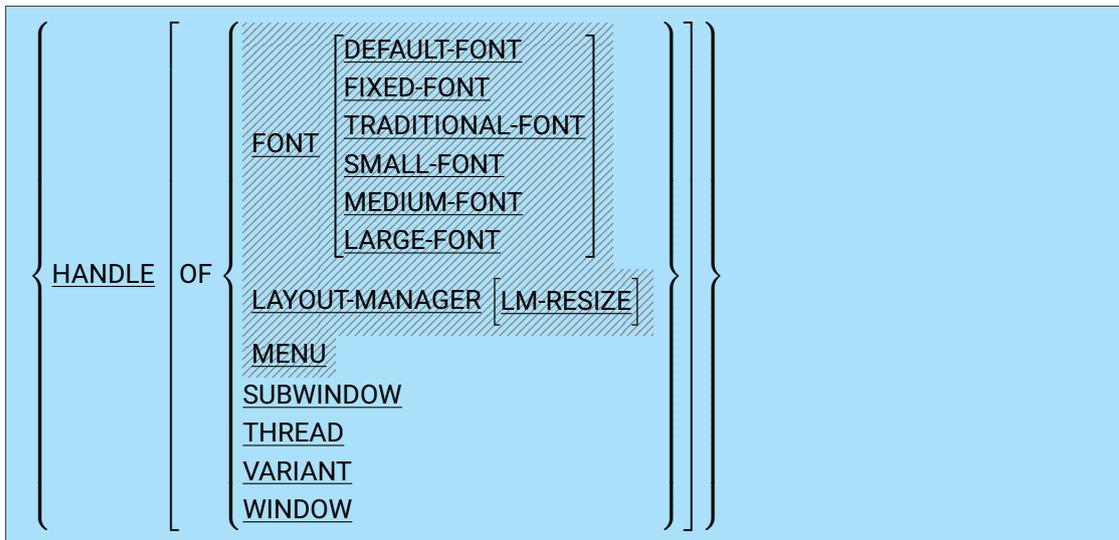


where *computation-usages* is

7. Data division



where *handle-usages* is



7. *Data division*

Syntax rules

General rules

7.10.77. USING clause

USING *identifier-1*

Syntax rules

General rules

7.10.78. VALUE clause

The VALUE clause specifies the initial value of the local-storage and working-storage section data items and the values to be used in INITIALIZE statements.

The VALUE clause for condition-names specifies the values under which a condition-name is true (or false).

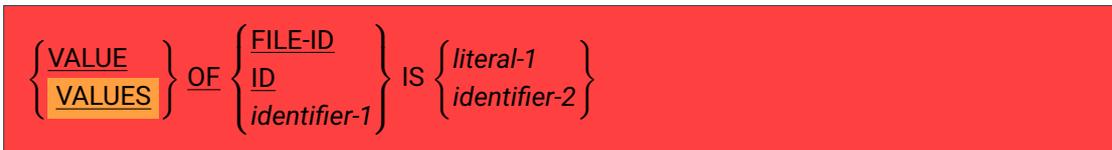
Format 1 (initialization)

$$\left\{ \begin{array}{l} \text{VALUE} \\ \text{VALUES} \end{array} \right\} \left[\begin{array}{l} \text{IS} \\ \text{ARE} \end{array} \right] \text{literal-1}$$
Format 2 (condition)

$$\left\{ \begin{array}{l} \text{VALUE} \\ \text{VALUES} \end{array} \right\} \left[\begin{array}{l} \text{IS} \\ \text{ARE} \end{array} \right] \left\{ \text{literal-2} \left[\begin{array}{l} \text{THROUGH} \\ \text{THRU} \end{array} \right] \text{literal-3} \right\} \dots$$

$$\left[\text{WHEN SET TO FALSE IS literal-4} \right]$$
Syntax rules**General rules**

7.10.79. VALUE OF clause



Syntax rules

General rules

7. Data division

7.10.80. VARYING clause

The VARYING clause declares counters to be used in printing repeated items in the report writer.

```
VARYING identifier-1 FROM number-1 BY number-2
```

Syntax rules

General rules

7.10.81. VOLATILE clause

The VOLATILE clause indicates an item may change in ways outside the control of the program.



VOLATILE

Syntax rules

General rules

8. Procedure division

8.1. Concepts

8.1.1. Exceptions

- EC-ALL
- EC-ARGUMENT
 - EC-ARGUMENT-FUNCTION
 - EC-ARGUMENT-IMP
- EC-BOUND
 - EC-BOUND-IMP
 - EC-BOUND-ODO
 - EC-BOUND-OVERFLOW
 - EC-BOUND-PTR
 - EC-BOUND-REF-MOD
 - EC-BOUND-SET
 - EC-BOUND-SUBSCRIPT
 - EC-BOUND-TABLE-LIMIT
- EC-DATA
 - EC-DATA-CONVERSION
 - EC-DATA-IMP
 - EC-DATA-INCOMPATIBLE
 - EC-DATA-OVERFLOW
 - EC-DATA-PTR-NULL
- EC-FLOW
 - EC-FLOW-GLOBAL-EXIT
 - EC-FLOW-GLOBAL-GOBACK
 - EC-FLOW-IMP
 - EC-FLOW-RELEASE

8. Procedure division

- EC-FLOW-REPORT
- EC-FLOW-RETURN
- EC-FLOW-SEARCH
- EC-FLOW-USE
- EC-FUNCTION
 - EC-FUNCTION-PTR-INVALID
 - EC-FUNCTION-PTR-NULL
- EC-I-O
 - EC-I-O-AT-END
 - EC-I-O-EOP
 - EC-I-O-EOP-OVERFLOW
 - EC-I-O-FILE-SHARING
 - EC-I-O-IMP
 - EC-I-O-INVALID-KEY
 - EC-I-O-LINAGE
 - EC-I-O-LOGIC-ERROR
 - EC-I-O-PERMANENT-ERROR
 - EC-I-O-RECORD-OPERATION
- EC-IMP
 - EC-IMP-ACCEPT
 - EC-IMP-DISPLAY
 - EC-IMP-FEATURE-DISABLED
 - EC-IMP-FEATURE-MISSING
 - EC-IMP-UTC-UNKNOWN
- EC-JSON
 - EC-JSON-IMP
- EC-LOCALE
 - EC-LOCALE-IMP
 - EC-LOCALE-INCOMPATIBLE
 - EC-LOCALE-INVALID
 - EC-LOCALE-INVALID-PTR
 - EC-LOCALE-MISSING

8. Procedure division

- EC-LOCALE-SIZE
- EC-OO
 - EC-OO-CONFORMANCE
 - EC-OO-EXCEPTION
 - EC-OO-FINALIZABLE
 - EC-OO-IMP
 - EC-OO-METHOD
 - EC-OO-NULL
 - EC-OO-RESOURCE
 - EC-OO-UNIVERSAL
- EC-ORDER
 - EC-ORDER-IMP
 - EC-ORDER-NOT-SUPPORTED
- EC-OVERFLOW
 - EC-OVERFLOW-IMP
 - EC-OVERFLOW-STRING
 - EC-OVERFLOW-UNSTRING
- EC-PROGRAM
 - EC-PROGRAM-ARG-MISMATCH
 - EC-PROGRAM-ARG-OMITTED
 - EC-PROGRAM-CANCEL-ACTIVE
 - EC-PROGRAM-IMP
 - EC-PROGRAM-NOT-FOUND
 - EC-PROGRAM-PTR-NULL
 - EC-PROGRAM-RECURSIVE-CALL
 - EC-PROGRAM-RESOURCES
- EC-RAISING
 - EC-RAISING-IMP
 - EC-RAISING-NOT-SPECIFIED
- EC-RANGE
 - EC-RANGE-IMP
 - EC-RANGE-INDEX

8. Procedure division

- EC-RANGE-INSPECT-SIZE
- EC-RANGE-INVALID
- EC-RANGE-PERFORM-VARYING
- EC-RANGE-PTR
- EC-RANGE-SEARCH-INDEX
- EC-RANGE-SEARCH-NO-MATCH
- EC-REPORT
 - EC-REPORT-ACTIVE
 - EC-REPORT-COLUMN-OVERLAP
 - EC-REPORT-FILE-MODE
 - EC-REPORT-IMP
 - EC-REPORT-INACTIVE
 - EC-REPORT-LINE-OVERLAP
 - EC-REPORT-NOT-TERMINATED
 - EC-REPORT-PAGE-LIMIT
 - EC-REPORT-PAGE-WIDTH
 - EC-REPORT-SUM-SIZE
 - EC-REPORT-VARYING
- EC-SCREEN
 - EC-SCREEN-FIELD-OVERLAP
 - EC-SCREEN-IMP
 - EC-SCREEN-ITEM-TRUNCATED
 - EC-SCREEN-LINE-NUMBER
 - EC-SCREEN-STARTING-COLUMN
- EC-SIZE
 - EC-SIZE-ADDRESS
 - EC-SIZE-EXPONENTIATION
 - EC-SIZE-IMP
 - EC-SIZE-OVERFLOW
 - EC-SIZE-TRUNCATION
 - EC-SIZE-UNDERFLOW
 - EC-SIZE-ZERO-DIVIDE

8. Procedure division

- EC-SORT-MERGE
 - EC-SORT-MERGE-ACTIVE
 - EC-SORT-MERGE-FILE-OPEN
 - EC-SORT-MERGE-IMP
 - EC-SORT-MERGE-RELEASE
 - EC-SORT-MERGE-RETURN
 - EC-SORT-MERGE-SEQUENCE
- EC-STORAGE
 - EC-STORAGE-IMP
 - EC-STORAGE-NOT-ALLOC
 - EC-STORAGE-NOT-AVAIL
- EC-USER
- EC-VALIDATE
 - EC-VALIDATE-CONTENT
 - EC-VALIDATE-FORMAT
 - EC-VALIDATE-IMP
 - EC-VALIDATE-RELATION
 - EC-VALIDATE-VARYING
- EC-XML
 - EC-XML-CODESET
 - EC-XML-CODESET-CONVERSION
 - EC-XML-COUNT
 - EC-XML-DOCUMENT-TYPE
 - EC-XML-IMPLICIT-CLOSE
 - EC-XML-INVALID
 - EC-XML-NAMESPACE
 - EC-XML-STACKED-OPEN
 - EC-XML-RANGE

8.2. Procedure division header

```

PROCEDURE DIVISION [call-convention-phrase] [linkage-phrase]
[using-chaining-clause] [RETURNING {identifier-1
OMITTED}].
[declaratives]
[section-name-2 SECTION.
paragraph-name-2.
imperative-statement-1 .] ...

```

where *using-chaining-clause* is

```

{ USING
  CHAINING }
{ BY { REFERENCE
      VALUE } [param-size] [OPTIONAL] identifier-2 [memory-size] } ...

```

where *param-size* is

```

[ [UNSIGNED] SIZE IS { AUTO
integer-1 } ]
SIZE IS DEFAULT

```

where *memory-size* is

```

WITH MEMORY SIZE IS [identifier-3
literal-1]

```

where *declaratives* is

```

DECLARATIVES.
[ section-name-1 SECTION. use-statement [paragraph-name-2.
imperative-statement-2 .] ... ] ...
END DECLARATIVES.

```

Syntax rules

General rules

8.3. Common phrases

8.3.1. Call-convention phrase

<p>[<u>C</u> <u>EXTERN</u> <u>PASCAL</u> <u>STATIC</u> <u>STDCALL</u> <i>mnemonic-name-1</i>]</p>

Syntax rules

General rules

8.3.2. LINKAGE phrase

WITH

<u>C</u>
<u>PASCAL</u>
<u>STDCALL</u>

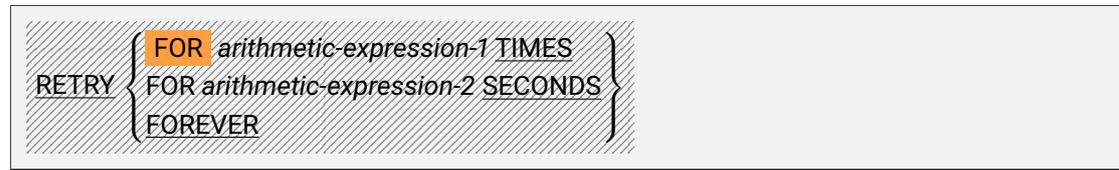
LINKAGE

Syntax rules

General rules

8. Procedure division

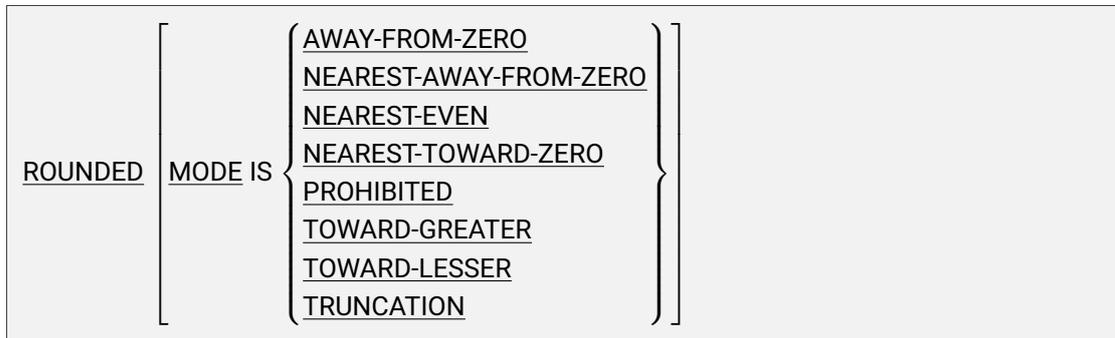
8.3.3. RETRY phrase



Syntax rules

General rules

8.3.4. **ROUNDED** phrase



Syntax rules

General rules

8.3.5. SIZE phrase

SIZE IS AUTO
SIZE IS DEFAULT
SIZE IS integer-1
UNSIGNED SIZE IS AUTO
UNSIGNED SIZE IS integer-2

Syntax rules

General rules

8.4. ACCEPT statement

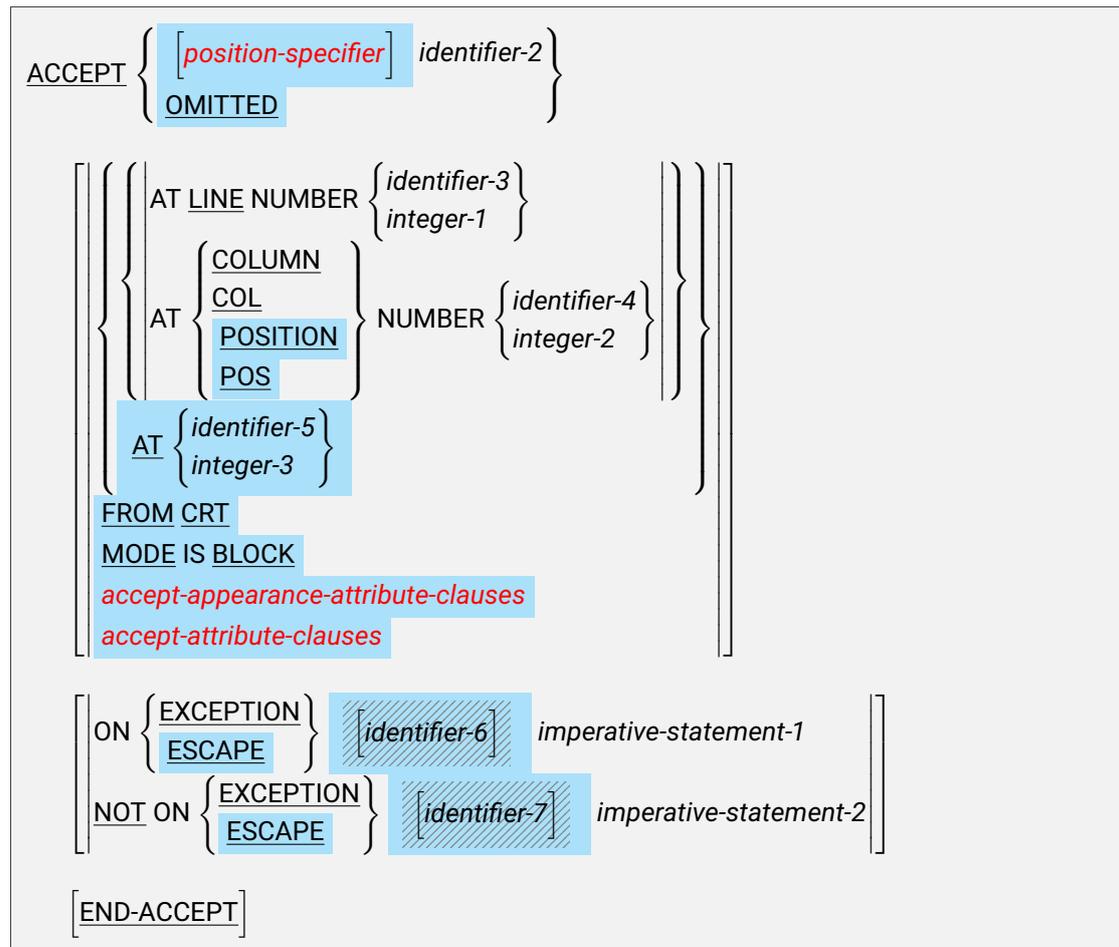
The ACCEPT statement transfers data provided by the user or the operating system to the specified data item.

Format 1 (device)

```
ACCEPT { identifier-1 } [FROM mnemonic-name-1] [END-ACCEPT]
```

Format 2 (screen)

8. Procedure division



where *position-specifier* is

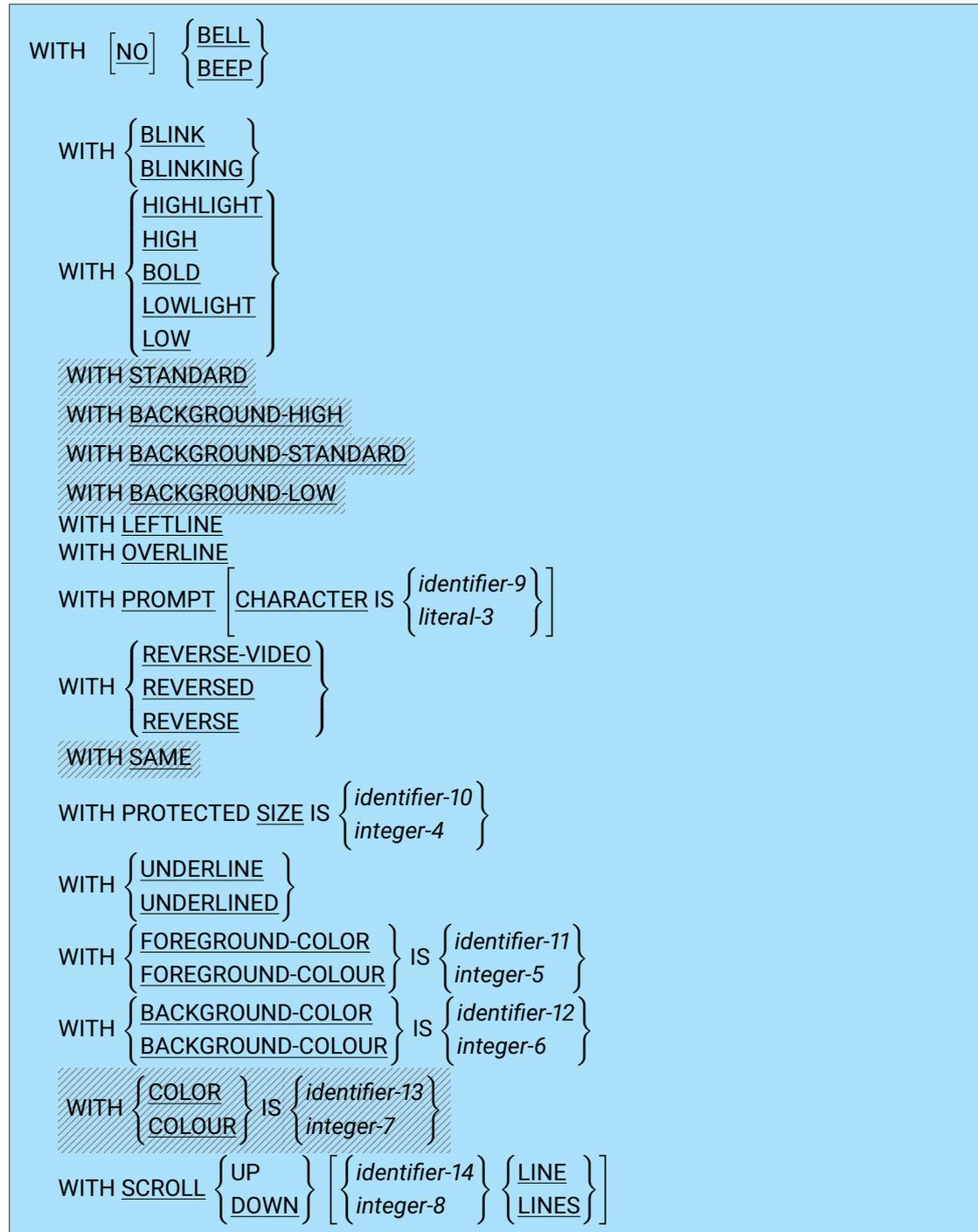
$$\left\{ \begin{array}{l} (\textit{position-specifier-num} , \textit{position-specifier-num}) \\ (, \textit{position-specifier-num}) \\ (\textit{position-specifier-num} ,) \end{array} \right\}$$

where *position-specifier-num* is

$$\left\{ \begin{array}{l} \textit{identifier-8} \\ \textit{literal-1} \end{array} \right\} \left[\left[\begin{array}{l} + \\ - \end{array} \right] \textit{literal-2} \right]$$

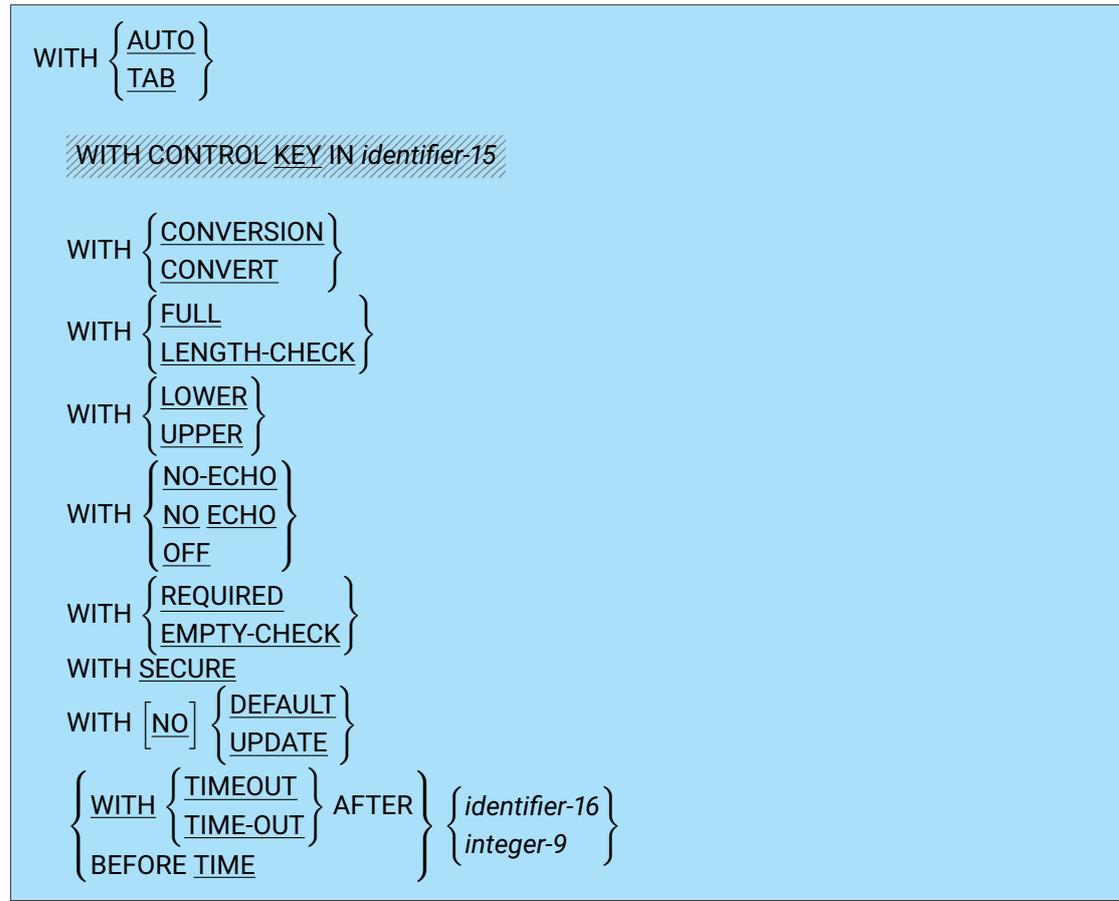
where *accept-appearance-attribute-clauses* is

8. Procedure division

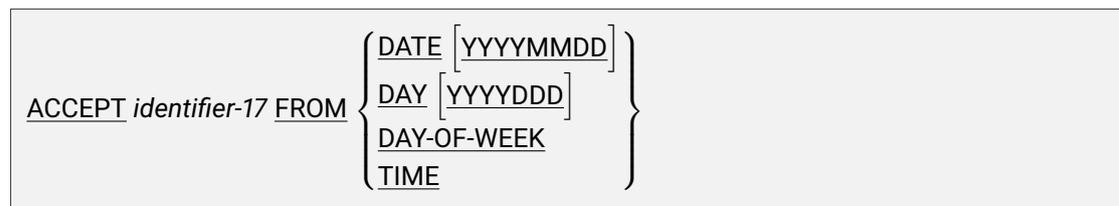


where *accept-attribute-clauses* is

8. Procedure division

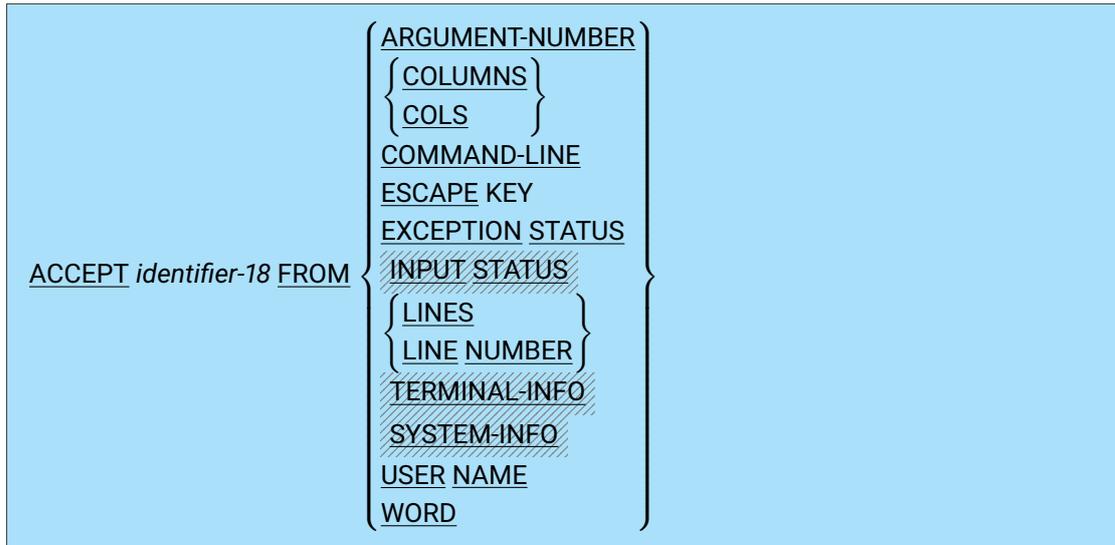


Format 3 (temporal)

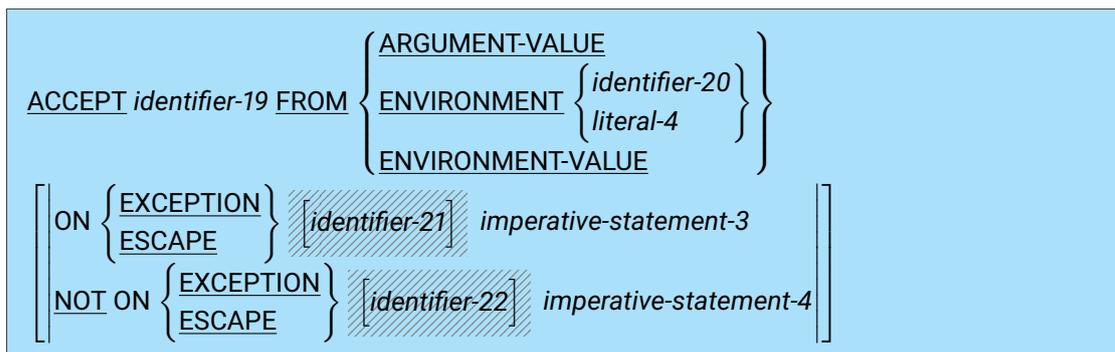


Format 4 (environment)

8. Procedure division



Format 5 (environment-exception)



Format 6 (message)



Format 7 (from screen)

8.5. ADD statement

The ADD statement adds two or more numbers and stores the result.

Format 1 (simple)

```

ADD { identifier-1
      literal-1 } ... TO { identifier-2 } ...

[ ON SIZE ERROR imperative-statement-1
  NOT ON SIZE ERROR imperative-statement-2 ]

[ END-ADD ]

```

Format 2 (giving)

```

ADD { identifier-3
      literal-2 } ... [ TO [ identifier-4 ] ... ]

GIVING { identifier-5 [ rounded-phrase ] } ...

[ ON SIZE ERROR imperative-statement-3
  NOT ON SIZE ERROR imperative-statement-4 ]

[ END-ADD ]

```

Format 3 (corresponding)

8. Procedure division

```
ADD { CORRESPONDING  
CORR } identifier-6 TO identifier-7 [rounded-phrase]  
  
[ ON SIZE ERROR imperative-statement-5  
NOT ON SIZE ERROR imperative-statement-6 ]  
  
[END-ADD]
```

Format 4 (table)

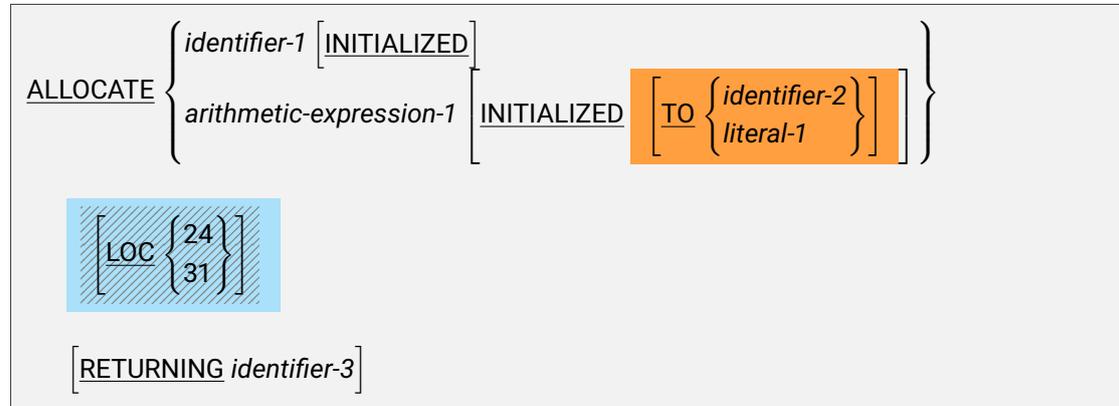
```
ADD TABLE identifier-8 TO identifier-9 [rounded-phrase]  
  
[ FROM INDEX integer-1 TO integer-2 ]  
  
[ DESTINATION INDEX integer-3 ]  
  
[ ON SIZE ERROR imperative-statement-7  
NOT ON SIZE ERROR imperative-statement-8 ]  
  
[END-ADD]
```

Syntax rules

General rules

8.6. ALLOCATE statement

The ALLOCATE statement requests memory from the operating system for a BASED data item or to be referenced by a data-pointer.



Syntax rules

General rules

8.7. ALTER statement

The ALTER statement changes the target of a GO TO statement. Its use is strongly discouraged and commonly proscribed.

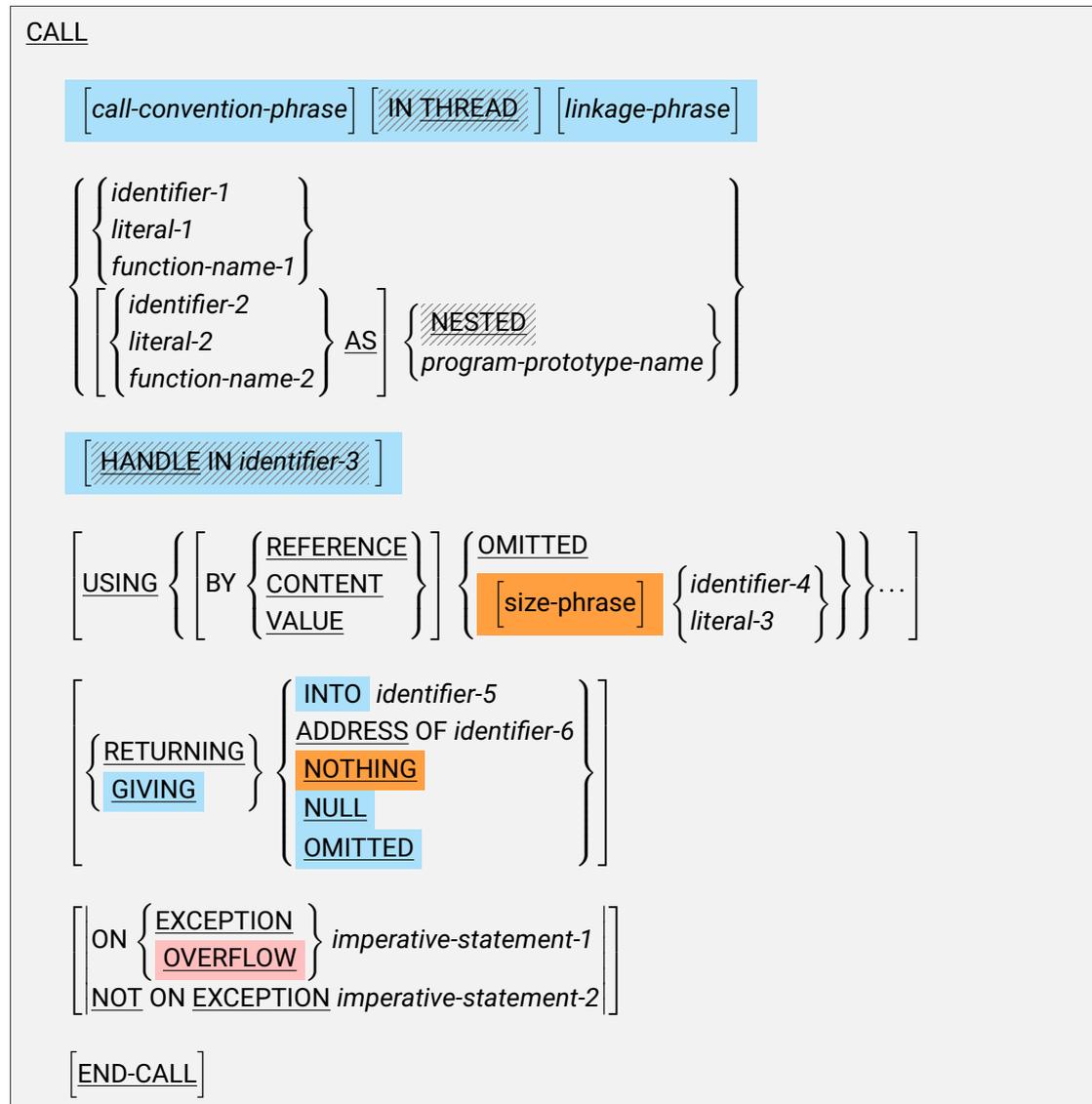
```
ALTER { procedure-name-1 TO PROCEED TO procedure-name-2 } ...
```

Syntax rules

General rules

8.8. CALL statement

The CALL statement transfers execution to another program, optionally with arguments and storing a return value.



Syntax rules

General rules

8.9. CANCEL statement

The CANCEL statement unloads a program from the operating system memory. This has the effect of freeing all the program's working-storage items and closing all files the program left open.

$\text{CANCEL } \left\{ \begin{array}{l} \text{identifier-1} \\ \text{literal-1} \end{array} \right\} \dots$
--

Syntax rules

General rules

8.10. CLOSE statement

The CLOSE statement prevents the program from accessing or modifying an open file. It first processes all pending writes for the file and releases all locks still held in the file.

Format 1 (file)

$\text{CLOSE} \left\{ \begin{array}{l} \text{file-name-1} \\ \left\{ \begin{array}{l} \text{REEL} \\ \text{UNIT} \end{array} \right\} \left[\text{FOR REMOVAL} \right] \\ \text{WITH NO REWIND} \\ \text{WITH LOCK} \end{array} \right\} \dots$
--

Format 2 (window)

$\text{CLOSE WINDOW } \text{identifier-1} \left[\text{WITH NO DISPLAY} \right]$
--

Syntax rules

General rules

8.11. COMMIT statement

The COMMIT statement forces all pending file writes to be processed.

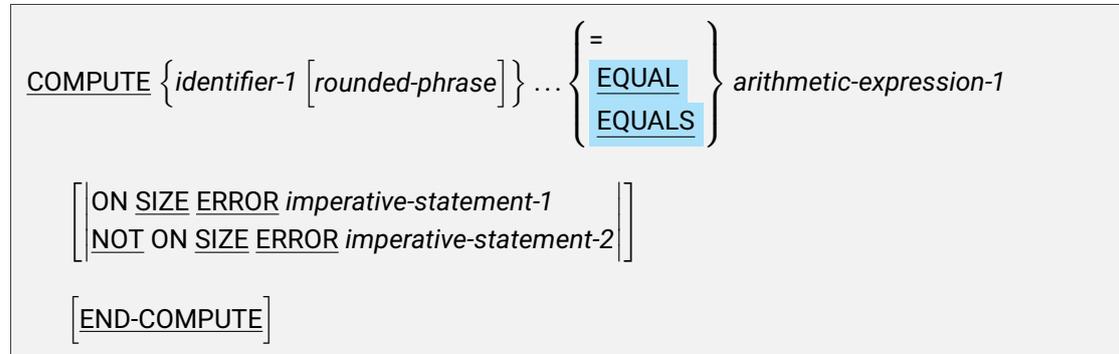
COMMIT

Syntax rules

General rules

8.12. COMPUTE statement

The COMPUTE statement evaluates an arithmetic expression and stores the result.



Syntax rules

General rules

8.13. CONTINUE statement

The plain format of the CONTINUE statement has no effect on program execution. The after format of the CONTINUE statement pauses execution for a specified length of time.

Format 1 (plain)

```
CONTINUE
```

Format 2 (after)

```
CONTINUE AFTER expression-1 SECONDS
```

Syntax rules

General rules

8.14. DELETE statement

The DELETE statement removes a record from a file or removes an entire file.

Format 1 (record)

```
DELETE file-name-1 RECORD
```

```
[retry-phrase]
```

```
[ [INVALID KEY imperative-statement-1  
  NOT INVALID KEY imperative-statement-2 ] ]
```

```
[END-DELETE]
```

Format 2 (file)

```
DELETE FILE { file-name-2 } ... [END-DELETE]
```

Syntax rules

General rules

8.15. DESTROY statement

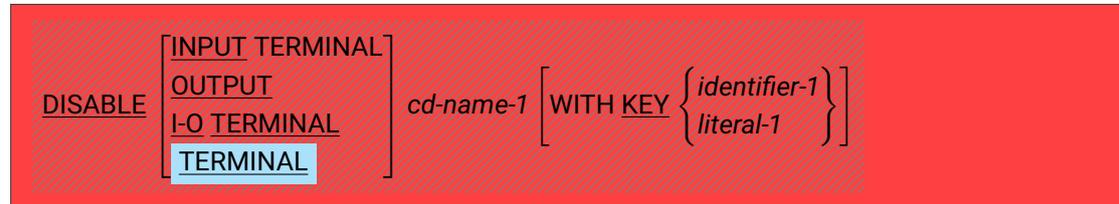
`DESTROY` { `ALL CONTROLS` }
`identifier-1 ...`

Syntax rules

General rules

8.16. DISABLE statement

The DISABLE statement prevents the program from modifying or accessing an enabled communication descriptor.



Syntax rules

General rules

8.17. DISPLAY statement

The DISPLAY statement displays data to a user or sends data to the operating system.

Format 1 (device)

```

DISPLAY { identifier-1 } ... [ UPON mnemonic-name-1 ]
      { literal-1 }      [ WITH NO ADVANCING ]

[ ON EXCEPTION imperative-statement-1 ]
[ NOT ON EXCEPTION imperative-statement-2 ]

[ END-DISPLAY ]

```

Format 2 (environment)

```

DISPLAY { identifier-2 } UPON { ARGUMENT-NUMBER
      { literal-2 }          { COMMAND-LINE
                              { ENVIRONMENT-NAME
                              { ENVIRONMENT-VALUE }
                              }
                              }
                              }

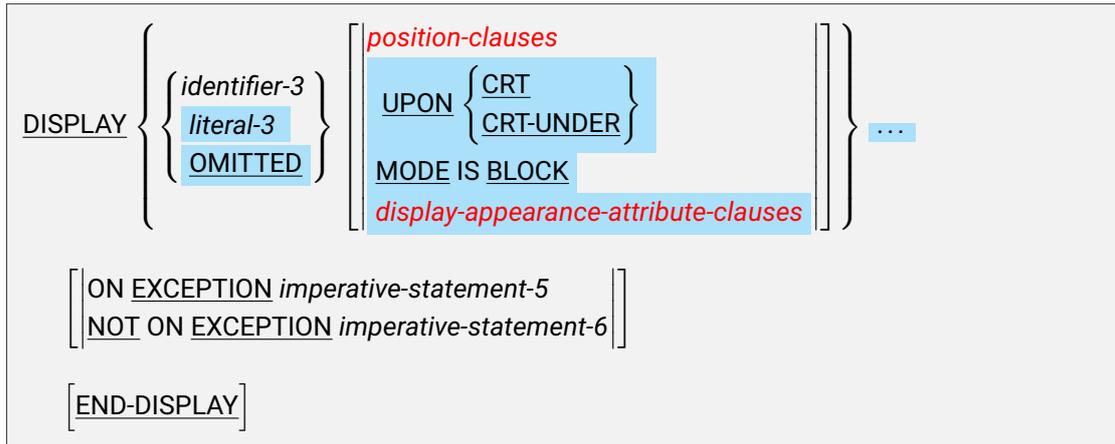
[ ON EXCEPTION imperative-statement-3 ]
[ NOT ON EXCEPTION imperative-statement-4 ]

[ END-DISPLAY ]

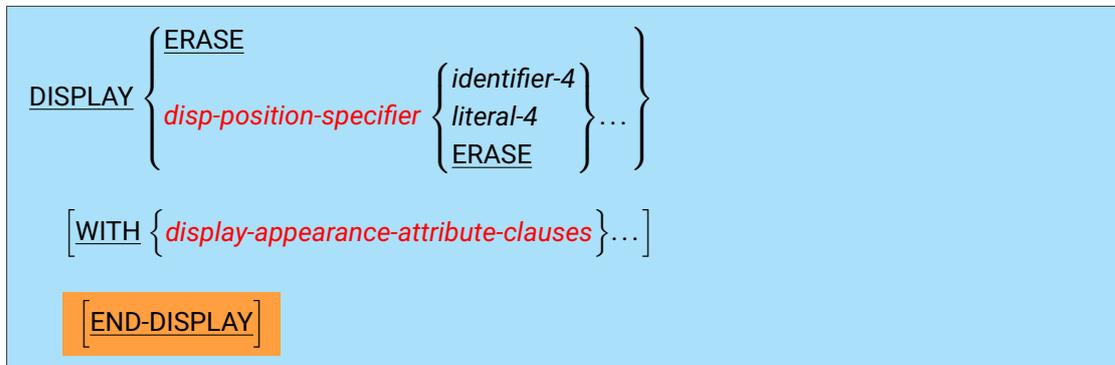
```

Format 3 (screen)

8. Procedure division

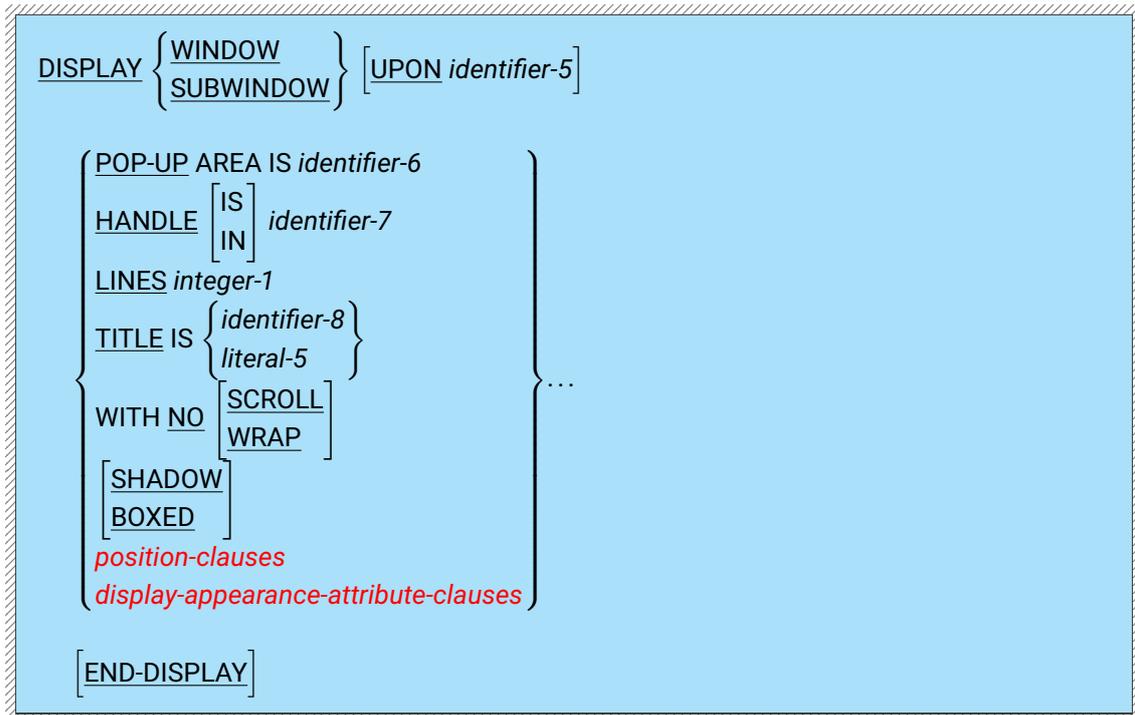


Format 4 (Microsoft screen)

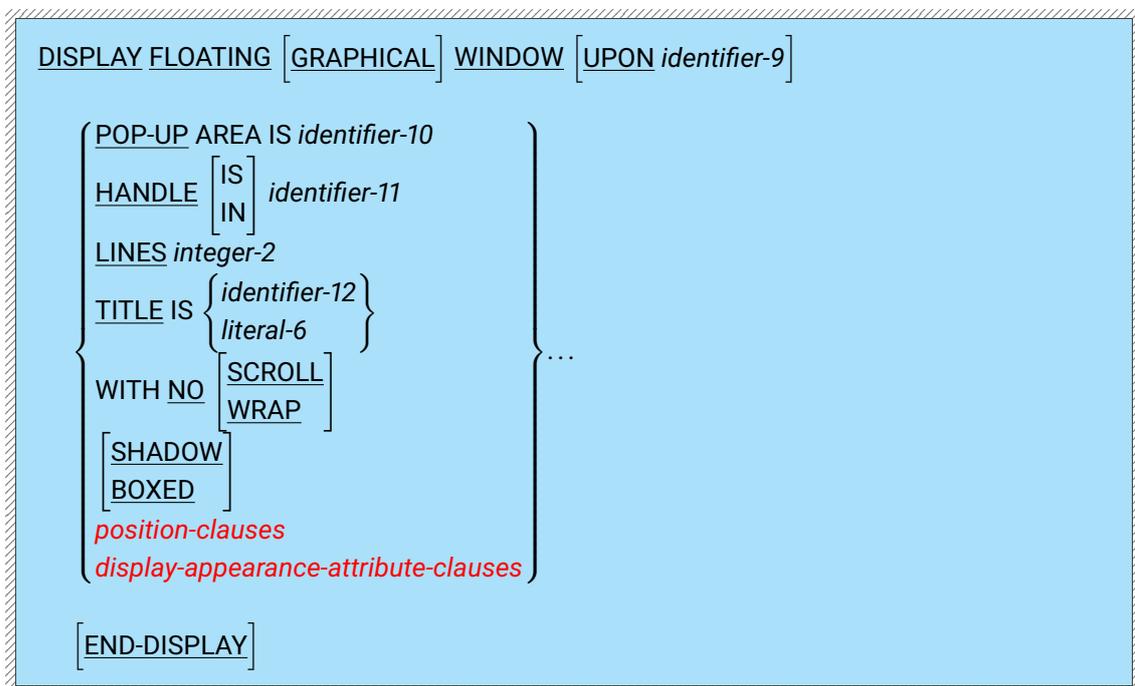


Format 5 (ordinary window)

8. Procedure division

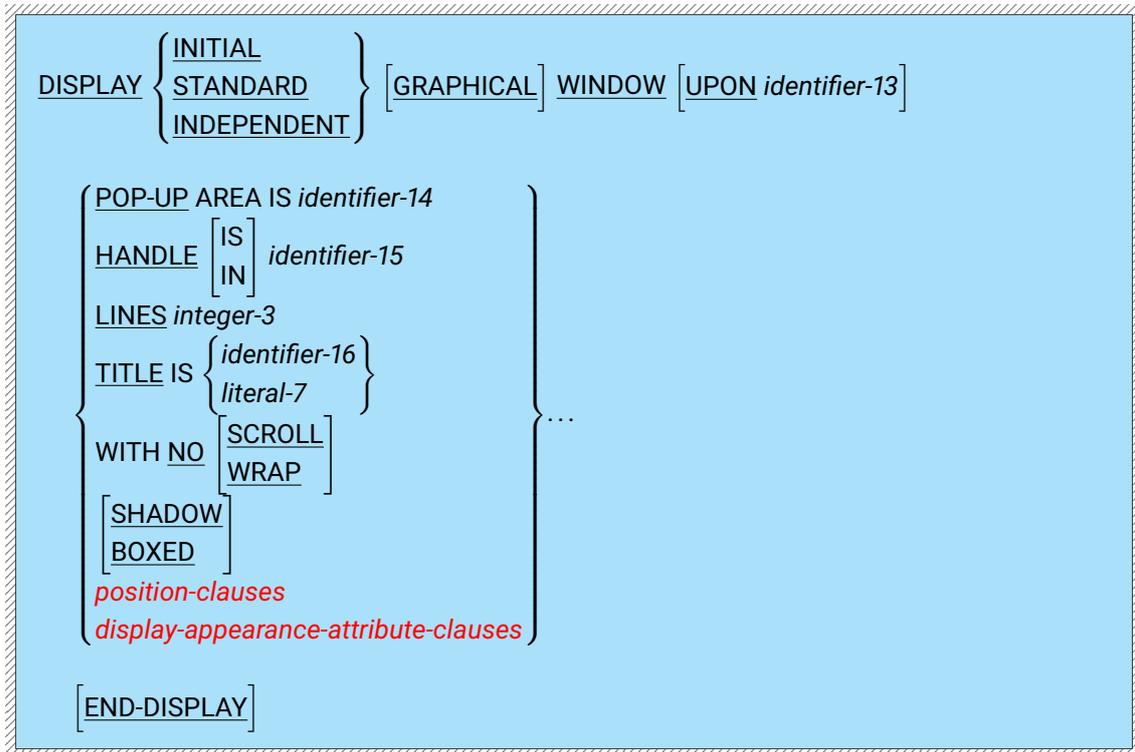


Format 6 (floating window)



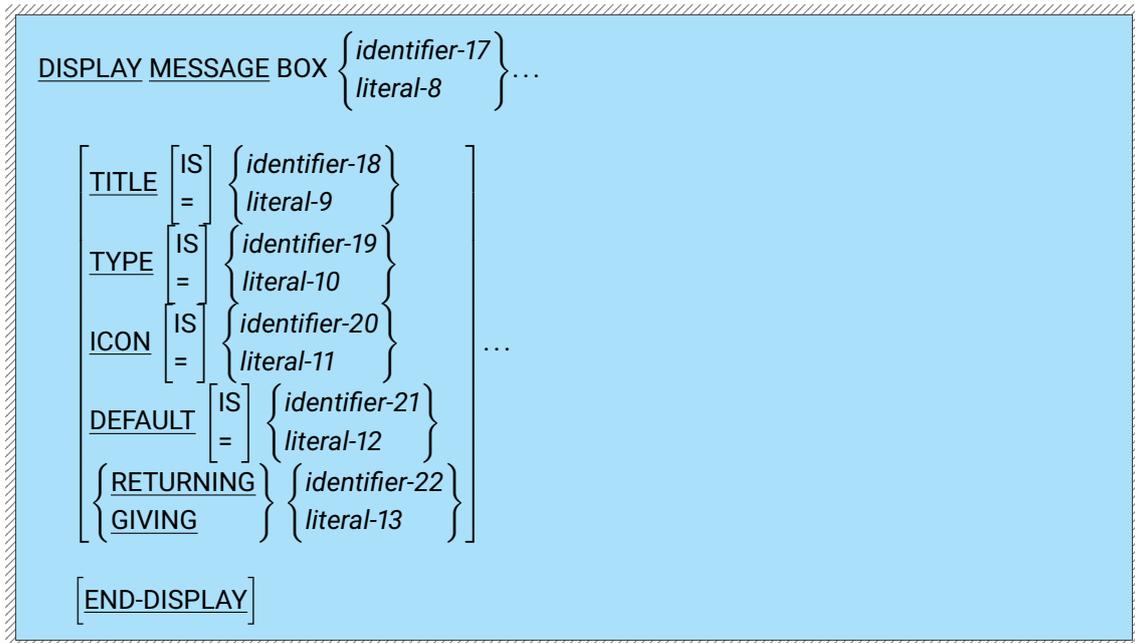
8. Procedure division

Format 7 (special window)

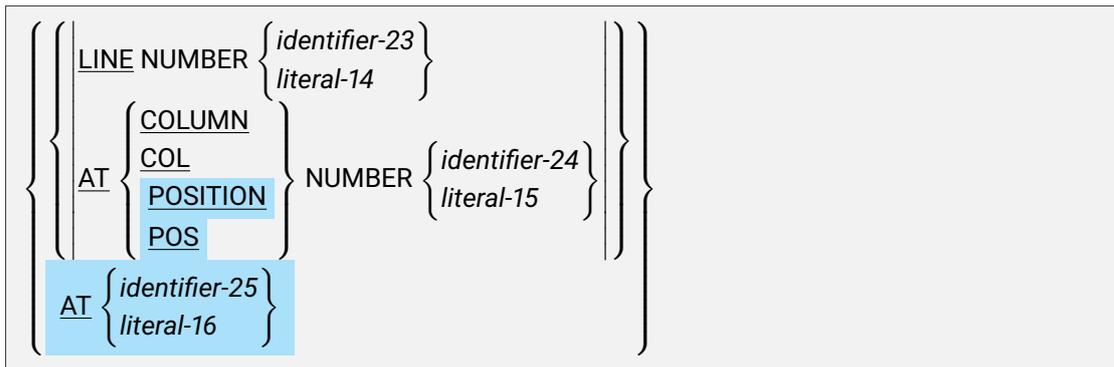


Format 8 (message box)

8. Procedure division



where *position-clauses* is



where *display-appearance-attribute-clauses* is

8. Procedure division

WITH	{	BELL	}				
		BEEP	}				
WITH	BLANK	{	LINE	}			
			SCREEN	}			
WITH	{	BLINK	}				
		BLINKING	}				
WITH	{	CONVERSION	}				
		CONVERT	}				
WITH	ERASE	{	EOL	}			
			EOS	}			
		[TO	END	OF]	{	LINE
							SCREEN
						}	
WITH	{	HIGHLIGHT	}				
		HIGH	}				
		BOLD	}				
		LOWLIGHT	}				
		LOW	}				
WITH	STANDARD						
WITH	BACKGROUND-HIGH						
WITH	BACKGROUND-STANDARD						
WITH	BACKGROUND-LOW						
WITH	OVERLINE						
WITH	{	REVERSE-VIDEO	}				
		REVERSED	}				
		REVERSE	}				
WITH	SAME						
WITH	SIZE	IS	{	identifier-26	}		
				literal-17	}		
WITH	{	UNDERLINE	}				
		UNDERLINED	}				
WITH	{	FOREGROUND-COLOR	}	[IS]	{
		FOREGROUND-COLOUR	}		=		{
							identifier-27
							integer-4
WITH	{	BACKGROUND-COLOR	}	[IS]	{
		BACKGROUND-COLOUR	}		=		{
							identifier-28
							integer-5
WITH	{	COLOR	}	IS	{	identifier-29	}
		COLOUR	}			integer-6	}
WITH	SCROLL	{	UP	}	[identifier-30]
			DOWN	}		integer-7]
							{
							LINE
							LINES
							}

8. Procedure division

where *disp-position-specifier* is

$$\left\{ \begin{array}{l} (\textit{disp-position-specifier-num} , \textit{disp-position-specifier-num}) \\ (, \textit{disp-position-specifier-num}) \\ (\textit{disp-position-specifier-num} ,) \end{array} \right\}$$

where *disp-position-specifier-num* is

$$\left\{ \begin{array}{l} \textit{identifier-31} \\ \textit{literal-18} \end{array} \right\} \left[\left\{ \begin{array}{l} + \\ - \end{array} \right\} \textit{literal-19} \right]$$

Syntax rules

General rules

8.18. DIVIDE statement

The DIVIDE statement divides one or more numbers by another and stores the results.

Format 1 (into)

```

DIVIDE { identifier-1 } INTO { { identifier-2 } [rounded-phrase] } ...
      { literal-1 }
      { literal-2 }

[ ON SIZE ERROR imperative-statement-1
  NOT ON SIZE ERROR imperative-statement-2 ]

[ END-DIVIDE ]

```

Format 2 (giving)

```

DIVIDE { identifier-3 } { BY } { identifier-4 }
      { literal-3 }     { INTO } { literal-4 }

GIVING { { identifier-5 } [rounded-phrase] } ...
       { literal-5 }

[ REMAINDER { identifier-6 } ]
           { literal-6 }

[ ON SIZE ERROR imperative-statement-3
  NOT ON SIZE ERROR imperative-statement-4 ]

[ END-DIVIDE ]

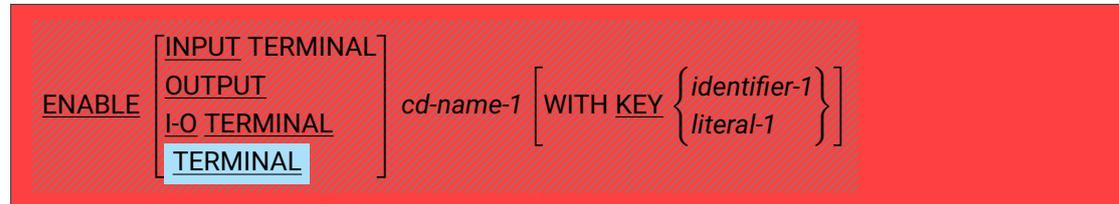
```

Syntax rules

General rules

8.19. ENABLE statement

The ENABLE statement allows the program to access and modify a communication descriptor.



Syntax rules

General rules

8.20. ENTRY statement

The ENTRY statement indicates an alternative point of entry into the program.

Format 1 (usual)

```

ENTRY [call-convention-phrase] literal-1
    [linkage-phrase]
    [ USING { [ BY { REFERENCE } ] { OMITTED } { [size-phrase] { identifier-1 } } ... } ]
    [ VALUE ] { literal-2 } ]

```

Format 2 (for GO TO)

```

ENTRY FOR GO TO literal-3

```

Syntax rules

General rules

8.21. EVALUATE statement

The EVALUATE statement evaluates one or more conditions and execute the statements corresponding to the first true condition.

$$\text{EVALUATE } \left\{ \begin{array}{l} \text{expression-1} \\ \text{TRUE} \\ \text{FALSE} \end{array} \right\} \left[\text{ALSO } \left\{ \begin{array}{l} \text{expression-2} \\ \text{TRUE} \\ \text{FALSE} \end{array} \right\} \right] \dots$$

$$\left\{ \text{WHEN } \textit{selection-object} \left[\text{ALSO } \textit{selection-object} \right] \dots \textit{imperative-statement-1} \right\} \dots$$

$$\left[\text{WHEN OTHER } \textit{imperative-statement-2} \right]$$

$$\left[\text{END-EVALUATE} \right]$$

where *selection-object* is

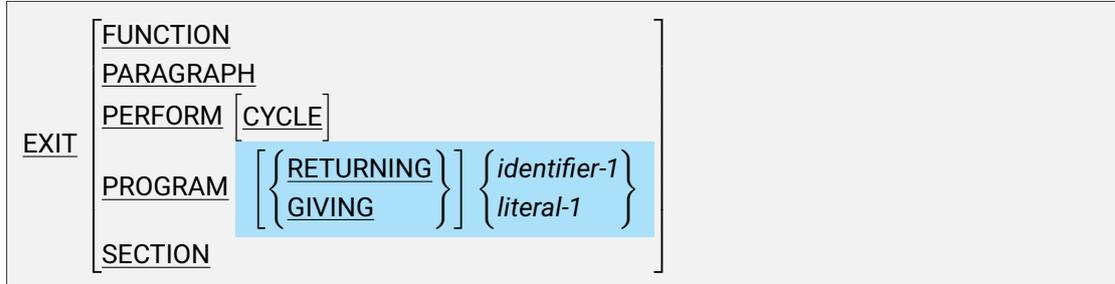
$$\left\{ \begin{array}{l} \textit{partial-expression-1} \left[\left\{ \begin{array}{l} \text{THROUGH} \\ \text{THRU} \end{array} \right\} \textit{expression-3} \right] \\ \text{ANY} \\ \text{TRUE} \\ \text{FALSE} \end{array} \right\}$$

Syntax rules

General rules

8.22. EXIT statement

The EXIT statement indicates the end of a path of execution.



Syntax rules

General rules

8.23. FREE statement

The FREE statement returns memory to the operating system.

```
FREE { identifier-1 } ...
```

Syntax rules

General rules

8.24. GENERATE statement

The GENERATE statement a specified report entry.

```
GENERATE report-name-1
```

Syntax rules

General rules

8.25. GO TO statement

The GO TO statement transfer execution to another part of the program.

$\text{GO TO } \left\{ \begin{array}{l} \text{procedure-name-1} \dots \\ \text{ENTRY entry-name} \dots \end{array} \right\} \left[\text{DEPENDING ON identifier-1} \right]$
--

Syntax rules

General rules

8.26. GOBACK statement

The GOBACK statement terminates execution in the program, returning control to the calling program or, if no such program exists, to the operating system.

$\text{GOBACK} \left[\left\{ \begin{array}{l} \text{RETURNING} \\ \text{GIVING} \end{array} \right\} \left\{ \begin{array}{l} \text{identifier-1} \\ \text{literal-1} \end{array} \right\} \right]$
--

Syntax rules

General rules

8.27. IF statement

The IF statement evaluates a condition and executes statements depending on whether the condition was true or false.

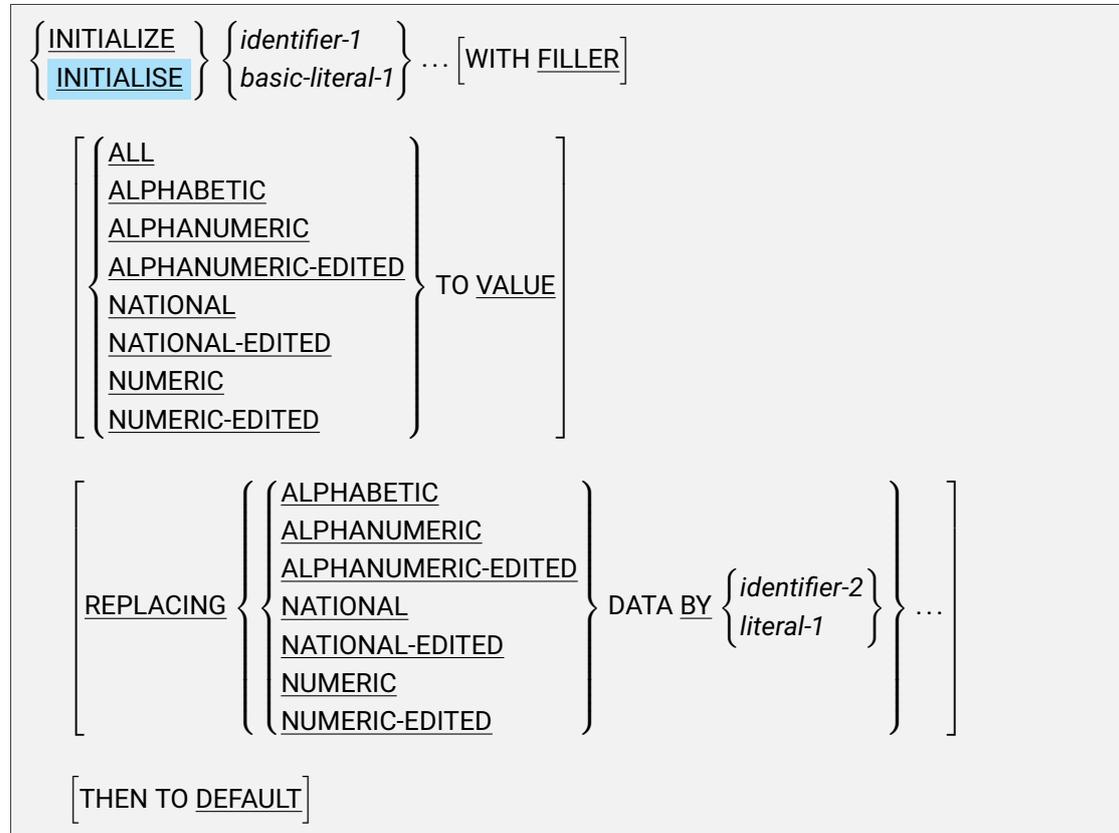
```
IF condition THEN { imperative-statement-1 }  
                   { ELSE imperative-statement-2 } ...  
  
[ END-IF ]
```

Syntax rules

General rules

8.28. INITIALIZE statement

The INITIALIZE statement sets data items to their default values.



Syntax rules

General rules

8.29. INITIATE statement

The INITIATE statement allows the program to begin generating the specified report.

```
INITIATE {report-name-1} ...
```

Syntax rules

General rules

8.30. INQUIRE statement

Format 1 (control)

INQUIRE { identifier-1 } { { control-property } { [IN] identifier-2 } ... }
 { CONTROL } { LAYOUT-DATA } { = }

Format 2 (window)

INQUIRE { identifier-3 } { WINDOW [identifier-4] } { { LAYOUT-MANAGER } { [IN] identifier-5 } ... }
 { SIZE } { TITLE } { = }

Syntax rules

General rules

8.31. INSPECT statement

The INSPECT statement counts the number of occurrences of a character string, replaces occurrences or both.

$$\text{INSPECT} \left\{ \begin{array}{l} \text{identifier-1} \\ \text{literal-1} \\ \text{function-name-1} \end{array} \right\} \left\{ \begin{array}{l} \text{tallying-phrase} [\text{replacing-phrase}] \\ \text{replacing-phrase} \\ \text{converting-phrase} \end{array} \right\}$$

where *tallying-phrase* is

$$\text{TALLYING} \left\{ \left\{ \begin{array}{l} \text{identifier-2} \\ \text{literal-2} \end{array} \right\} \text{FOR} \left\{ \begin{array}{l} \text{CHARACTERS} \\ \text{ALL} \\ \text{LEADING} \\ \text{TRAILING} \end{array} \right\} \left\{ \begin{array}{l} \text{identifier-3} \\ \text{literal-3} \end{array} \right\} \right\} \dots [\text{before-after-phrase}] \dots$$

where *replacing-phrase* is

$$\text{REPLACING} \left\{ \left\{ \begin{array}{l} \text{CHARACTERS} \\ \text{ALL} \\ \text{LEADING} \\ \text{FIRST} \\ \text{TRAILING} \end{array} \right\} \left\{ \begin{array}{l} \text{identifier-4} \\ \text{literal-4} \end{array} \right\} \right\} \text{BY} \left\{ \begin{array}{l} \text{identifier-5} \\ \text{literal-5} \end{array} \right\} [\text{before-after-phrase}] \dots$$

where *converting-phrase* is

$$\text{CONVERTING} \left\{ \begin{array}{l} \text{identifier-6} \\ \text{literal-6} \end{array} \right\} \text{TO} \left\{ \begin{array}{l} \text{identifier-7} \\ \text{literal-7} \end{array} \right\} [\text{before-after-phrase}]$$

where *before-after-phrase* is

$$\left[\begin{array}{l} \text{BEFORE INITIAL} \left\{ \begin{array}{l} \text{identifier-8} \\ \text{literal-8} \end{array} \right\} \\ \text{AFTER INITIAL} \left\{ \begin{array}{l} \text{identifier-9} \\ \text{literal-9} \end{array} \right\} \end{array} \right]$$

Syntax rules

General rules

8.32. JSON GENERATE statement

The JSON GENERATE statement converts a JSON record based on the structure and content of the input data item.

```
JSON GENERATE identifier-1 FROM identifier-2
```

```
[COUNT IN identifier-3]
```

```
[NAME OF {identifier-4 IS literal-1}...]
```

```
[SUPPRESS {identifier-5} ...]
```

```
[ [ON EXCEPTION imperative-statement-1  
  NOT ON EXCEPTION imperative-statement-2 ] ]
```

```
[END-JSON]
```

Syntax rules

General rules

8.33. JSON PARSE statement

The JSON PARSE statement starts an event-driven JSON parser; the processing procedure is performed as each component of the JSON document is identified in order.

```
JSON PARSE identifier-1 INTO identifier-2
```

```
[ WITH DETAIL ]
```

```
[ NAME OF { identifier-3 IS literal-1 } ... ]
```

```
[ SUPPRESS { identifier-4 } ... ]
```

```
[ [ ON EXCEPTION imperative-statement-1  
  NOT ON EXCEPTION imperative-statement-2 ] ]
```

```
[ END-JSON ]
```

Syntax rules

General rules

8.34. MERGE statement

The MERGE statement reads multiple files with the same record description, combines their records and sorts them.

```

MERGE identifier-1 [ ON { ASCENDING } KEY [ identifier-2 ] ... ] ...
    [ WITH DUPLICATES [ IN ORDER ] ]
    [ COLLATING SEQUENCE IS { IS alphanumeric-collating-sequence [ national-collating-sequence ] }
      { FOR ALPHANUMERIC IS alphanumeric-collating-sequence }
      { FOR NATIONAL IS national-collating-sequence } } ]
    [ USING { file-name-1 } ... ]
    [ GIVING { file-name-2 } ... ]
    [ OUTPUT PROCEDURE IS procedure-name-1 [ { THROUGH } procedure-name-2 ]
      [ THRU ] ] ]

```

Syntax rules

General rules

8.35. MODIFY statement

Format 1 (control)

$\text{MODIFY } \left\{ \begin{array}{l} \text{identifier-1} \\ \text{CONTROL} \end{array} \right\} \left\{ \text{control-attributes} \right\} \dots \left[\text{END-MODIFY} \right]$

Format 2 (window)

$\text{MODIFY } \left\{ \begin{array}{l} \text{identifier-2} \\ \text{WINDOW } \left[\text{identifier-3} \right] \end{array} \right\} \left\{ \left\{ \begin{array}{l} \text{LAYOUT-MANAGER} \\ \text{SIZE} \\ \text{TITLE} \end{array} \right\} \left[\begin{array}{l} \text{IN} \\ = \end{array} \right] \text{identifier-4} \right\} \dots \left[\text{END-MODIFY} \right]$

Syntax rules

General rules

8.36. MOVE statement

The MOVE statement sets the value of one or more data items.

$\text{MOVE } \left[\begin{array}{l} \text{CORRESPONDING} \\ \text{CORR} \end{array} \right] \left\{ \begin{array}{l} \text{identifier-1} \\ \text{literal-1} \end{array} \right\} \text{TO } \left\{ \text{identifier-2} \right\} \dots$
--

Syntax rules

General rules

8.37. MULTIPLY statement

The MULTIPLY statement multiplies multiple numbers and stores the result.

Format 1 (simple)

```

MULTIPLY { identifier-1 }
          { literal-1 } BY { { identifier-2 }
                           { literal-2 } } [rounded-phrase] } ...

[ ON SIZE ERROR imperative-statement-1
  NOT ON SIZE ERROR imperative-statement-2 ]

[ END-MULTIPLY ]

```

Format 2 (giving)

```

MULTIPLY { identifier-3 }
          { literal-3 } BY { identifier-4 }
                           { literal-4 }

GIVING { { identifier-5 }
         { literal-5 } } [rounded-phrase] } ...

[ ON SIZE ERROR imperative-statement-3
  NOT ON SIZE ERROR imperative-statement-4 ]

[ END-MULTIPLY ]

```

Syntax rules**General rules**

8.38. NEXT SENTENCE statement

The NEXT SENTENCE statement transfers execution to the first statement following the current sentence.

NEXT SENTENCE

Syntax rules

General rules

8.39. OPEN statement

The OPEN statement allows the program to access or modify specified files.

`OPEN { open-body }...`

where *open-body* is

`[EXCLUSIVE] { INPUT
OUTPUT
I-O
EXTEND } [sharing-mode] [retry-phrase] { file-name-1 }... [open-options]`

where *sharing-mode* is

`SHARING WITH { ALL OTHER
NO OTHER
READ ONLY }`

where *open-options* is

`[FOR WITH] { LOCK
MASS-UPDATE
BULK-ADDITION }`
`ALLOWING { ALL
READERS
UPDATERS
WRITERS
NO OTHERS }`
`{ DISP
LEAVE
REREAD }`
`[WITH NO REWIND]
REVERSED]`

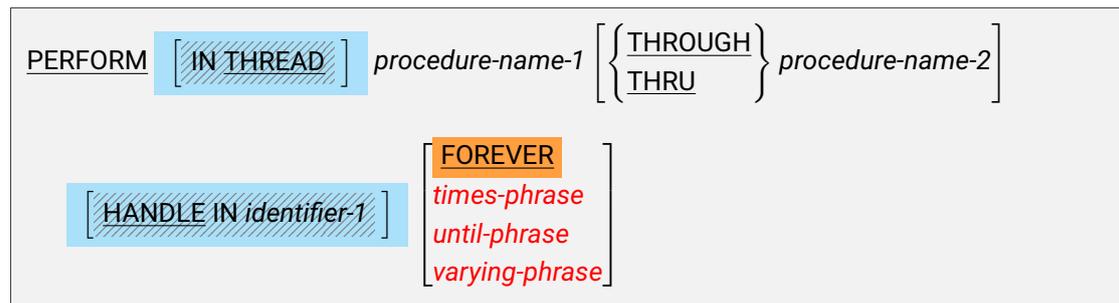
Syntax rules

General rules

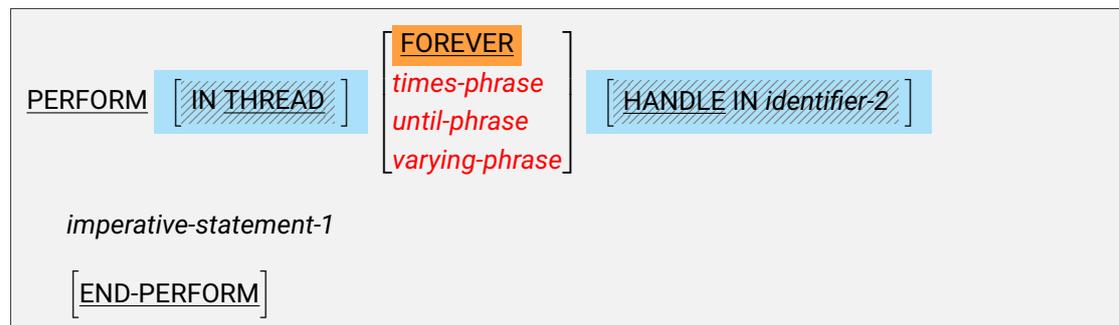
8.40. PERFORM statement

The PERFORM statement executes the specified procedures or statements one or more times.

Format 1 (procedure)



Format 2 (inline)



where *times-phrase* is



where *until-phrase* is



and where *varying-phrase* is

8. Procedure division

$$\left[\text{WITH TEST } \left\{ \begin{array}{l} \text{BEFORE} \\ \text{AFTER} \end{array} \right\} \right]$$
$$\text{VARYING } \textit{identifier-4} \text{ FROM } \left\{ \begin{array}{l} \textit{identifier-5} \\ \textit{literal-2} \end{array} \right\} \left[\text{BY } \left\{ \begin{array}{l} \textit{identifier-6} \\ \textit{literal-3} \end{array} \right\} \right] \text{UNTIL } \textit{condition-2}$$
$$\left[\text{AFTER } \textit{identifier-7} \text{ FROM } \left\{ \begin{array}{l} \textit{identifier-8} \\ \textit{literal-4} \end{array} \right\} \left[\text{BY } \left\{ \begin{array}{l} \textit{identifier-9} \\ \textit{literal-5} \end{array} \right\} \right] \text{UNTIL } \textit{condition-3} \right] \dots$$

Syntax rules

General rules

8.41. PURGE statement

The PURGE statement

```
PURGE cd-name-1
```

Syntax rules

General rules

8.42. RAISE statement

The RAISE statement activates an exception condition.

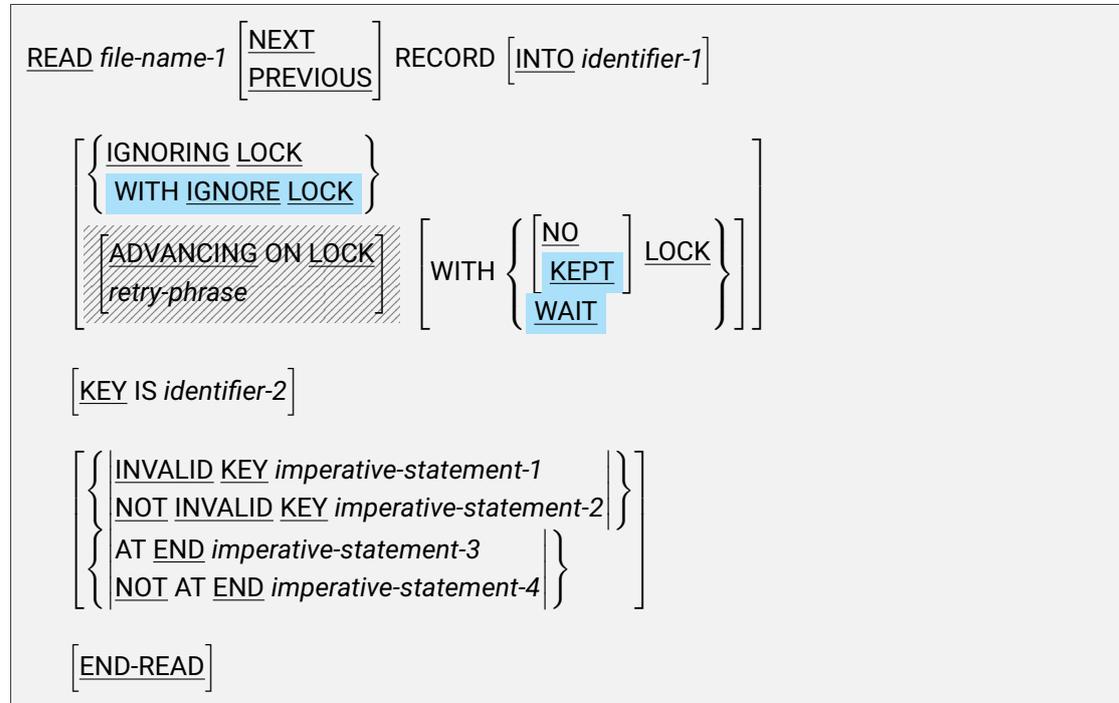
```
RAISE { EXCEPTION exception-name }  
      { identifier-1 }
```

Syntax rules

General rules

8.43. READ statement

The READ statement transfer data from a file to the file's record or to a data item.



Syntax rules

General rules

8.44. READY statement

The READY TRACE statement causes the name of procedures reached by execution to be displayed.

```
READY TRACE
```

Syntax rules

General rules

8.45. RECEIVE statement

The RECEIVE statement transfers data from a communication descriptor to a data item.

```
RECEIVE cd-name-1 { MESSAGE } INTO identifier-1  
                { SEGMENT }
```

```
[ WITH DATA imperative-statement-1 ]  
[ NO DATA imperative-statement-2 ]
```

```
[ END-RECEIVE ]
```

Syntax rules

General rules

8.46. RELEASE statement

The RELEASE statement provides a record for sorting.

$\text{RELEASE } \textit{identifier-1} \left[\text{FROM} \left\{ \begin{array}{l} \textit{identifier-2} \\ \textit{literal-1} \\ \textit{function-call-1} \end{array} \right\} \right]$
--

Syntax rules

General rules

8.47. RESET statement

The RESET TRACE stops the names of procedures reached by execution being displayed.

```
RESET TRACE
```

Syntax rules

General rules

8.48. RETURN statement

The RETURN statement retrieves records from the sorting process in order.

```
RETURN file-name-1 RECORD [INTO identifier-1]
```

```
AT END imperative-statement-1
```

```
[NOT AT END imperative-statement-2]
```

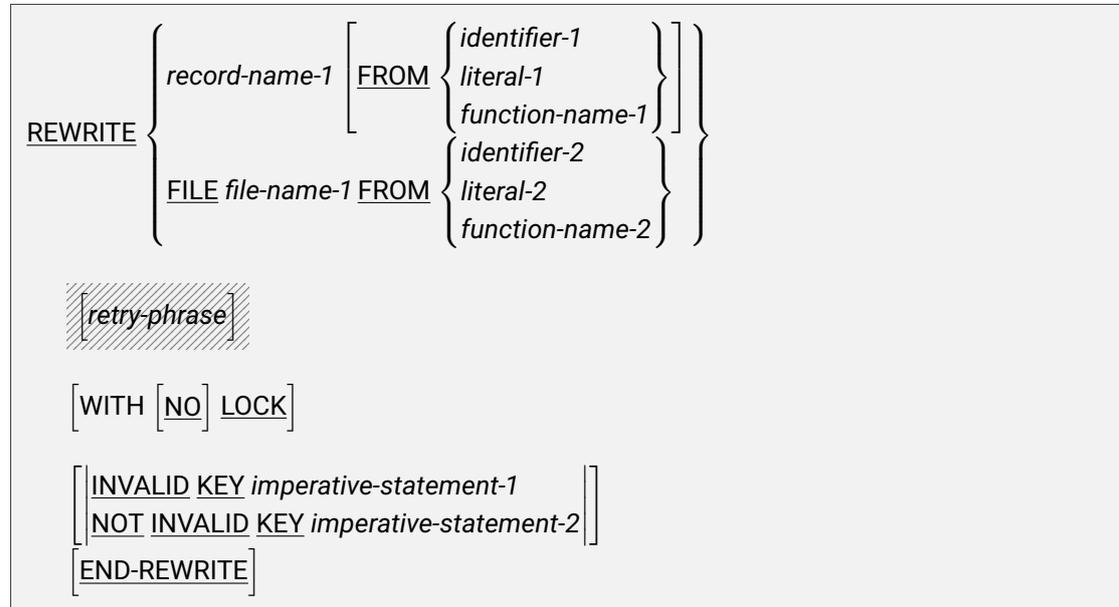
```
[END-RETURN]
```

Syntax rules

General rules

8.49. REWRITE statement

The REWRITE statement replaces an existing record in the file with one provided by the program.



Syntax rules

General rules

8.50. ROLLBACK statement

The ROLLBACK statement deletes any pending file writes.

ROLLBACK

Syntax rules

General rules

8.51. SEARCH statement

The SEARCH statement iterates through a table to find a record satisfying a condition.

Format 1 (simple)

```
SEARCH identifier-1 [VARYING identifier-2]  
  
  [AT END imperative-statement-1]  
  
  {WHEN condition-1 imperative-statement-2} ...  
  
  [END-SEARCH]
```

Format 2 (all)

```
SEARCH ALL identifier-3  
  
  [AT END imperative-statement-3]  
  
  WHEN expression-1 imperative-statement-4  
  
  [END-SEARCH]
```

Syntax rules

General rules

8.52. SEND statement

The SEND statement provides data to a communication descriptor.

Format 1 (from)

```
SEND cd-name-1 FROM identifier-1
```

Format 2 (with indicator)

```
SEND cd-name-2 [FROM identifier-2] WITH { identifier-3
      ESI
      EMI
      EGI
      }
      [ { BEFORE
        AFTER
        } ADVANCING { { identifier-4 } [ LINE
                       { literal-1 } [ LINES
                       mnemonic-name-1
                       ]
                     }
      ]
      [ REPLACING LINE ]
```

Syntax rules

General rules

8.53. SET statement

The SET statement sets the value or properties of a data item.

Format 1 (simple)

$$\text{SET } \underline{\text{identifier-1}} \text{ TO } \left\{ \begin{array}{l} \underline{\text{identifier-2}} \\ \underline{\text{literal-1}} \\ \underline{\text{arithmetic-expression-1}} \end{array} \right\}$$

Format 2 (entry)

$$\text{SET } \underline{\text{identifier-3}} \text{ TO ENTRY } \left\{ \begin{array}{l} \underline{\text{identifier-4}} \\ \underline{\text{literal-2}} \end{array} \right\}$$

Format 3 (environment)

$$\text{SET ENVIRONMENT } \left\{ \begin{array}{l} \underline{\text{identifier-5}} \\ \underline{\text{literal-3}} \end{array} \right\} \text{ TO } \left\{ \begin{array}{l} \underline{\text{identifier-6}} \\ \underline{\text{literal-4}} \end{array} \right\}$$

Format 4 (attribute)

$$\text{SET } \underline{\text{identifier-7}} \text{ ATTRIBUTE } \left\{ \begin{array}{l} \left\{ \begin{array}{l} \underline{\text{BELL}} \\ \underline{\text{BEEP}} \end{array} \right\} \\ \underline{\text{BLINK}} \\ \underline{\text{HIGHLIGHT}} \\ \underline{\text{LOWLIGHT}} \\ \underline{\text{REVERSE-VIDEO}} \\ \underline{\text{UNDERLINE}} \\ \underline{\text{LEFTLINE}} \\ \underline{\text{OVERLINE}} \end{array} \right\} \left\{ \begin{array}{l} \underline{\text{ON}} \\ \underline{\text{OFF}} \end{array} \right\} \dots$$

8. Procedure division

Format 5 (arithmetic)

SET { *index-name-1* } ... { UP
DOWN } BY *arithmetic-expression-2*

Format 6 (on/off)

SET { { *mnemonic-name-1* } ... TO { ON
OFF } } ...

Format 7 (true/false)

SET { { *condition-name-1* } ... TO { TRUE
FALSE } } ...

Format 8 (exception)

SET LAST EXCEPTION TO OFF

Format 9 (thread)

SET THREAD [*identifier-8*] PRIORITY TO { *identifier-9*
integer-1 }

Format 10 (file-handler)

SET { *identifier-10* } ... TO ADDRESS OF { FH-FCD
FH-KEYDEF } OF *file-name-1*

8. Procedure division

Syntax rules

General rules

8.54. SORT statement

The SORT statement sorts the record of a file or a table.

```

SORT identifier-1 [ ON { ASCENDING
                    DESCENDING } KEY [identifier-2]... ] ...

[ WITH DUPLICATES [ IN ORDER ] ]

[ COLLATING SEQUENCE IS { IS alphanumeric-collating-sequence [ national-collating-sequence ]
                          { FOR ALPHANUMERIC IS alphanumeric-collating-sequence
                            FOR NATIONAL IS national-collating-sequence } } ]

[ USING { file-name-1 }...
  INPUT PROCEDURE IS procedure-name-1 [ { THROUGH
                                          THRU } procedure-name-2 ] ]

[ GIVING { file-name-2 }...
  OUTPUT PROCEDURE IS procedure-name-3 [ { THROUGH
                                           THRU } procedure-name-4 ] ]

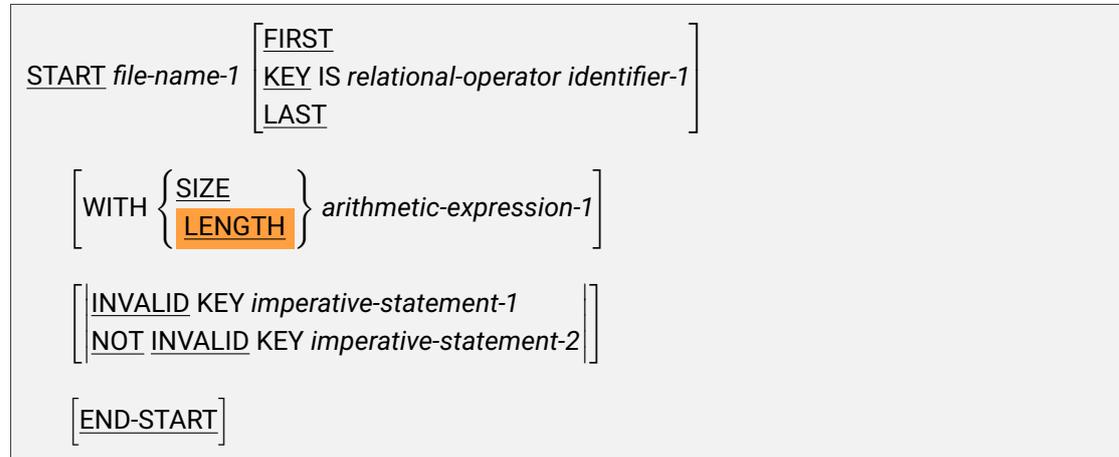
```

Syntax rules

General rules

8.55. START statement

The START statement changes the record currently being considered. It may also change the order in which records are accessed.



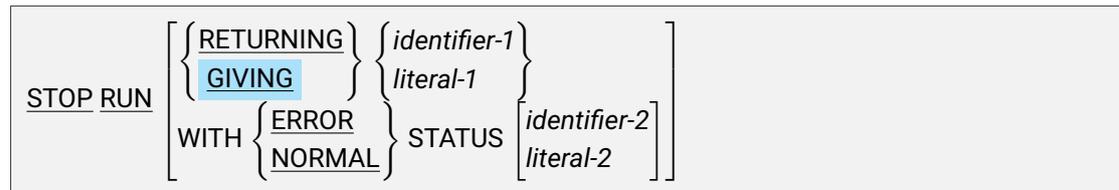
Syntax rules

General rules

8.56. STOP statement

The STOP statement terminates the run unit and returns control to the operating system.

Format 1 (standard)



Format 2 (literal)

STOP literal-3

Format 3 (identifier)

STOP identifier-3

Format 4 (ACUCOBOL)

STOP RUN { identifier-4 }
 { literal-4 }

Format 5 (thread)

STOP THREAD [identifier-5]

8. *Procedure division*

Syntax rules

General rules

8.57. STRING statement

The STRING statement appends multiples character strings and stores the result.

$$\text{STRING} \left\{ \left\{ \begin{array}{l} \text{identifier-1} \\ \text{literal-1} \end{array} \right\} \left[\text{DELIMITED BY} \left\{ \begin{array}{l} \text{SIZE} \\ \text{identifier-2} \\ \text{literal-2} \end{array} \right\} \right] \right\} \dots \text{INTO identifier-3}$$

[WITH POINTER IS identifier-4]

[[ON OVERFLOW imperative-statement-1
 [NOT ON OVERFLOW imperative-statement-2]]]

[END-STRING]

Syntax rules

General rules

8.58. SUBTRACT statement

The SUBTRACT statement subtracts one set of numbers from another set of numbers and stores the results.

Format 1 (simple)

```

SUBTRACT { identifier-1 } ... FROM { { identifier-2 } [rounded-phrase] } ...
        { literal-1 }
        { literal-2 }

[ ON SIZE ERROR imperative-statement-1 ]
[ NOT ON SIZE ERROR imperative-statement-2 ]

[ END-SUBTRACT ]

```

Format 2 (giving)

```

SUBTRACT { identifier-3 } ... FROM { identifier-4 }
        { literal-3 }         { literal-4 }

GIVING { { identifier-5 } [rounded-phrase] } ...
        { literal-5 }

[ ON SIZE ERROR imperative-statement-3 ]
[ NOT ON SIZE ERROR imperative-statement-4 ]

[ END-SUBTRACT ]

```

Format 3 (corresponding)

8. Procedure division

```
SUBTRACT { CORR  
          CORRESPONDING } identifier-6 FROM identifier-7 [rounded-phrase]  
  
  [ ON SIZE ERROR imperative-statement-5  
    NOT ON SIZE ERROR imperative-statement-6 ]  
  
  [END-SUBTRACT]
```

Format 4 (table)

```
SUBTRACT TABLE identifier-8 TO identifier-9 [rounded-phrase]  
  
  [ FROM INDEX integer-1 TO integer-2 ]  
  
  [ DESTINATION INDEX integer-3 ]  
  
  [ ON SIZE ERROR imperative-statement-7  
    NOT ON SIZE ERROR imperative-statement-8 ]  
  
  [END-SUBTRACT]
```

Syntax rules

General rules

8.59. SUPPRESS statement

The SUPPRESS statement suppresses the writing of a report group.

```
SUPPRESS PRINTING
```

Syntax rules

General rules

8.60. TERMINATE statement

The TERMINATE statement prevents further writing of a report.

```
TERMINATE {report-name-1} ...
```

Syntax rules

General rules

8.61. TRANSFORM statement

The TRANSFORM statement replaces instances of character with another character.

```
TRANSFORM identifier-1 FROM { identifier-2 } TO { identifier-3 }  
                                  { literal-1 }
```

Syntax rules

General rules

8.62. UNLOCK statement

The UNLOCK statement releases all currently held locks on a file.

```
UNLOCK file-name-1 [RECORD  
RECORDS]
```

Syntax rules

General rules

8.63. UNSTRING statement

The UNSTRING statement extracts substrings from a string and copies the substrings to specified data items.

```

UNSTRING { identifier-1 }
         { literal-1 }

[ DELIMITED BY [ ALL ] { identifier-2 } { literal-2 } { OR [ ALL ] { identifier-3 } { literal-3 } } ... ]

INTO { identifier-4 [ DELIMITER IN identifier-5 ] [ COUNT IN identifier-6 ] } ...

[ WITH POINTER IS identifier-7 ]

[ TALLYING IN identifier-8 ]

[ [ ON OVERFLOW imperative-statement-1
  NOT ON OVERFLOW imperative-statement-2 ] ]

[ END-UNSTRING ]

```

Syntax rules

General rules

8.64. USE statement

The USE statement indicates when a declarative should be executed.

Format 1 (file exception)

$$\text{USE } [\text{GLOBAL}] \text{ AFTER STANDARD } \left\{ \begin{array}{l} \text{EXCEPTION} \\ \text{ERROR} \end{array} \right\} \text{ PROCEDURE ON}$$

$$\left\{ \begin{array}{l} \{ \text{file-name-1} \} \dots \left[\begin{array}{l} \text{INPUT} \\ \text{OUTPUT} \\ \text{I-O} \\ \text{EXTEND} \end{array} \right] \dots \end{array} \right\}$$

Format 2 (debugging)

$$\text{USE FOR DEBUGGING ON } \left\{ \begin{array}{l} \text{procedure-name-1} \\ \text{ALL PROCEDURES} \\ \text{ALL REFERENCES OF identifier-1} \end{array} \right\} \dots$$

Format 3 (start/end)

$$\text{USE AT PROGRAM } \left\{ \begin{array}{l} \text{START} \\ \text{END} \end{array} \right\}$$

Format 4 (reporting)

$$\text{USE } [\text{GLOBAL}] \text{ BEFORE REPORTING } \text{identifier-2}$$

8. Procedure division

Format 5 (exception)

USE { EXCEPTION exception-name [FILE file-name-2] ... } ...

Syntax rules

General rules

8.65. VALIDATE statement

The VALIDATE statement performs various validation tasks for a data item.

```
VALIDATE {identifier-1}...
```

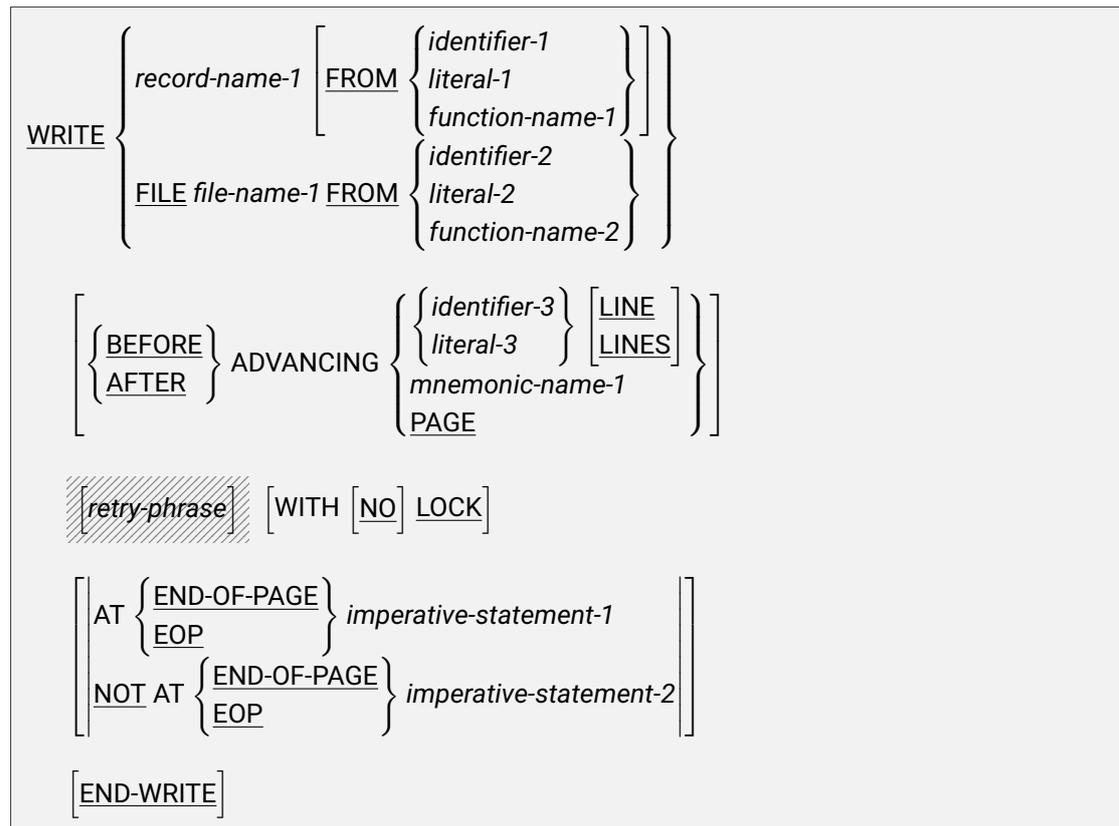
Syntax rules

General rules

8.66. WRITE statement

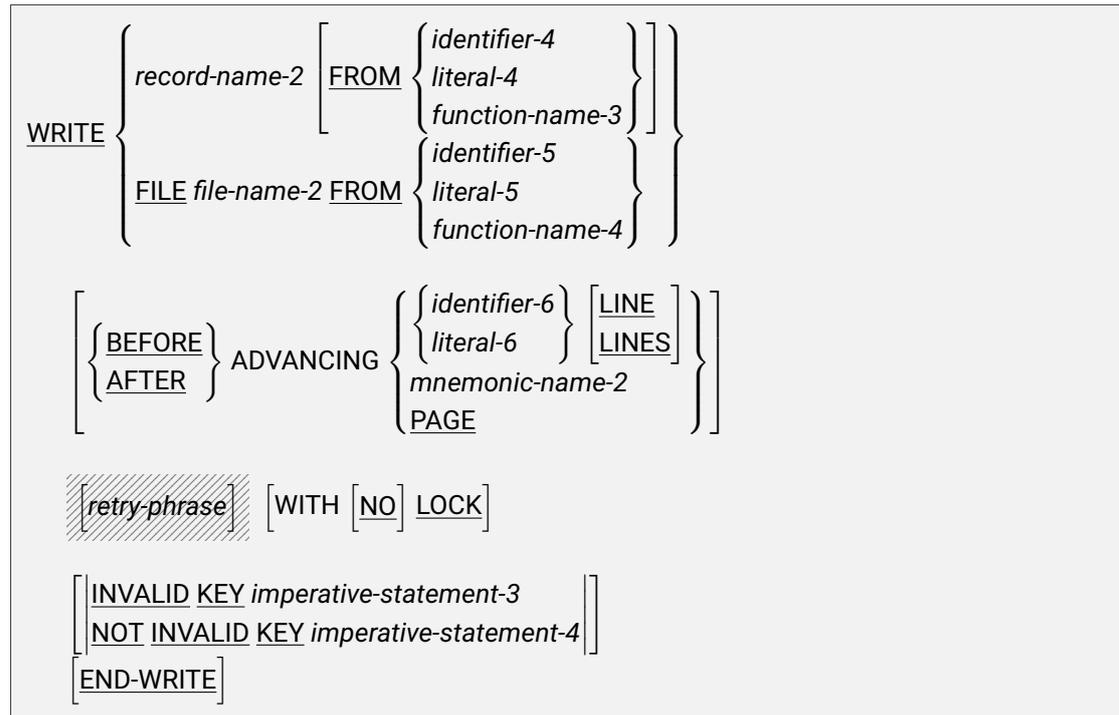
The WRITE statement provides a new record to the file.

Format 1 (sequential)



Format 2 (random)

8. Procedure division



Syntax rules

General rules

8.67. XML GENERATE statement

The XML GENERATE statement creates an XML document based on the structure and content of the input data item.

```

XML GENERATE identifier-1 FROM identifier-2

  [COUNT IN identifier-3]

  [WITH ENCODING { identifier-4
                  literal-1 } ]

  [WITH XML-DECLARATION]

  [WITH ATTRIBUTES]

  [NAMESPACE IS { identifier-5
                  literal-2 } [NAMESPACE-PREFIX IS { identifier-6
                                                       literal-3 } ] ]

  [NAME OF { identifier-7 IS literal-4 }...]

  [TYPE OF { identifier-8 IS { ATTRIBUTE
                              ELEMENT
                              CONTENT } }...]

  [SUPPRESS { identifier-9
              { { NUMERIC
                  NONNUMERIC } { ATTRIBUTE
                              ELEMENT
                              CONTENT } }
              { { ATTRIBUTE
                  ELEMENT
                  CONTENT } }
              WHEN [ non-quote-alnum-fig-const OR ]... non-quote-alnum-fig-const } ]

  [ON EXCEPTION imperative-statement-1
   NOT ON EXCEPTION imperative-statement-2 ]

  [END-XML]

```

where *non-quote-alnum-fig-const* is

8. Procedure division

<u>ZERO</u>	}
<u>ZEROES</u>	
<u>ZEROS</u>	
<u>SPACE</u>	
<u>SPACES</u>	
<u>HIGH-VALUE</u>	
<u>HIGH-VALUES</u>	
<u>LOW-VALUE</u>	
<u>LOW-VALUES</u>	

Syntax rules

General rules

9. Intrinsic functions

9.1. ABS function

FUNCTION { ABS
ABSOLUTE-VALUE } (*argument-1*)

Syntax rules

General rules

9.2. ACOS function

```
FUNCTION ACOS ( argument-1 )
```

Syntax rules

General rules

9.3. ANNUITY function

FUNCTION ANNUITY (*argument-1* *argument-2*)

Syntax rules

General rules

9.4. ASIN function

```
FUNCTION ASIN ( argument-1 )
```

Syntax rules

General rules

9.5. ATAN function

```
FUNCTION ATAN ( argument-1 )
```

Syntax rules

General rules

9.6. **BOOLEAN-OF-INTEGER** function

`FUNCTION BOOLEAN-OF-INTEGER (argument-1 argument-2)`

Syntax rules

General rules

9.7. BYTE-LENGTH function

FUNCTION BYTE-LENGTH (*argument-1*)

Syntax rules

General rules

9.8. CHAR function

FUNCTION CHAR (*argument-1*)

Syntax rules

General rules

9.9. CHAR-NATIONAL function

`FUNCTION CHAR-NATIONAL (argument-1)`

Syntax rules

General rules

9.10. COMBINED-DATETIME function

FUNCTION COMBINED-DATETIME (*argument-1* *argument-2*)

Syntax rules

General rules

9.11. CONCATENATE function

FUNCTION CONCATENATE ({ *argument-1* } ...)

Syntax rules

General rules

9.12. CONTENT-LENGTH function

FUNCTION CONTENT-LENGTH (*argument-1*)

Syntax rules

General rules

9.13. CONTENT-OF function

FUNCTION CONTENT-OF (*argument-1* [*argument-2*])

Syntax rules

General rules

9.14. COS function

```
FUNCTION COS ( argument-1 )
```

Syntax rules

General rules

9.15. CURRENCY-SYMBOL function

FUNCTION CURRENCY-SYMBOL

Syntax rules

General rules

9.16. **CURRENT-DATE** function

FUNCTION <u>CURRENT-DATE</u>

Syntax rules

General rules

9.17. DATE-OF-INTEGER function

FUNCTION DATE-OF-INTEGER (*argument-1*)

Syntax rules

General rules

9.18. DATE-TO-YYYYMMDD function

```
FUNCTION DATE-TO-YYYYMMDD ( argument-1 [ argument-2 [ argument-3 ] ] )
```

Syntax rules

General rules

9.19. DAY-OF-INTEGER function

FUNCTION DAY-OF-INTEGER (*argument-1*)

Syntax rules

General rules

9.20. DAY-TO-YYYYDDD function

```
FUNCTION DAY-TO-YYYYDDD ( argument-1 [ argument-2 [ argument-3 ] ] )
```

Syntax rules

General rules

9.21. DISPLAY-OF function

FUNCTION DISPLAY-OF (argument-1)

Syntax rules

General rules

9.22. E function

FUNCTION E

Syntax rules

General rules

9.23. EXCEPTION-FILE function

FUNCTION EXCEPTION-FILE

Syntax rules

General rules

9.24. EXCEPTION-FILE-N function

~~FUNCTION EXCEPTION-FILE-N~~

Syntax rules

General rules

9.25. EXCEPTION-LOCATION function

FUNCTION EXCEPTION-LOCATION

Syntax rules

General rules

9.26. EXCEPTION-LOCATION-N function

~~FUNCTION EXCEPTION-LOCATION-N~~

Syntax rules

General rules

9.27. EXCEPTION-STATEMENT function

FUNCTION EXCEPTION-STATEMENT

Syntax rules

General rules

9.28. EXCEPTION-STATUS function

FUNCTION EXCEPTION-STATUS

Syntax rules

General rules

9.29. EXP function

```
FUNCTION EXP ( argument-1 )
```

Syntax rules

General rules

9.30. EXP10 function

FUNCTION EXP10 (*argument-1*)

Syntax rules

General rules

9.31. FACTORIAL function

FUNCTION FACTORIAL (*argument-1*)

Syntax rules

General rules

9.32. FORMATTED-CURRENT-DATE function

FUNCTION FORMATTED-CURRENT-DATE (*argument-1*)

Syntax rules

General rules

9.33. FORMATTED-DATE function

FUNCTION FORMATTED-DATE (*argument-1* *argument-2*)

Syntax rules

General rules

9.34. FORMATTED-DATETIME function

FUNCTION FORMATTED-DATETIME

(*argument-1* *argument-2* *argument-3* [*argument-4*
SYSTEM-OFFSET])

Syntax rules

General rules

9.35. FORMATTED-TIME function

FUNCTION FORMATTED-TIME (*argument-1* *argument-2* [*argument-3*
SYSTEM-OFFSET])

Syntax rules

General rules

9.36. FRACTION-PART function

FUNCTION FRACTION-PART (*argument-1*)

Syntax rules

General rules

9.37. HIGHEST-ALGEBRAIC function

FUNCTION HIGHEST-ALGEBRAIC (*argument-1*)

Syntax rules

General rules

9.38. INTEGER function

FUNCTION INTEGER (*argument-1*)

Syntax rules

General rules

9.39. INTEGER-OF-BOOLEAN function

FUNCTION INTEGER-OF-BOOLEAN (argument-1)

Syntax rules

General rules

9.40. INTEGER-OF-DATE function

FUNCTION INTEGER-OF-DATE (*argument-1*)

Syntax rules

General rules

9.41. INTEGER-OF-DAY function

FUNCTION INTEGER-OF-DAY (*argument-1*)

Syntax rules

General rules

9.42. INTEGER-OF-FORMATTED-DATE function

FUNCTION INTEGER-OF-FORMATTED-DATE (*argument-1* *argument-2*)

Syntax rules

General rules

9.43. INTEGER-PART function

FUNCTION INTEGER-PART (*argument-1*)

Syntax rules

General rules

9.44. LENGTH function

FUNCTION LENGTH (*argument-1* [PHYSICAL])

Syntax rules

General rules

9.45. LENGTH-AN function

`FUNCTION LENGTH-AN (argument-1)`

Syntax rules

General rules

9.46. LOCALE-COMPARE function

```
FUNCTION LOCALE-COMPARE ( argument-1 argument-2 [ argument-3 ] )
```

Syntax rules

General rules

9.47. LOCALE-DATE function

```
FUNCTION LOCALE-DATE ( argument-1 [ argument-2 ] )
```

Syntax rules

General rules

9.48. LOCALE-TIME function

```
FUNCTION LOCALE-TIME ( argument-1 [ argument-2 ] )
```

Syntax rules

General rules

9.49. LOCALE-TIME-FROM-SECONDS function

```
FUNCTION LOCALE-TIME-FROM-SECONDS ( argument-1 [ argument-2 ] )
```

Syntax rules

General rules

9.50. LOG function

FUNCTION LOG (*argument-1*)

Syntax rules

General rules

9.51. LOG10 function

```
FUNCTION LOG10 ( argument-1 )
```

Syntax rules

General rules

9.52. LOWER-CASE function

FUNCTION LOWER-CASE (*argument-1*)

Syntax rules

General rules

9.53. LOWEST-ALGEBRAIC function

FUNCTION LOWEST-ALGEBRAIC (*argument-1*)

Syntax rules

General rules

9.54. MAX function

```
FUNCTION MAX ( {argument-1}... )
```

Syntax rules

General rules

9.55. MEAN function

```
FUNCTION MEAN ( {argument-1}... )
```

Syntax rules

General rules

9.56. MEDIAN function

```
FUNCTION MEDIAN ( { argument-1 } ... )
```

Syntax rules

General rules

9.57. MIDRANGE function

```
FUNCTION MIDRANGE ( { argument-1 } ... )
```

Syntax rules

General rules

9.58. MIN function

```
FUNCTION MIN ( {argument-1}... )
```

Syntax rules

General rules

9.59. MOD function

`FUNCTION MOD (argument-1 argument-2)`

Syntax rules

General rules

9.60. MODULE-CALLER-ID function

`FUNCTION MODULE-CALLER-ID`

Syntax rules

General rules

9.61. **MODULE-DATE** function

FUNCTION MODULE-DATE

Syntax rules

General rules

9.62. **MODULE-FORMATTED-DATE** function

FUNCTION MODULE-FORMATTED-DATE

Syntax rules

General rules

9.63. MODULE-ID function

FUNCTION MODULE-ID

Syntax rules

General rules

9.64. **MODULE-PATH** function

FUNCTION MODULE-PATH

Syntax rules

General rules

9.65. **MODULE-SOURCE** function

`FUNCTION MODULE-SOURCE`

Syntax rules

General rules

9.66. **MODULE-TIME** function

FUNCTION MODULE-TIME

Syntax rules

General rules

9.67. **MONETARY-DECIMAL-POINT** function

FUNCTION MONETARY-DECIMAL-POINT

Syntax rules

General rules

9.68. **MONETARY-THOUSANDS-SEPARATOR** function

`FUNCTION MONETARY-THOUSANDS-SEPARATOR`

Syntax rules

General rules

9.69. NATIONAL-OF function

FUNCTION NATIONAL-OF (*argument-1* [*argument-2*])

Syntax rules

General rules

9.70. **NUMERIC-DECIMAL-POINT function**

FUNCTION NUMERIC-DECIMAL-POINT

Syntax rules

General rules

9.71. NUMERIC-THOUSANDS-SEPARATOR function

FUNCTION NUMERIC-THOUSANDS-SEPARATOR

Syntax rules

General rules

9.72. NUMVAL function

FUNCTION NUMVAL (*argument-1*)

Syntax rules

General rules

9.73. NUMVAL-C function

```
FUNCTION NUMVAL-C ( argument-1 [ argument-2 ] )
```

Syntax rules

General rules

9.74. NUMVAL-F function

```
FUNCTION NUMVAL-F ( argument-1 )
```

Syntax rules

General rules

9.75. ORD function

`FUNCTION ORD (argument-1)`

Syntax rules

General rules

9.76. ORD-MAX function

```
FUNCTION ORD-MAX ( { argument-1 } ... )
```

Syntax rules

General rules

9.77. ORD-MIN function

```
FUNCTION ORD-MIN ( {argument-1} ... )
```

Syntax rules

General rules

9.78. PI function

FUNCTION PI

Syntax rules

General rules

9.79. PRESENT-VALUE function

`FUNCTION PRESENT-VALUE (argument-1 argument-2 [argument-3] ...)`

Syntax rules

General rules

9.80. RANDOM function

```
FUNCTION RANDOM ( [argument-1] )
```

Syntax rules

General rules

9.81. RANGE function

`FUNCTION RANGE ({ argument-1 } ...)`

Syntax rules

General rules

9.82. REM function

```
FUNCTION REM ( argument-1 argument-2 )
```

Syntax rules

General rules

9.83. REVERSE function

FUNCTION REVERSE (*argument-1*)

Syntax rules

General rules

9.84. SECONDS-FROM-FORMATTED-TIME function

FUNCTION SECONDS-FROM-FORMATTED-TIME (*argument-1* *argument-2*)

Syntax rules

General rules

9.85. SECONDS-PAST-MIDNIGHT function

FUNCTION SECONDS-PAST-MIDNIGHT (*argument-1*)

Syntax rules

General rules

9.86. SIGN function

FUNCTION SIGN (*argument-1*)

Syntax rules

General rules

9.87. SIN function

```
FUNCTION SIN ( argument-1 )
```

Syntax rules

General rules

9.88. SQRT function

```
FUNCTION SQRT ( argument-1 )
```

Syntax rules

General rules

9.89. STANDARD-COMPARE function

~~FUNCTION STANDARD-COMPARE~~

(~~argument-1 argument-2 [argument-3] [argument-4]~~)

Syntax rules

General rules

9.90. STANDARD-DEVIATION function

FUNCTION STANDARD-DEVIATION ({ *argument-1* } ...)

Syntax rules

General rules

9.91. STORED-CHAR-LENGTH function

FUNCTION STORED-CHAR-LENGTH (*argument-1*)

Syntax rules

General rules

9.92. SUBSTITUTE function

```
FUNCTION SUBSTITUTE ( argument-1 { argument-2 argument-3 } ... )
```

Syntax rules

General rules

9.93. SUBSTITUTE-CASE function

FUNCTION SUBSTITUTE-CASE (*argument-1* { *argument-2* *argument-3* }...)

Syntax rules

General rules

9.94. SUM function

```
FUNCTION SUM ( { argument-1 } ... )
```

Syntax rules

General rules

9.95. TAN function

FUNCTION TAN (*argument-1*)

Syntax rules

General rules

9.96. TEST-DATE-YYYYMMDD function

```
FUNCTION TEST-DATE-YYYYMMDD ( argument-1 )
```

Syntax rules

General rules

9.97. TEST-DAY-YYYYDDD function

```
FUNCTION TEST-DAY-YYYYDDD ( argument-1 )
```

Syntax rules

General rules

9.98. TEST-FORMATTED-DATETIME function

```
FUNCTION TEST-FORMATTED-DATETIME ( argument-1 argument-2 )
```

Syntax rules

General rules

9.99. TEST-NUMVAL function

```
FUNCTION TEST-NUMVAL ( argument-1 )
```

Syntax rules

General rules

9.100. TEST-NUMVAL-C function

FUNCTION TEST-NUMVAL-C (*argument-1* *argument-2*)

Syntax rules

General rules

9.101. TEST-NUMVAL-F function

FUNCTION TEST-NUMVAL-F (*argument-1*)

Syntax rules

General rules

9.102. TRIM function

```
FUNCTION TRIM ( argument-1 [ LEADING  
TRAILING ] )
```

Syntax rules

General rules

9.103. UPPER-CASE function

FUNCTION UPPER-CASE (*argument-1*)

Syntax rules

General rules

9.104. VARIANCE function

FUNCTION VARIANCE ({ *argument-1* } ...)

Syntax rules

General rules

9.105. **WHEN-COMPILED** function

FUNCTION WHEN-COMPILED

Syntax rules

General rules

9.106. YEAR-TO-YYYY function

```
FUNCTION YEAR-TO-YYYY ( argument-1 [ argument-2 [ argument-3 ] ] )
```

Syntax rules

General rules

9. *Intrinsic functions*

Appendices

A. GNU Free Documentation License

Version 1.3, 3 November 2008

Copyright © 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.

`<http://fsf.org/>`

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

A.1. Preamble

The purpose of this License is to make a manual, textbook, or other functional and useful document “free” in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of “copyleft”, which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

A.2. Applicability and definitions

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The “**Document**”, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as “**you**”. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A “**Modified Version**” of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A “**Secondary Section**” is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document’s overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The “**Invariant Sections**” are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The “**Cover Texts**” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A “**Transparent**” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not “Transparent” is called “**Opaque**”.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD,

A. GNU Free Documentation License

and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The “**Title Page**” means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, “Title Page” means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

The “**publisher**” means any person or entity that distributes copies of the Document to the public.

A section “**Entitled XYZ**” means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as “**Acknowledgements**”, “**Dedications**”, “**Endorsements**”, or “**History**”.) To “**Preserve the Title**” of such a section when you modify the Document means that it remains a section “Entitled XYZ” according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

A.3. Verbatim copying

You may copy and distribute the Document in any medium, either commercially or non-commercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

A.4. Copying in quantity

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

A.5. Modifications

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the

A. GNU Free Documentation License

“History” section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

- K. For any section Entitled “Acknowledgements” or “Dedications”, Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled “Endorsements”. Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled “Endorsements” or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version’s license notice. These titles must be distinct from any other section titles.

You may add a section Entitled “Endorsements”, provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

A.6. Combining documents

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled “History” in the various original documents, forming one section Entitled “History”; likewise combine any sections Entitled “Acknowledgements”, and any sections Entitled “Dedications”. You must delete all sections Entitled “Endorsements”.

A.7. Collections of documents

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

A.8. Aggregation with independent works

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an “aggregate” if the copyright resulting from the compilation is not used to limit the legal rights of the compilation’s users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document’s Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

A.9. Translation

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled “Acknowledgements”, “Dedications”, or “History”, the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

A.10. Termination

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

A.11. Future revisions of this license

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License “or any later version” applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy’s public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

A.12. Relicensing

“Massive Multiauthor Collaboration Site” (or “MMC Site”) means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A “Massive Multiauthor Collaboration” (or “MMC”) contained in the site means any set of copyrightable works thus published on the MMC site.

“CC-BY-SA” means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

“Incorporate” means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is “eligible for relicensing” if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

A.13. Addendum: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright © YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled “GNU Free Documentation License”.

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the “with ... Texts.” line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

B. Language element lists

This appendix is required for conformance to the COBOL standard.

B.1. Implementor-defined language element list

1. **ACCEPT statement (conversion of data):**
2. **ACCEPT statement (device used when FROM is unspecified):**
3. **ACCEPT statement, screen format (result when screen items overlap):**
4. **ACCEPT statement, screen format (when data is verified, behavior for inconsistent data):**
5. **ACCEPT statement (size of data transfer):**
6. **Alignment of alphanumeric group items (relative to first elementary item):**
7. **Alignment of data for increased efficiency (special or automatic alignment: interpretation, implicit filler, semantics of statements):**
8. **ALPHABET clause (ordinal number of characters in the native coded character sets):**
9. **Alphanumeric literals (number of hexadecimal digits that map to an alphanumeric character):**
10. **ASSIGN clause, USING phrase (meaning and rules for operands; consistency rules):**
11. **BACKGROUND-COLOR clause (the background color when the clause is not specified or the value specified is not in the range 0 to 7):**
12. **Byte (number of bits in):**
13. **CALL statement (rules for program-name formation for a non-COBOL program):**
14. **CALL statement (runtime resources that are checked):**
15. **CALL statement (rules for locating a non-COBOL program):**
16. **CALL statement (calling a non-COBOL program):**

B. Language element lists

17. **CALL statement (other effects of the CALL statement):**
18. **CANCEL statement (result of canceling an active program when EC-PROGRAM-CANCEL-ACTIVE is not enabled):**
19. **CANCEL statement (result of canceling a non-COBOL program):**
20. **Case mapping:**
21. **CHAR function (which one of the multiple characters is returned):**
22. **CHAR-NATIONAL function (which one of the multiple characters is returned):**
23. **Characters prohibited from use in text-words in COPY ... REPLACING and REPLACE statements:**
24. **CLOSE statement (closing operations):**
25. **COBOL character repertoire (encoding of, mapping of, substitute graphics):**
26. **COBOL character repertoire (if more than one encoding in a compilation group, control functions if any):**
27. **Color number (for a monochrome terminal, the mapping of the color attributes onto other attributes):**
28. **Compiler directives, compiler directive IMP (syntax rules and general rules):**
29. **Computer's coded character set (characters in and encoding of computer's alphanumeric coded character set and computer's national coded character set, encoding for usage DISPLAY and usage NATIONAL):**
30. **Computer's coded character set (coded character values for certain COBOL items):**
31. **Computer's coded character set (correspondence between alphanumeric and national characters):**
32. **Computer's coded character set (for literals, correspondence between compile-time and runtime character sets, when conversion takes place):**
33. **Computer's coded character set (correspondence between lowercase and uppercase letters when a locale is not in effect):**
34. **Computer's coded character set (when composite alphanumeric and national, mapping of characters to each):**
35. **Computer's coded character set (when more than one encoding, the mechanism for selecting encoding for runtime):**

B. Language element lists

36. **Computer's coded character set (whether UTF-8 or mixed alphanumeric and national characters recognized in class alphanumeric; applicable syntax and general rules):**
37. **COPY statement (rules for identifying and locating default library text):**
38. **CRT status 9xxx (the value of xxx for unsuccessful completion with implementor-defined conditions):**
39. **Cultural ordering table (allowable content of literal defines a cultural ordering table):**
40. **Currency symbol (equivalence of non-COBOL characters):**
41. **Currency symbol (implementor-defined prohibition of non-COBOL characters):**
42. **Cursor (the cursor movement if keys are defined that change the cursor position):**
43. **Data storage (possible representations when implementation provides multiple ways of storing data):**
44. **Default encoding specifications (for standard decimal floating-point usages):**
45. **Default endianness specifications (for standard floating-point usages):**
46. **DEFINE directive (mechanism for providing value of a compilation-variable-name from the operating environment):**
47. **Devices that allow concurrent access:**
48. **DISPLAY statement (data conversion):**
49. **DISPLAY statement (format for display of a variable-length group):**
50. **DISPLAY statement (size of data transfer):**
51. **DISPLAY statement (standard display device):**
52. **Dynamic-capacity table (determination of highest permissible occurrence number):**
53. **Dynamic-capacity table (physical allocation):**
54. **Dynamic-length elementary items (maximum length):**
55. **Dynamic-length elementary items (structure if dynamic-length-structure-name-1 is not specified):**
56. **ENTRY-CONVENTION clause (entry-convention-names, their meanings and the default when not specified):**

B. Language element lists

57. **EXIT and GOBACK statements (execution continuation in a non-COBOL runtime element):**
58. **Exponentiation (results for certain operand values):**
59. **External repository (mechanism for specifying whether checking and updating occur):**
60. **External repository information (other information beyond the required information):**
61. **Externalized names (formation and mapping rules):**
62. **Fatal exception condition (whether detected at compile time, circumstances under which detected):**
63. **Fatal exception condition (whether or not execution will continue, how it will continue, and how any receiving operands are affected when events that would cause a fatal exception to exist occur but checking for that condition is not enabled):**
64. **FILE-CONTROL entry, ASSIGN clause (TO phrase meaning and rules):**
65. **FILE-CONTROL entry, ASSIGN clause (consistency rules for external file connectors):**
66. **FILE-CONTROL entry, ASSIGN clause (USING phrase meaning and rules):**
67. **Figurative constant values (representation of zero, space, and quote):**
68. **File sharing (interaction with other facilities and languages):**
69. **File sharing (which devices allow concurrent access to the file):**
70. **File sharing (default mode when unspecified):**
71. **Fixed file attribute (whether the ability to share a file is a fixed file attribute):**
72. **FLAG-02 directive (warning mechanism):**
73. **FLAG-85 directive (warning mechanism):**
74. **Floating-point numeric item (alignment when used as a receiving operand):**
75. **Floating-point numeric literals (maximum permitted value and minimum permitted value of the exponent):**
76. **BACKGROUND-COLOR clause (the foreground color when the clause is not specified or the value specified is not in the range 0 to 7):**
77. **FORMAT clause (representation produced):**

B. Language element lists

78. **FORMAT clause (exclusions on restoring to same internal representation):**
79. **Format validation (rules for checking items of usages other than display or national):**
80. **FORMATTED-CURRENT-DATE (accuracy of returned time):**
81. **Function-identifier (execution of a non-COBOL function when a function-prototype-name is specified):**
82. **Function-identifier (object time resources that are checked):**
83. **Function-identifier (result when argument rules are violated and checking for the EC-ARGUMENT-FUNCTION exception condition is not enabled):**
84. **Function keys (context-dependent keys, function number, and method for enabling and disabling):**
85. **Function returned values (characteristics, representation, and returned value for native arithmetic):**
86. **Hexadecimal alphanumeric literals (mapping for non-existing corresponding character):**
87. **Hexadecimal alphanumeric literals (mapping when characters not multiples of four bits):**
88. **Hexadecimal national literals (mapping for non-existing corresponding character):**
89. **Hexadecimal national literals (mapping when characters not multiples of four bits):**
90. **Implementor-defined exception conditions, EC-IMP-xxx (specification and meaning of xxx):**
91. **Implementor-defined level-2 exception conditions, EC-level-2-IMP (specification and meaning of the specified level-2 exception condition):**
92. **INVOKE statement (behavior when invoking a non-COBOL method):**
93. **INVOKE statement (runtime resources that are checked):**
94. **I-O status (action taken for fatal exception conditions):**
95. **I-O status (if more than one value applies):**
96. **I-O status, permanent error (technique for error correction):**
97. **I-O status 0x (value of x):**

B. Language element lists

98. **I-O status 24** (manner in which the boundaries of a file are defined):
99. **I-O status 34** (manner in which the boundaries of a file are defined):
100. **I-O status 52** (conditions under which deadlock is detected):
101. **I-O status 9x** (value of x):
102. **LEAP-SECOND directive** (whether a value greater than 59 seconds may be reported and, if so, the maximum number of seconds that may be reported):
103. **LEAP-SECOND directive** (whether standard numeric time form values greater than or equal to 86,400 may be reported):
104. **Life cycle for objects** (timing and algorithm for taking part in continued execution):
105. **Linkage section** (whether access to linkage section items is meaningful when called from a non-COBOL program):
106. **Listings** (whether and when produced by the compiler, effect of logical conversion):
107. **Locale specification** (how user and system defaults defined; at least one user and one system default):
108. **Locale specification** (manner of implementation):
109. **Locale switch** (whether a switch by a non-COBOL runtime module is recognized by COBOL):
110. **METHOD-ID paragraph** (actual method-name used when PROPERTY phrase is specified):
111. **National literals** (number of hexadecimal digits that map to a national character):
112. **Native arithmetic** (techniques used, intermediate data item):
113. **Native arithmetic** (when an operand or arithmetic expression is an integer):
114. **NULL** (value of NULL):
115. **OBJECT-COMPUTER paragraph** (default object computer):
116. **OBJECT-COMPUTER paragraph** (computer-name and implied equipment configuration):
117. **OCCURS clause** (range of values allowed in the index):
118. **OPEN statement** (validation of fixed file attributes):

B. Language element lists

119. **OPEN statement with OUTPUT phrase (positioning of the output file with regard to physical page boundaries):**
120. **OPEN statement without the SHARING phrase and no SHARING clause in the file control entry (definition of sharing mode established for each file connector):**
121. **Parameterized classes and interfaces (when expanded):**
122. **Procedure division header rules when either the activating or the activated runtime element is not a COBOL element (restrictions and mechanisms for all supported language products with details such as the matching of parameters, data type representation, returning of a value, and omission of parameters):**
123. **Program-address identifier (relation between address and non-COBOL program):**
124. **Program-name (formation rules for a non-COBOL program):**
125. **RANDOM function (seed value when no argument on first reference):**
126. **RANDOM function (subset of the domain of argument-1):**
127. **RECORD clause (calculations to derive size of records on storage medium):**
128. **RECORD clause (implicit RECORD clause if RECORD clause is not specified):**
129. **RECORD clause (whether fixed or variable records produced for fixed-or-variable-length format):**
130. **RECORD DELIMITER clause (consistency rules when used with external file connectors):**
131. **RECORD DELIMITER clause (feature-name and associated method for determining length of variable-length records):**
132. **RECORD DELIMITER clause (if not specified, method for determining length of variable-length records):**
133. **Record locking (circumstances other than a locked logical record that return a locked record status):**
134. **Record locking (default mode when unspecified by user):**
135. **Record locking (maximum number allowed for a run unit):**
136. **Record locks (maximum number allowed for a file connector):**
137. **Reference format (control characters in a free-form line):**
138. **Reference format (meaning of lines and character positions in free-form and fixed-form format):**

B. Language element lists

139. **Reference format (rightmost character position of program-text area):**
140. **Report file (record structure):**
141. **Report writer printable item (fixed correspondence between columns and national characters):**
142. **REPOSITORY paragraph (how external repository and class-specifier determine which class is used):**
143. **REPOSITORY paragraph, INTERFACE phrase (how interface specifier and external repository determine which interface is used):**
144. **REPOSITORY paragraph (when the AS phrase is required):**
145. **RESERVE clause (number of input-output areas, if not specified):**
146. **RETRY phrase (interval between attempts to obtain access to a locked file or record):**
147. **RETRY phrase (maximum meaningful time-out value and internal representation; technique for determining frequency of retries):**
148. **Run unit (relationship and interaction with non-COBOL components):**
149. **Run unit termination (whether locale reset):**
150. **SAME SORT/SORT-MERGE AREA clause (extent of allocation):**
151. **SEARCH ALL statement (varying of the search index during the search operation):**
152. **SECONDS-PAST-MIDNIGHT function returned value (precision):**
153. **SECURE clause (cursor movement when data is entered into a field for which the SECURE clause is specified):**
154. **SELECT WHEN clause (whether a SELECT WHEN takes effect for READ statements and REWRITE or WRITE statements with the FILE phrase in the absence of a CODE-SET clause or a FORMAT clause):**
155. **SET statement (effect of SET on function whose address is being stored in a function-pointer):**
156. **SET statement (effect of SET on program whose address is being stored in a program-pointer):**
157. **SET statement (value in NaN payload):**
158. **SIGN clause (representation when PICTURE contains character 'S' with no optional SIGN clause):**

B. Language element lists

159. **SIGN** clause (valid sign when **SEPARATE CHARACTER** phrase not present):
160. **Size error condition** (whether or not range of values allowed for the intermediate data item is to be checked):
161. **SPECIAL-NAMES** paragraph (allowable locale-names and literal values):
162. **SPECIAL-NAMES** paragraph, **ALPHABET** clause (coded character set referenced by **STANDARD-2** phrase):
163. **SPECIAL-NAMES** paragraph, **ALPHABET** clause, **code-name-1** (alphanumeric coded character set and collating sequence; ordinal number of characters; correspondence with native alphanumeric character set):
164. **SPECIAL-NAMES** paragraph, **ALPHABET** clause, **code-name-1** and **code-name-2** (the names supported for **code-name-1** and **code-name-2**):
165. **SPECIAL-NAMES** paragraph, **ALPHABET** clause, **code-name-2** (national coded character set and collating sequence; ordinal number of characters; correspondence with native national character set):
166. **SPECIAL-NAMES** paragraph, **ALPHABET** clause, literal phrase (ordinal number of characters not specified):
167. **SPECIAL-NAMES** paragraph, **ALPHABET** clause, **STANDARD-1** and **STANDARD-2** phrases (correspondence with native character set):
168. **SPECIAL-NAMES** paragraph, **ALPHABET** clause, **UCS-4**, **UTF-8**, and **UTF-16** phrases (correspondence with native character set):
169. **SPECIAL-NAMES** paragraph, **device-name** (names available, restrictions on use):
170. **SPECIAL-NAMES** paragraph, **feature-name** (names available, any positioning rules, any restrictions on use):
171. **SPECIAL-NAMES** paragraph, **switch-name** (names available, which switches may be referenced by the **SET** statement, scope of, and external facility for modification):
172. **Standard intermediate data item** (representation):
173. **STOP** statement (constraints on the value of the **STATUS** literal or on the contents of the data item referenced by the **STATUS** identifier):
174. **STOP** statement (mechanism for error termination):
175. **Subscripts** (mapping indexes to occurrence numbers):
176. **Switch-name** (identifies an external switch):

B. Language element lists

177. **SYNCHRONIZED** clause (effect on elementary items and containing records or groups; implicit filler generation):
178. **SYNCHRONIZED** clause (how records of a file are handled):
179. **SYNCHRONIZED** clause (positioning when neither **RIGHT** or **LEFT** is specified):
180. **System-names** (rules for formation of a system-name):
181. **Terminal screen** (correspondence of a column and a character in the computer's national coded character set):
182. **Text manipulation** (stage of processing the **LISTING** and **PAGE** directives and the **SUPPRESS** phrase of **COPY**):
183. **Text manipulation** (stage of processing parameterized class expansion):
184. **Time formats and corresponding function values** (maximum precision not less than nine fractional digits):
185. **THROUGH** phrase in **VALUE** clause and **EVALUATE** statement (collating sequence used for determining range of values when no alphabet-name is specified):
186. **TURN** directive (whether location information is available when the **LOCATION** phrase is not specified):
187. **USAGE BINARY** clause (computer storage allocation, alignment and representation of data):
188. **USAGE BINARY-CHAR, BINARY-SHORT, BINARY-LONG, BINARY-DOUBLE** (allow wider range than minimum specified):
189. **USAGE BINARY-SHORT, BINARY-LONG, BINARY-DOUBLE, FLOAT-SHORT, FLOAT-LONG, FLOAT-EXTENDED** (representation and length of data item associated with):
190. **USAGE COMPUTATIONAL** clause (alignment and representation of data):
191. **USAGE DISPLAY** (size and representation of characters):
192. **USAGE FLOAT-SHORT, FLOAT-LONG, FLOAT-EXTENDED** (size and permitted range of value):
193. **USAGE FUNCTION-POINTER** clause (alignment, size, and representation of data; and allowable languages):
194. **USAGE INDEX** clause (alignment and representation of data):
195. **USAGE NATIONAL** (size and representation of characters):
196. **USAGE OBJECT REFERENCE** clause (amount of storage allocated):

B. Language element lists

197. **USAGE PACKED-DECIMAL** clause (computer storage allocation, alignment and representation of data):
198. **USAGE POINTER** clause (alignment, size, representation, and range of values):
199. **USAGE PROGRAM-POINTER** clause (alignment, size, and representation of data; and allowable languages):
200. **USE** statement (action taken following execution of the USE procedure when I-O status value indicates a fatal exception condition):
201. **User-defined words** (whether extended letters may be specified in user-defined words externalized to the operating environment):
202. **Variable-length data items** (actual time when the resources used are freed):
203. **WRITE** statement (mnemonic-name-1):
204. **WRITE** statement (page advance when mnemonic-name-1 specified):

B.2. Optional language element list

1. **ACCEPT and DISPLAY screen handling:** Complete support is claimed with the exception of OCCURS items.
2. **ARITHMETIC IS STANDARD:** No support is claimed.
3. **Dynamic capacity tables:** The syntax is recognised, but no functionality is claimed.
4. **DYNAMIC LENGTH elementary items:** No support is claimed.
5. **Extended letters:** No support is claimed.
6. **File sharing and record locking:** Support is claimed, but the level of support is processor-dependent.
7. **FORMAT and SELECT WHEN file handling:** No support is claimed.
8. **Locale support and related functions:** The syntax is recognised, but no functionality is claimed.
9. **Object orientation:** No support is claimed.
10. **Report writer:** Complete support is claimed.
11. **RESUME statement:** No support is claimed.
12. **REWRITE FILE and WRITE FILE:** Complete support is claimed.
13. **VALIDATE:** No support is claimed.

B.3. Non-standard extension list

Many extensions found in other compilers. Some own extensions.

Note: Flagging syntax extensions You can flag many syntax extensions when compiling with `-std=cobol2014` or one of the other strict COBOL syntax definitions.

B. Language element lists