





**COBOL class definition structure**

**COBOL method definition structure**

CLASS-ID paragraph  
METHOD-ID paragraph

**Millennium Language Extensions and  
date fields**

Millennium Language Extensions syntax  
Terms and concepts



DAY-TO-YYYYDDD . . . . .	424
FACTORIAL . . . . .	425
INTEGER . . . . .	426
INTEGER-OF-DATE . . . . .	427
INTEGER-OF-DAY . . . . .	428
INTEGER-PART . . . . .	429
LENGTH . . . . .	430
LOG 4329 . w(43)T. . . . .G. i238465 . . . . .	431
LOG10 . . . . .	432

## Compiler directives

### CALLINTERFACE

UNDATE

YEAR-TO-YYYY  
YEARWINDOW

BASIS statement  
CBL (PROCESS) statement  
\*CONTROL (\*CBL) statement

DELETE statement  
EJECT statement

INSERT statement  
READY or RESET TRACE statement

SERVICE LABEL statement  
SERVICE RELOAD statement  
SKIP1/2/3 statements  
TITLE statement

---

## About this book

This book presents the syntax of COBOL for OS/390 & VM, COBOL Set for AIX, and VisualAge COBOL (collectively referred to in this book as IBM COBOL). To indicate platform-specific information, this book use the following methods:

IBM extensions in text are shown this way.

















In some cases, the basic character set is extended with the national character set.



**COBOL words with multi-byte characters**





External class-name

Under OS/390 and VM, the only DBCS character string system-name allowed is computer-name.



if the QUOTE compiler option is in effect  
or  
The apostrophe character ('), if the APOST compiler option is in effect  
or an  
apostrophe

INVOKE,

Under AIX and Windows, you cannot specify the SYMBOLIC CHARACTER clause if a DBCS or EUC code page is indicated by the locale setting. For more information on locale, see Appendix F, “Locale considerations (workstation only)”

INVOKE,

or INVOKE of a method,

Programs that possess the RECURSIVE attribute

Programs compiled with the THREAD option (Workstation only)

or

method

ADDRESS OF (for each record in the Linkage Section)















Under OS/390 and CMS, it has the implicit definition:

```
1      SORT-MESSAGE GLOBAL PICTURE X(8)  USAGE DISPLAY VALUE "SYSOUT"
```

You can use the SORT-MESSAGE special register to specify the ddname of a data set that the sort utility should use in place of the SYSOUT data set.

The ddname specified in SORT-MESSAGE is equivalent to the name specified on the 'MSGDDN=' control statement in the sort control file.

You can specify the SORT-MESSAGE special register in a function wherever an alphanumeric argument is allowed.

TALLY





to 'F', inclusive. Two hexadecimal digits represent a single character in the





















---

**Area A**

end class, and end method

## End Class, and End Method

definition. class definition, or method

```
END CLASS CLASS-NAME.  
END METHOD METHOD-NAME.
```

### For classes

Class-name must be identical to the class-name of the corresponding CLASS-ID paragraph.

### For methods

Method-name must be identical to the method-name of the corresponding METHOD-ID paragraph.







Area A or Area B

**Blank lines**

A

**For classes and methods**









## **Referencing data names, copy libraries, and Procedure Division names**



## **Uniqueness of reference**

Procedure Division names that are explicitly referenced in a program must be unique within a section. A section-name, described under “Procedures” on page 208, is the highest and only qualifier available for a paragraph-name and









## Uniqueness of reference

As an IBM ext/GSpion, IBMindex-name cIBMbe used to reference IBy table. However,























---

COBOL program structure

---

**COBOL program structure**

ID

RECURSIVE

ID

## COBOL program structure

---

### **Nested programs**

A COBOL program can contain other COBOL programs, which in turn can contain



## COBOL program structure



## COBOL class definition structure

A COBOL class definition describes a class or a metaclass. A class definition







CLASS-ID paragraph  
METHOD-ID paragraph

method, class definition, and method definition. class, or  
class, or method

or ID  
DIVISION

and as an IBM extension, can appear in any  
order.

**Class IDENTIFICATION DIVISION**







If program-name is a nonnumeric literal, it can be up to 160 characters in length. It cannot be a figurative constant.













A DBCS string must be preceded by a shift-out control character and followed by a shift-in control character. For example:F34

## COBOL Language Reference



and classes,













Under AIX and Windows, you cannot specify the ALPHABET

## **ALPHABET clause**

**literal-1**

**literal-2**

**literal-3**

Specifies that the collating sequence is to be determined by the program,  
according to the following rules:























**ASSIGN clause**

**S-**



## **ASSIGN clause**

### **Environment variable contents for an indexed, relative or sequential VSAM**

**file:** For an indexed, relative or sequential VSAM file, the environment variable must contain a DSN option in the following format:



**USING data-name-9**

## **ASSIGN clause**

In this example, `my-comment-am` is the comment, and

## **ORGANIZATION clause**

The value of the file ID specified with a data-name is obtained when the file is OPENed. On each subsequent OPEN for the file, the value is reobtained.

File declarations for an external file must have the same file-system identifier.

The RESERVE clause is not supported for line-sequential files.

Under AIX and Windows,

**ORGANIZATION IS LINE SEQUENTIAL (format 4)**

Supported under all platforms except VM: a predecessor-successor relationship among the records in the file is established by the order in which records are placed in the file when it is created or extended. A record in a LINE SEQUENTIAL file can consist only of printable characters.







## **ACCESS MODE clause**

**r6YNAMICclauses**

### **Line-sequential files**

Same as for sequential files (described above).











Under OS/390 and VM, data-name-8 must be defined as a group item of 6 bytes in the Working-Storage or Linkage Section of the Data Division.

Specify data-name-8 only if the file is a VSAM file (that is, ESDS, KSDS, RRDS).

Under OS/390 and VM, for VSAM files the 6-byte VSAM return code is comprised of the following:





**SORT/MERGE considerations:**

When the RERUN clause is specified in the I-O-CONTROL paragraph,





**APPLY**

---

Local-Storage Section

DATE FORMAT clause

RECORDING MODE clause

and methods.

or method

classes,

**Note:** A method File Section can define EXTERNAL files only. A single run-unit



**Note:**













**External floating-point**

The data is aligned on the leftmost digit position; the exponent is adjusted accordingly.

For internal floating-point items, the size of the item in storage is determined by its USAGE clause. USAGE COMPUTATIONAL-1 reserves 16 bytes of storage for the









BLOCK

*integer-2*      CHARACTERS



## BLOCK CONTAINS clause

Two programs in a run unit can reference global file connectors in the following circumstances:

1. An external file connector can be referenced from any program that describes that file connector.
2. If a program is contained within another program, both programs can refer to a global file connector by referring to an associated global file-name either in the containing program or in any program that directly or indirectly contains the containing program.

---

## BLOCK CONTAINS clause

The BLOCK CONTAINS clause specifies the size of the physical records. The characters in the BLOCK CONTAINS clause reflect the number of bytes in the record.

For example, if you have a block with 10 DBCS characters, the BLOCK CONTAINS clause should say `BLOCK CONTAINS 2 CHARACTERS`.

If the records in the file are not blocked, the BLOCK CONTAINS clause can be omitted. When it is omitted, the compiler assumes that records are not blocked. Even if each physical record contains only one complete logical record, coding `BLOCK CONTAINS 1 RECORD` would result in fixed blocked records.

The BLOCK CONTAINS clause can be omitted when the associated File Control entry specifies a VSAM file; the concept of blocking has no meaning for VSAM files; the clause is syntax checked, but it has no effect on the execution of the program.

For EXTERNAL files, the value of all BLOCK CONTAINS clauses of corresponding EXTERNAL files must match within the run unit. This conformance is in terms of character positions and does not depend upon whether the value was specified as CHARACTERS or as RECORDS.

**integer-1, integer-2**



Under OS/390, the RECORD CONTAINS 0 CHARACTERS clause can be specified for input QSAM files containing fixed-length records; the



LABEL RECORD IS data-name

**data-name-2**

User labels are present in addition to standard labels. Data-name-2 specifies the name of a user label record. Data-name-2 must appear as the subject of a record description entry associated with the file.









Use the RECORD CONTAINS *integer*

The CODE-SET clause is syntax checked, but hec.09 Tl- Dfect on tCODexecution of tCO

*date-format-clause*



**BLANK WHEN ZERO clause**

ZEROS  
ZEROES







## DATE FORMAT clause

When a windowed date field contains a trigger in this way, it is expanded as if the



## **DATE FORMAT clause**

In class conditions

In sign conditions

In DISPLAY statements

## **Language elements that do not accept windowed date fields as arguments**

Windowed date fields cannot be used as:



For items described as USAGE IS POINTER, USAGE IS



Data-name-2 cannot be a windowed date field.

Under OS/390 and VM, a key can have COMPUTATIONAL-1, COMPUTATIONAL-2, COMPUTATIONAL-3, or COMPUTATIONAL-4 usage.

Under AIX and Windows, a key can have COMPUTATIONAL-1, COMPUTATIONAL-2, COMPUTATIONAL-3, COMPUTATIONAL-4, or COMPUTATIONAL-5 usage.





















either in the CURRENCY compiler option or

If the CURRENCY SIGN clause is specified, the CURRENCY and NOCURRENCY compiler options are ignored. If the CURRENCY SIGN clause is not specified and

## PICTURE clause

A currency symbol can be used only to define a numeric-edited item with USAGE DISPLAY.

In the following description of the PICTURE clause, *cs*

G N

The symbol G or N can appear alone in the PICTURE character-string.

E

DBCS items  
External floating-point items





































## **Slack bytes**

There are two types of slack bytes:

**Slack bytes** are the 14r3TypesadT ei3 crequirTj ems, if necessins, howTm( )T 14 ISnBec/[(ED claus

USAGE IS POINTER, USAGE IS PROCEDURE-POINTER, USAGE IS  
OBJECT REFERENCE, and COMPUTATIONAL-1 data items  
for COMPUTATIONAL-2 data items.















External floating-point

If the ARITH(EXTEND) compiler option is in effect, then the maximum length of an external decimal item is 31 digits.

**Effect of CHAR(EBCDIC) compiler option (workstation only):** Character data











For information on floating-point literal values, see









## VALUE clause



class definitird













A nondeclarative procedure can be referenced in a GO TO statement in a declarative procedure.

You can include a statement that executes a previously called USE procedure that

If there are no declaratiwrs (format-2), a pratgraph-name isno t required in her Procedue nDivision.

As an IBM extension, all pratgraphs dono t need to be contained within sections, even if one or moe npratgraphs re nso contained.Arithmetic expressions



## Arithmetic with date fields

Arithmetic operations that include a date field are restricted to:

- Adding a non-date to a date field

- Subtracting a non-date from a date field

Arithmetic statements that specify a year-last date field, except as a receiving





If the year part does not fall within the century window, then the receiving field is unmodified, and the size error imperative statement is executed when any remaining arithmetic operations are complete.

For example:





## Conditional expressions

or internal-decimal

DBCS

DBCS  
KANJI

NOT DBCS  
NOT KANJI

Condition-names with DBCS and floating-point values are allowed.

**Condition-name conditions and windowed date field**







Under AIX and Windows, comparisons of DBCS data items and literals are based on a collation sequence according to the COLLSEQ compiler option:

If the COLLSEQ(NATIVE) compiler option is in effect, then the collating sequence is determined by the locale. For information on the locale, see Appendix F, “Locale considerations (workstation only)” on page 515.





	IFP	EFP	FPL
		NN	
		NN	
		NN	
		NN	
		NN	
		NN	
		NN	
		NN	
	NU	NU	
	NU	NU	
	NU	NU	NU
	NU	NU	NU
	NU	NU	NU
	NU	NU	NU
Internal floating-point (IFP)			

## Conditional expressions





## Comparison of DBCS operands



## Date fields in sign conditions

The operand in a sign condition can be a date field, but is treated as a non-r 0 /GS the sign condition test. Thus, if the operand is an identifier of a windowed r 0 field, r 0 windowing is not done, so the sign condition can be used to test a windowed r 0 field/GS an all-zero value.

However, if the operand is an arithmetic expression, then any windowed r 0 fields in the expression will be expanded during the computation of the arithmetic result, prior to using the result/GS the sign condition test.

FGS example, given that:

- Identifier WIN-DATE is defined as a windowed r 0 field, and contains a value of zero

- Compiler option DATEPROC is in effect

- Compiler option YEARWINDOW(*starting-year*) is in effect, with a *starting-year*













## Statement categories

---

*Table 31 (Page 2 of 2). Imperative statements*

---

or method

INVOKE



**or method**

INVOKE...ON EXCEPTION









If the ARITH(EXTEND) compiler option is in

**Multiple results**

When an arithmetic statement has multiple results, execution conceptually proceeds as follows:

The statement performs all arithmetic operations to find the result to be placed in the receiving items, and stores that result in a temporary location.





---

Table 35 (Page 2 of 4). Status key values and meanings

---

Under AIX and Windows, file status 39 is not supported for line-sequential files or Btrieve files.







GOBACK,



## **ACCEPT Statement**

### **System input device**

Record length of 80 characters is assumed even if a logical record length of other than 80 characters is specified.

The system input device is read until identifier-1 is filled or EOF is encountered. If the length of identifier-1 is not an even multiple of the system input device record length, the final record will be truncated as required. If EOF is encountered after data has been moved, and before

**environment-name**







---

**ADD statement**



**ADD statement**

**SIZE ERROR phrases**

Do not use the ALTER statement in programs that have the RECURSIVE attribute,

**ALTER statement**

Programs defined with the RECURSIVE attribute can execute a CALL statement













Identifier-1 can be alphabetic or zoned decimal data item. It cannot be a windowed date field.



WITH NO REWIND











**END-COMPUTE phrase**

This explicit scope terminator serves to delimit the scope of the COMPUTE statement. END-COMPUTE permits a conditional COMPUTE statement to be nested in another conditional statement. END-COMPUTE can also be used with an imperative COMPUTE statement.

For more informa.3(pr se t)Tj/120 1 Tf42.3730 0 TD0 Tc3 Tw“t



## **DELETE statement**

The DELETE statement removes a record from an indexed or relative file. For indexed files, the key can then be reused for record addition. For space, it then becomes available for new records.

When the DELETE statement is executed, the associated file must be opened in I-O mode.





*environment-name-1*







date fields.

Identifier-1 and identifier-2 cannot be

If identifier-3 or identifier-4 is a date field, then see “Storing arithmetic results that involve date fields”





**EVALUATE statement**

---

**EVALUATE statement**



For comparisons involving date fields, see

**EVALUATE statement**







As an IBM extension, the EXIT PROGRAM statement does not have to be the last statement in a sequence of imperative statements, but the statements following the EXIT PROGRAM will not be executed if a CALL statement is active.









## IF statement

---

## IF statement

The IF statement evaluates a condition and provides for alternative actions in the

END-IF can be specified with NEXT SENTENCE as an IBM extension.

END-IF can be specified with NEXT SENTENCE. However, if the NEXT SENTENCE phrase is executed, control will not pass to the next statement following the END-IF but instead will pass to the statement after the closest following period.



and DBCS

A floating-point data item or floating-point literal will be treated as if it is in the NUMERIC category.

**DBCS**  
**EGCS**

## **INSPECT statement**







## REPLACING phrase (formats 2 and 3)

This phrase fills all or portions of a data item with specified characters, such as spaces or zeros.

**identifier-3 or literal-1**

Is the **subject field**

An external floating point item

An external floating point item



























The MERGE statement is not supported under OS/390 Unix.

KEY data items can be floating-point items.  
KEY data items cannot be variably-located.

Under OS/390 and VM, if the KEY is a DBCS item, the sequence of the KEY values are based on the binary collating sequence of the hexadecimal values of the DBCS characters.

Under AIX and Windows, if the KEY is a DBCS item, then



STANDARD-2

## MERGE statement





**Floating-point**—includes internal floating-point items (defined as USAGE COMP-1 or USAGE COMP-2), external floating-point items (defined as USAGE DISPLAY), and floating-point literals.

**DBCS**—includes DBCS data items (defined explicitly or implicitly as USAGE DISPLAY-1) and DBCS literals.



External floating point	Internal floating point	DBCS <sup>1</sup>
-------------------------------	-------------------------------	-------------------









Format 3

--	--

## **OPEN statement**

The I-O phrase is not valid for line-sequential files.

As an IBM extension, the EXTEND phrase is allowed for files that specify the LINAGE clause.

If the PASSWORD clause is specified in the FILE-CONTROL entry, the password data item must contain the valid password before the OPEN statement is executed. If the valid password is not present, the OPEN statement execution is unsuccessful.



Table 46 shows the permissible input-output statements for line-sequential files. An 'X' indicates that the specified statement can be used with the open mode given at the top of the column.

line-sequential	and
-----------------	-----







## PERFORM statement

---

X PER

Imperative-statement-1 is optional as an IBM extension.



Imperative-statement-1 is optional as an IBM extension.

date fields.

These identifiers cannot be windowed

Floating-point data items and literals can be used anywhere a numeric data item or literal can be specified.PERFORM statementThe format 4 VARYING phrase PERFORM statement

## **Varying identifiers**

The way in which operands are increased or decreased depends on the number of variables specified. In the following discussion, every reference to identifier-n refers equally to index-name-n (except when identifier-n is the object of the BY phrase).

If identifier-2 or identifier-5 is subscripted, the subscripts are evaluated each time

## **PERFORM statement**

### **Varying two identifiers**

```
PERFORM PROCEDURE-NAME-1 THROUGH PROCEDURE-NAME-2  
  VARYING IDENTIFIER-2 FROM IDENTIFIER-3  
    BY IDENTIFIER-Q UNTIL CONDITION-3
```









PREVIOUS

PREVIOUS is only supported under AIX and Windows.



1. If the file referenced by file-name-1 is described as containing variable-length records, or as a QSAM file with RECORDING MODE

## **READ statement**

undefined. Figure 16 illustrates this concept. If the range of the current record exceeds the record description entries for file-name-1, the record is truncated on

If the AT END phrase is not specified, a USE AFTER STANDARD

## **READ statement**

For indexed files, the key sequence is the sequence of ascending values of the current key of reference. For relative files, the key sequence is the sequence of ascending values of relative record numbers for records that exist in the file.

Before the READ statement is executed, the file position indicator must be set by a successful OPEN, START, or READ statement. When the READ statement is executed, the record indicated by the file position indicator is made available, if it is still accessible through the path indicated by the file position indicator.

If the record is no longer accessible (because it has been deleted, for example), the executed,le. (or previous)

(or alternatively for AIX and

Windows, the PREVIOUS phrase)

(or alternatively for AIX and Windows, no previous record exists),

Under AIX and Windows see the discussion on PREVIOUS RECORD on 342.

**READ statement**

If the KEY phrase is not specified, the prime RECORD KEY becomes the key of reference for this request. When dynamic access is specified, the prime RECORD





**RELEASE statement**

When FROM identifier-1 is specified, the information is still available in identifier-1.

When control passes from the INPUT PROCEDURE, the sort file consists of all those records placed in it by execution of RELEASE statements.





The REWRITE statement is not supported for line-sequential files.

Record-name-1 can define a floating-point data item or DBCS data item.  
Identifier-1 must be a DBCS data item if record-name-1 is a DBCS data item.















A DBCS item if the ASCENDING/DESCENDING KEY is defined as a DBCS item.

A floating-point data item

A data item defined with USAGE IS POINTER, USAGE IS PROCEDURE-POINTER, or USAGE IS OBJECT REFERENCE

A windowed date field



Setting USAGE IS POINTER data items to a data address







**identifier-4**

Receiving fields.

Must be described as USAGE IS POINTER.

**ADDRESS OF identifier-5**

Receiving fields.

**identifier-5**













## **SORT Statement**

of the records contained in the input files must not be greater than the largest record described for file-name-1. For more information, see the *IBM COBOL Programming Guide* for your platform.

## **INPUT PROCEDURE phrase**





## **SORT special registers**

The special registers, SORT-CORE-SIZE, SORT-MESSAGE, and SORT-MODE-SIZE,

LESS  
< THAN

NOT GREATER  
NOT > THAN  
LESS OR EQUAL







## STRING statement

None of the identifiers in a STRING statement can be windowed date fields.

### **identifier-1 through identifier-3**

Can be DBCS data items. If one of these identifiers is a DBCS item, then all of them, and all literals, must be DBCS items. Cannot be external floating-point items.

### **literal-1 and literal-2**

Can be DBCS literals. If one of these is a DBCS literal, then all of them must be DBCS literals, and identifier-1 through identifier-3 must be DBCS items.

SPACE is the only figurative constant allowed for DBCS items.





**STRING statement**

## **STRING statement**

When the POINTER phrase is not specified, no pointer is available to the user. However, a conceptual implicit pointer with an initial value of 1 is used by the system.

Conceptually, when the STRING statement is executed, the initial pointer value (explicit or implicit) is the first character position within the receiving field into







None of the identifiers in an UNSTRING statement can be windowed date fields.

As an IBM extension, identifier-1 can be reference-modified. It can be an

A DBCS data item

If any are DBCS items, then all must be DBCS items. Figurative constants SPACE and SPACES are allowed for DBCS items.

A DBCS literal

If any are DBCS literals, all must be DBCS literals. Figurative constants SPACE and SPACES are allowed for DBCS literals.

or DBCS





When identifier-1 (the sending field) is a DBCS data item, identifier-7 indicates the relative DBCS character position (not the relative Dbyte position ending field).





**UNSTRING statement**

---

UN





The mnemonic-name phrase can also be specified for stacker selection with a card punch file. When using stacker selection, WRITE AFTER ADVANCING must be used.

As an IBM extension, you can specify both the ADVANCING PAGE and







## **WRITE statement**

For a 2-line print file, the lines are printed on line 1 (top edge of card) and line 3 (between rows 11 and 12). Line control cannot be specified. Automatic spacing is provided.

For a multiline print file, up to 25 lines of characters can be printed. Line control can be specified. If line control is not specified, automatic spacing is provided.

Line control is specified by issuing WRITE AFTER ADVANCING statements for the print function file. If line control is used for one such statement, it must be

**WRITE statement**

---

## Part 7. Intrinsic functions

**Intrinsic functions** . . . . .

DATE-TO-YYYYMMDD  
DATEVAL

DAY-TO-YYYYDDD

UNDATE

YEAR-TO-YYYY  
YEARWINDOW

## Intrinsic functions

---

Argument-1 cannot be a windowed date field, except in the UNDATE intrinsic function.



An argument cannot be a DBCS literal or data item. See “Function definitions” on



Floating-point literals are allowed wherever a numeric argument is allowed, and in arithmetic expressions used in functions that allow a numeric argument. They are *not*





The behavior of functions marked “DP”

DP			
DATE-TO-YYYYMMDD <sup>DP</sup>	I1, I2	I	Standard date equivalent (YYYYMMDD) of I1 (standard date with a windowed year, YMMDD), according to the 100-year interval whose ending year is specified by the sum of I2 and the year at execution time
DATEVAL <sup>DP</sup>	I1 or X1	I	Date field equivalent of I1 or X1
DP			
DAY-TO-YYYYDDD			

**Intrinsic functions**

*Table 51 (Page 2 of 3). Table of functions*

Function name	Arguments	Type	Value returned
---------------	-----------	------	----------------









**ASIN**























If the

INTEGER

---

INTEGER



Under OS/390 and VM, the INTDATE compiler option affects the starting date for the integer date functions. For details, see the











**LOWER-CASE**

**MAX**

---

If more than one argument-1 is specified, the combination of alphabetic and alphanumeric arguments is allowed.













## NUMVAL

---

## NUMVAL

The NUMVAL function returns the numeric value represented by the alphanumeric character string specified in each argument.

If the ARITH(EXTEND) compiler option is in effect, then the total number of digits must not exceed 31.













## **PRESENT-VALUE**

---

**RANDOM**

The RANDOM function returns a numeric value that is a pseudorandom number from a rectangular distribution.

The function type is numeric.



---

**REM**

The REM function returns a numeric value that is the remainder of argument-1 divided by argument-2.

The function type is numeric.

**Format**





**SQRT**





**SUM**

---

**SUM**

The SUM function returns a value that is the sum of the arguments.



UNDATE

---

UNDATE

**UPPER-CASE**

**VARIANCE**

---

WHEN-COMPILED

YEAR-TO-YYYY

---

YEAR-TO-YYYY

















*literal-1*

*literal-2*

SUPPRESS



Under AIX and Windows:





However, when operand-1 consists solely of a separator comma or semicolon, it participates in the match as a text-word (in













As an IBM extension, pseudo-text-1 can consist entirely of a separator comma or a separator semicolon.

The REPLACE statement can be used with DBCS literals and DBCS or EUC names. Pseudo-text can contain DBCS or EUC character-strings, but the characters cannot be continued across lines.











As IBM extensions to the COBOL 85 Standard, the following apply to declarative procedures:

For AIX, Windows, OS/390, and VM:





The UNIT phrase is not applicable to files in the random access mode, because only FILE labels are processed in this mode.

**file-name-1**





### Compiler directives

A **compiler directive** is a statement that causes the compiler to take a specific action during compilation.

Currently, `CALLINTERFACE` is the only compiler directive supported.

---

### CALLINTERFACE

The `CALLINTERFACE` directive specifies the interface convention for

**CALLINTERFACE**

**DESC, DESCRIPTOR**

**CALLINTERFACE**

(Section A)  
:  
>>CALLINTERFACE OPT  
(Section B)  
:  
>>CALLINTERFACE DESC  
(Section C)

the following specifications are in effect:





---

## Part 9. Appendixes





**Compiler limits**

*Table 54 (Page 3 of 4). Compiler limits*

Language element	Compiler limit
------------------	----------------



## **EBCDIC and ASCII collating sequences**



## ASCII code values





## ASCII code values

*Table 56 (Page 3 of 4). ASCII collating sequence*

<b>Ordinal Number</b>	<b>Symbol</b>	<b>Meaning</b>	<b>Decimal Representation</b>	<b>Hex Representation</b>
71	F		70	46
72	G		71	47
73	H		72	48
74	I		73	49

**ASCII code values**









**Reserved words**

<i>Table 57 (Page 3 of 6). Reserved words</i>			
Reserved word	IBM COBOL	Standard only	RFD
EGI		X	

<i>Table 57 (Page 3 of 6). Reserved words</i>			
Reserved word	IBM COBOL	Standard only	RFD
EXTENDord			





**Reserved words**

<i>Table 57 (Page 5 of 6). Reserved words</i>			
Reserved word	IBM COBOL	Standard only	RFD

<i>Table 57 (Page 5 of 6). Reserved words</i>			
Reserved word	IBM COBOL	Standard only	RFD

Table 57 (Page 6 of 6). Reserved words

--	--







A *locale* different countries.

---

## **Appendix G. Summary of language difference: host COBOL**





## Appendix H. Industry specifications

|

3.











# IBM VisualAge COBOL

*Fact Sheet, GC26-9052*

*Getting Started on Windows, GC26-8944*

*Language Reference, SC26-5046*  
~~*GC26-8944*~~



---

## Glossary

The terms in this glossary are defined in accordance

**assignment-name**

\* **class condition.** The proposition, for which a truth

—

**CORBA**

followed by the key words END DECLARATIVES. A declarative is composed of a section header, followed

**enclave**

\* **external switch.** A hardware or software device, defined and named by the implementor, which is used



H

(2) A numeric data item defined in the Data Division that does not include any digit positions to the right of the decimal point.

(3) A numeric function whose definition provides that all digits to the right of the decimal point are zero in







**parameter.** Parameters are used to pass data values



**\* relative record number**



Sign condition

**Single Byte Character Set (SBCS).** A set of characters in which each character is represented by a single byte.  
See also "

\* **subscript.** An occurrence number represented by either an integer, a data-name optionally followed by an integer with the operator + or -, or an index-name optionally followed by an integer with the operator + or -, that identifies a particular element in a table. A subscript may be the word ALL when the subscripted identifier is used as a function argument or a function

an OCCURS DEPENDING ON clause in its data

---

## Index

### Special Characters

- , (comma)
  - insertion character 171
  - symbol in PICTURE
    - clause 163, 166
- / (slash)
  - insertion character 171
  - symbol in PICTURE
    - clause 163, 166
- (/) comment line 33
- (period) 163
- <= (less than or equal to) 219





condition (*continued*)





DOWN BY phrase, SET  
statement 363

figurative constant (*continued*)



# K

Kanji 216





- program-name, rules for
  - referencing 61
- program, separately-compiled 58
- programming notes
  - ACCEPT statement 250
  - altered GO TO statement 258
  - arithmetic statements 242
  - data manipulation
    - statements 379, 386
  - DELETE statement 275
  - DISPLAY statement 279
  - EXCEPTION/ERROR
    - procedures 484





SET statement (*continued*)



USAGE clause (*continued*)

description 187

DISPLAY phrase 190

DISPLAY-1 phrase 190

elementary item size 129



---









