



- D *name*[=*expr*]  
Define a CDF name (for use with >>**IF**) to have the value of *expr*.
- E  
Write the CDF-processed COBOL input to standard output in *free-form reference format*. Certain non-COBOL markers are included in the output to indicate where copybook files were included. For line-number consistency with the input, blank lines are retained.  
  
Unlike the C compiler, This option does not prevent compilation. To prevent compilation, use the **-fsyntax-only** option.
- fdefaultbyte=*value*  
Use *value*, a number between 0 and 255, as the default value for all WORKING-STORAGE data items that have no VALUE clause. By default, alphanumeric data items are initialized with blanks, and numeric data items are initialized to zero. This option overrides the default with *value*.
- fsyntax-only  
Invoke only the parser. Check the code for syntax errors, but don't do anything beyond that.
- copyext *ext*  
For the CDF directive  
COPY *name*  
if *name* is unquoted, several varieties of *name* are tried, as described below under *Copybooks*. The **-copyext** option extends the names searched to include *ext*. If *ext* is all uppercase or all lowercase, both forms are tried, with preference given to the one supplied. If *ext* is mixed-case, only that version is tried. For example, with  
-copyext .abc  
given the CDF directive  
COPY *name*  
**gcobol** will add to possible names searched *name*.abc and *name*.ABC in that order.
- ffixed-form  
Use strict *fixed-form reference format* in reading the COBOL input: 72-character lines, with a 6-character sequence area, and an indicator column. Data past column 72 are ignored.
- ffree-form  
Force the COBOL input to be interpreted as *free-form reference format*. Line breaks are insignificant, except that '\*' at the start of a line acts as a comment marker. Equivalent to **-indicator-column 0**.
- findicator-column  
describes the location of the Indicator Area in a COBOL file in *Reference Format*, where the first 6 columns — known as the "Sequence Number Area" — are ignored, and the 7th column — the Indicator Area — may hold a character of significance to the compiler.  
  
Although *reference format*, strictly speaking, ignores data after column 72, with this option **gcobol** accepts long COBOL lines, sometimes known as *extended source format*. Text past column 72 is treated as ordinary COBOL text. (Line continuation remains in effect, however, provided no text appears *past* column 72.)  
  
There is no maximum line length. Regardless of source code format, the entire program could appear on one line.  
  
By default, **gcobol** auto-detects the source code format by examining the line that contains the text "program-id". When there are characters on past column 72 on that line, the file is assumed to be in *extended source format*, with the indicator area in column 7. Otherwise, columns 1-6 are examined. If those characters are all digits or blanks, the file is assumed to be in *fixed-form reference format*, also with the indicator in column 7. If not auto-detected as *fixed-form reference format* or *extended source format*, the file is assumed to be in *free-form reference format*.



`-dialect dialect-name`

By default, **gcobol** accepts COBOL syntax as defined by ISO/IEC 1989:2023, with some extensions for backward compatibility with COBOL-85. Additional syntax is supported with this option. The value of *dialect-name* may be

**ibm** to indicate IBM COBOL 6.4 syntax:

- **EJECT**
- **EQUAL** as assignment operator
- **LENGTH OF**
- **PROCEDURE POINTER**
- **SECTION** segment
- **STOP** <number>
- **VOLATILE**
- Per-program Registers
 

<b>RETURN-CODE</b>	S9(4)
<b>SORT-CONTROL</b>	X(160)
<b>SORT-CORE-SIZE</b>	S9(8)
<b>SORT-FILE-SIZE</b>	S9(8)
<b>SORT-MESSAGE</b>	X(8)
<b>SORT-MODE-SIZE</b>	S9(5)
<b>SORT-RETURN</b>	S9(4)
<b>TALLY</b>	9(5)
<b>WHEN-COMPILED</b>	X(16)
<b>XML-CODE</b>	S9(9)
<b>XML-EVENT</b>	X(30)
<b>XML-INFORMATION</b>	S9(9)
<b>XML-NAMESPACE</b>	X(0) to X(32,768)
<b>XML-NNAMESPACE</b>	N(0) to N(16,384)
<b>XML-NAMESPACE-PREFIX</b>	X(0) to X(4,096)
<b>XML-NNAMESPACE-PREFIX</b>	N(0) to N(2,048)
<b>XML-NTEXT</b>	N(0) to N(2,000,000)
<b>XML-TEXT</b>	X(0) to X(2,147,483,646)

**gnu** to indicate GnuCOBOL syntax, generally compatible with MicroFocus.

**mf** to indicate MicroFocus syntax:

- **BINARY-LONG-LONG**
- **CALL ... GIVING**
- **CDF \$IF**
- **COMPUTATIONAL-6**
- **COMPUTATIONAL** used with **PICTURE X**
- **INSPECT ... TRAILING**
- **OCCURS** at **LEVEL 01**
- **LEVEL 78** constants
- **MOVE POINTER**
- **RETURNING** <number>
- **USAGE IS TYPENAME**

`-include filename`

Process *filename* as if

COPY "*filename*"

appeared as the first line of the primary source file. If *filename* is not an absolute path, the directory searched is the current working directory, not the directory containing the main source file. The name is used verbatim. No permutations are applied, and no directories searched.









### Command-line Arguments

To read command-line arguments, use the registers **COMMAND-LINE** and **COMMAND-LINE-COUNT** in an **ACCEPT** statement (only). Used without a subscript, **COMMAND-LINE** returns the whole command line as a single string. With a subscript, **COMMAND-LINE** is a table of command-line arguments. For example, if the program is invoked as

```
./program -i input output
```

then

```
ACCEPT target FROM COMMAND-LINE(3)
```

moves *input* into *target*. The program name is the first thing in the whole command line and is found in **COMMAND-LINE(1)** **COMMAND-LINE** table.

To discover how many arguments were provided on the command line, use

```
ACCEPT target FROM COMMAND-LINE-COUNT
```

If **ACCEPT** refers to a nonexistent environment variable or command-line argument, the target is set to **LOW-VALUES**.

The system command line parameters can also be accessed through the **LINKAGE SECTION** in the program where execution starts. The data structure looks like this:

```
linkage          section.
01  argc         pic 999.
01  argv.
    02  argv-table occurs 1 to 100 times depending on argc.
    03  argv-element pointer.
01  argv-string  pic x(100) .
```

and the code to access the third parameter looks like this

```
procedure division using by value argc by reference argv.
set address of argv-string to argv-element(3)
display argv-string
```

### #line directive

The parser accepts lines in the form

```
#line lineno "filename".
```

The effect is to set the current line number to *lineno* and the current input filename to *filename*. Pre-processors may use this directive to control the filename and line numbers reported in error messages and in the debugger.

### SELECT ... ASSIGN TO

In the phrase

```
ASSIGN TO filename
```

*filename* may appear in quotes or not. If quoted, it represents a filename as known to the operating system. If unquoted, it names either a data element or an environment variable containing the name of a file. If *filename* matches the name of a data element, that element is used. If not, resolution of *filename* is deferred until runtime, when the name must appear in the program's environment.

### XML PARSE

**gcobol** emulates the IBM **XML PARSE** statement. The following values for **XML-EVENT** are defined:



















modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- \* Neither the name of the Symas Corporation nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.