

To: Distribution
From: F. C. Helwig
Date: October 5, 1978
Subject: COBOL Source Text Manipulation.

1. Introduction

The COBOL source text manipulation facility consists of the following statements:

- (1) The COPY Statement
- (2) The REPLACE Statement

These statements are briefly described in Attachment 1. The COPY statement is part of ANSI-COBOL. The REPLACE Statement is described in the COBOL Journal of Development and will become part of a future standard.

2. The COPY Statement

The simple COPY statement (i.e. a COPY statement not containing the REPLACING option) is supported by the released COBOL compiler (Version 3.0). Support for the COPY REPLACING statement will be provided by the proposed Multics command

```
expand_cobol_source, ecs
```

which is intended to replace the existing Multics command

```
format_cobol_source, fcs
```

The proposed command is briefly described in Attachment 2 and is a generalization of with the fcs command. The fcs command contains undocumented options and should be preserved.

If the ecs command is invoked either directly or indirectly then a source segment whose name ends in ".ex.cobol" is produced which does not contain COPY statements.

3. The REPLACE Statement

Support for the REPLACE statement will also be provided by the proposed ecs command.

4. Changes to the cobol Command

Multics Project internal working documentation. Not to be reproduced or distributed outside the Multics Project.

A new option

-expand, -exp

is provided for the cobol command. This option along with the existing option

-format, -fmt

will induce formatting and expansion (in that order) of the source module specified by the command. The resulting source module (whose entry name ends in ".ex.cobol") is permanently retained and is submitted to the ccompiler. Source expansion will also be induced if the first line in the source module consists of the character pair "%;". The first line does not appear in the resulting source module and formatting may or may not occur. A listing of the resulting source module must be obtained in order to conveniently perform source language debugging.

If the source file pathname does not end in the suffix ".ex.cobol" or ".cobol" then these suffixes are assumed in the stated order. If the suffix ".ex.cobol" is used then

- (1) Formatting and expansion will not be induced.
- (2) COPY and REPLACE statements are illegal.
- (3) The initial line "%;" is illegal.

Attachment 1

Source Text Manipulation Facilities

The COBOL source text manipulation facility is composed of the COPY statement and the REPLACE statement. The effect of a COPY statement is to generate text from a COBOL library which becomes part of the source program. The effect of the REPLACE statement is to substitute new text for text appearing in the source program. Any REPLACE statements appearing in a source program are processed after the processing of all COPY statements contained in the source program. The syntactic correctness of the library text cannot be independently determined. Except for COPY and REPLACE statements the syntactic correctness of the entire COBOL source program cannot be determined until all COPY and REPLACE statements have been processed.

The Copy Statement

The COPY statement incorporates text from a COBOL library into a source program. The syntax of the statement follows:

```
COPY text_name OF library_name
[ REPLACING ( left_pseudo_text BY
              right_pseudo_text )... ]
```

The library text is scanned and replaces the COPY statement in the source program. Both the library text and the pseudo-text consist of a string of COBOL text words (character strings and separators except for space). If the sequence of text words defined by left_pseudo_text is found in the library then it is replaced by the sequence of text words defined by right_pseudo_text.

The Replace Statement

The REPLACE statement is used to replace text appearing in the source program. The syntax of the statement follows:

```
Format 1  REPLACE
          {left_pseudo_text BY right_pseudo_text}...
```

```
Format 2  REPLACE OFF.
```

Format 1 of the REPLACE statement defines a replacement algorithm identical to that defined by a similar COPY statement with the same REPLACING option. Format 2 of the REPLACE statement discontinues text replacement.

The example which follows illustrates the processing of a source program containing COPY and REPLACE statements. The example is meant to be instructive rather than realistic.

(1) Before Source Expansion:

source.cobol

```
.....  
replace  
copy lib replacing replacing by replace.  
.....  
move copy to replace.  
.....
```

lib.cobol

```
==copy== by ==XX==  
==replacing== by ==YY==.
```

(2) After COPY Statement Expansion

source.ex.cobol

```
.....  
replace  
==copy== by ==XX==  
==replace== by ==YY==.  
.....  
move copy to replace.  
.....
```

(3) After REPLACE Statement Expansion

source.ex.cobol

```
.....  
move XX to YY.  
.....
```

Attachment 2

expand_cobol_source, ecs

(1) Formatting

Converts pseudo free-format COBOL source programs to standard fixed-format source programs acceptable to the COBOL compiler

(2) Expansion

Converts standard format COBOL source programs containing COPY and REPLACE statements to equivalent source programs not containing these statements

(3) The ecs command is a generalization of the fcs command as documented in the MPM.

Usage

expand_cobol_source, ecs

path1 {-control_args}

where

path1 input segment pathname, altered to end in ".cobol" or ".ex.cobol"

The output segment pathname is derived from the input segment pathname and always ends in ".ex.cobol".

control_args

one of the following

-upper_case, -uc upper case conversion, implies formatting
-lower_case, -lc lower case conversion, implies formatting
-format, -fmt format the COBOL source
-expand, -exp expand the cobol source