

Software Release Notice
for
VS COBOL
Version 7.52.07

March 2000

Copyright Getronics Co., 2000



DISCLAIMER OF WARRANTIES AND LIMITATION OF LIABILITIES

The staff of Getronics has taken due care in preparing this manual; however, nothing contained herein modifies or alters in any way the standard terms and conditions of the Getronics purchase, lease, or license agreement by which this software package was acquired, nor shall Getronics or its subsidiaries be liable for incidental or consequential damages in connection with or arising from the use of the software package, the accompanying manual, or any related manuals.

NOTICE

All Getronics Program Products are licensed to customers in accordance with the terms and conditions of the Getronics Standard Program Products License; no ownership of Getronics Software is transferred and any use beyond the terms of the aforesaid License, without the written authorization of Getronics is prohibited.

1.0 Release Abstract

The WANG VS COBOL compiler validates and translates source programs written in the Full ANSI Standard COBOL 74 Language and produces a program file which may be executed on any WANG VS System. The COBOL Compiler supports the full range of functions offered to a programmer using the VS System, such as Symbolic Debugger support, full workstation I/O, full disk I/O, EDITOR support, etc.

2.0 Pre-Requisites and Dependencies

2.1 Hardware Pre-Requisites

The compiler may be installed and run on any VS configuration.

2.2 Software Pre-Requisites

Versions 7.52.00 and up of the COBOL compiler and the object code they create must be run on a VS Operating System version 7.52.00 or greater to have full YEAR 2000 support and take advantage of the new DATE4 keyword documented further in this document. If this is not a requirement the COBOL compiler may be run on VS Operating System 7.21.09 or Operating Systems 7.40 and up.

In order to compile COBOL HLI programs specifying the option PREP=YES on the Compiler's OPTIONS screen. To insure that the proper version of the PACE Database Management System is currently installed check with your local sales representative. Please be sure to read the CSRN for the PACE Database Management System for any restrictions or special considerations required.

3.0 Restrictions and Special Considerations Since Version 05.30.00

COBOL programs that take advantage of the DATE4 keyword of the ACCEPT statement must be run on VSOS 7.52.00 or higher. These COBOL programs may be compiled on any currently supported version of the VS Operating System. If programs containing the DATE4 keyword are run on an operating system prior to 07.52.00 the first two high order bytes of the year may contain undefinable information. In most cases the century digits will contain zeros, for example October 31, 1995 would be expressed as 00951031.

4.0 Enhancements

4.1 DATE4 Keyword for the ACCEPT statement Version 07.52.00

The DATE4 keyword allows the ACCEPT statement to retrieve the entire 4-digit year. The syntax of the ACCEPT statements has been extended to support the new DATE4 keyword. The syntax of the ACCEPT statement has been extended as shown below.

General Format

```
ACCEPT identifier FROM  
    DATE  
    DAY  
    TIME  
    DATE4
```

Syntax Rules

1. DATE4 is a new reserved word.

General Rules

1. DATE4 is composed of the data elements 4-digit year, month of the year, and day of the month. The sequence of the data element codes is from high order to low order (left to right). For example October 31, 1995 would be expressed as 19951031. DATE4, when accessed by a COBOL program behaves as if it had been described in the COBOL program as an unsigned, elementary, numeric integer of eight digits in length. If programs containing the DATE4 keyword are run on an operating system prior to 07.52.00 the first two high order bytes of the year may contain undefinable information. In most cases the century digits will contain zeros, for example October 31, 1995 would be expressed as 00951031.

5.0 Changes to be aware of

5.1 Version 7.52.06

(internal distribution only)

When running COBOL outside of the EDITOR or ADEPT, the SOURCE and SYMB defaults on the options screen both used to default to NO. These options are now defaulted to YES to mirror the forms file used by both the EDITOR and ADEPT as well as the C74START file used by ADEPT.

5.2 Version 7.52.07

An internal change now places the compilation time in a previously unused area in the lengths block of the object 0 output file. This can be used by LINKER

versions 7.52.05 and greater to determine DATESEL criteria as well as print out the compilation times on the linkage map.

6.1 Problems Corrected

6.1 Problems Corrected In Release 7.52.01

PTR	Problem
M410012602	STRING statements work as expected without PC01 or other abend type of problems.

6.1 Problems Corrected In Release 7.52.02

(internal distribution only).

	Problem
Internal	When COPY REPLACING had an error in the copy file and there were two or more copy statements and SOURCE was set to NO but PMAP was set to YES the compiler would fail. This situation has been corrected.
Internal	A proper error message is now received when attempting to perform a START on a STANDARD-DISK file without specifying an ALTERNATE key.

6.2 Problems Corrected In Release 7.52.03

CTS#	Problem
E091295	Although the compiler functions as documented on page 11-11 the restriction of only allowing a 240 byte printer file record has been removed. The COBOL74 compiler now allows a printer file record length with the same length restrictions for consecutive variable files (2024). The actual record length that can be specified is 2022 bytes because of the two control characters automatically added to the record length by DMS.
Internal	When testing the DATE4 function on OS 7.21.09 it was discovered that the 4 digit year being returned did not always contain valid data. Changes have been made to the code generation for the DATE4 function to guarantee that the date will contain zeros in the century digits and that the

year will be valid when DATE4 is being used on an operating system prior to 7.52.00.

6.3 Problems Corrected In Release 7.52.04

CTS#	Problem
Internal	<p>The UNSTRING statement no longer incorrectly picks up a delimiter found across the boundary of two adjoining fields. Example:</p> <pre>FIELD1 PIC X(8) VALUE "APICOPYP" FIELD2 PIC X(3) VALUE "RCX" F1 PIC X(4) VALUE SPACES F2 PIC X(4) VALUE SPACES</pre> <p>UNSTRING FIELD1 DELIMITED BY "PRC" INTO F1 DELIMITER IN F2</p> <p>Correct result is: F1 = "APICOPYP" F2 = spaces</p>
Internal	<p>When the same line caused a level 4 error and then a level 8 error only the level 4 error was being printed while the compilation failed with a level 8 error. Now all errors are printed to avoid confusion.</p>
E118203	<p>A level 16 severity error message is now returned when the size of the object code in combination with the size of the literal pool exceeds memory capacity. This can be caused by too large a code section, static area, or literal pool. Complex nested if statements can also be responsible for excessive code. To view the cause of the problem the user should be familiar with the use of the PMAP. By determining memory use by the program it is possible to best determine the proper remedy for the problem.</p>
E112423	<p>When a cobol program begins the first few operations taken is that register 2 is loaded with the starting address of the code section and register 14 is loaded with the starting address of the static area. A level 16 severity error message is now returned when the static area begins past address 0017FFFF. This is because the RLA Address instruction assumes that an address beginning with '018' is a negative address and converts the 018 to 008, loading register 14 with an erroneous address. Note: on the PMAP</p>

listing addresses start with 000 versus 001 so 0017FFFF would appear as 07FFFF on the PMAP listing.

6.4 Problems Corrected In Release 7.52.05

CTS#	Problem
Internal	<p>The correction in 7.52.04 for E118203 caused an incorrect level 16 error if the following conditions were encountered: An error is encountered during WC1PASS2 and the error is greater than the code generation stop flag. A value clause is processed by the compiler after the previous error condition.</p> <p>An incorrect level 16 error is not longer encountered under these conditions.</p>
E139671	<p>When utilizing the lineage clause and performing an 'After advancing PAGE' phrase the print line contained an advancement value of the lineage plus top values. This caused the page to advance incorrectly. The compiler now functions as in the 5.20 series did which puts a true top of form hex '8001' in the first print line followed by any top of page line spacing in the following print line.</p>

6.5 Problems Corrected In Release 7.52.06

None.

6.6 Problems Corrected In Release 7.52.07

None.

7.0 New Known Anamolies Since Release 07.52.03

None.

8.0 Documentation and Functional Clarifications Since Release 07.52.03

The RECORD CONTAINS clause page 11-11 should be changed to show:

PRINTER files 2022

9.0 Media Contents

VOLUME = COBL74 LIBRARY = COBOL74

<i>Module</i>	<i>Version</i>	<i>Description</i>
COBOL	7.52.07	
WC1PASS1	7.52.07	
WC1PASS2	7.52.07	
WC1\$COB	7.52.07	
C74FORM	32	COBOL 74 forms file
INSTALL		COBOL install Procedure

LIBRARY = CSRNLIB

COB75207 COBOL 74 SRN

10.0 Installation Instructions

If restoring from tape run the BACKUP utility and backup the library COBOL74 on volume COBL74 to the disk volume that will contain the COBOL compiler, then proceed to run the INSTALL procedure. If restoring from disk, the INSTALL procedure may be run directly from your installation disk. You may wish to delete the INSTALL procedure once installation is complete.

INSTALL will place the COBOL compiler and form file onto your system. The procedure will allow you to enter the volume and library on which to put the compiler. The form file (C74FORM) will be placed in the library @SYSFRM@ on your system volume. If a compiler or form file already exists, you will be prompted and allowed to save them before installation of your new software. The procedure will also allow you to customize compilation defaults specified thru the form by optionally running the SETOPTS utility. Any previously set defaults will be lost and the new ones retained. The SETOPTS compilation defaults are not available when compiling directly from COBOL but are for use when compiling from the EDITOR or ADEPT.