

NRcobol

cHArRiOTt

COLLABORATORS

	<i>TITLE :</i> NRcobol		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	cHArRiOTt	August 7, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	NRcobol	1
1.1	main	1
1.2	introduction	2
1.3	installation	2
1.4	usage	3
1.5	syntax	4
1.6	ok	5
1.7	a	6
1.8	b	6
1.9	programsyntax	6
1.10	identification	7
1.11	environment	8
1.12	data	8
1.13	procedure	9
1.14	example	10
1.15	later!	12
1.16	implemented?	12
1.17	history	13
1.18	todo	13
1.19	bugs	14
1.20	feedback	14
1.21	contacts	14
1.22	greetings	15
1.23	distribution	15
1.24	credits	16
1.25	disclimer	16
1.26	1.5 Author	17
1.27	sorry!	17
1.28	thank-you!	18

Chapter 1

NRcobol

1.1 main

NRcobol 1.0b Copyright ©1997 by cHARriOTt

Introduction

Installation

Usage

Syntax

Implemented?

History

Todo

Bugs

Feedback

Contacts

Salutations

Distribution

Credits

Disclaimer

Apologies

Thank You

NUff REsPEct.....peace n more time.

1.2 introduction

Why COBOL?

To those of you thinking not another new language, COBOL is in fact one of the oldest high-level languages, its roots going back to the late 1950's. Ofcourse there are those of you who already know this but regard COBOL as not so much an high-level language, as the original computer virus. To these people i say RESPECT the source of the mother tounge. (sic!)

When so many languages have come and gone (what ever happended to Modulus II), COBOL has seen in the advent of STRUCTURED, and now OBJECT methodologies, its adapted absorbing the latest in Database Management, SQL etcetera, and gone on from strength to strength. Indeed as an introduction to records there's alot to be said in its favour. So when people say why, i say why not?

This implementation of COBOL is largely ANSI 85 compatable and takes the form of MicroSoft(TM) COBOL, with maybe the exception of some stuff in ENVIRONMENT DIVISION - SPECIAL-NAMES. And some stuff left over from the days when memory was premium and no one had the imagination to contemplate speed. see

Implemented?

This would have been bought to you sooner, if it weren't for ←
the moron, in
the bedsit downstairs, with something of an habit!! which in lean times even extend to a fixation with gas. 'would you adam n eve it'.

WARNING dope really @&*\$% you up! something to do with short term memory going to pot!

STOP THE PRESS!.

"COMPUTING" MAY 8th 97

'Cobol crisis looms' The skills crisis is luring Cobol programmers out of retirement with offers of up to £1000 (\$1700) per day..... hmm :-)

1.3 installation

Installation

Nothing could be simpler, Just copy the runtime program 'RUNCOB' anywhere in your path, that's ram: c: or in the same directory as the created .INT files.

Files with the extension .INT are created as a result of successfully running NRCOBOL ..see

Usage

1.4 usage

Usage

After creating your cobol source file with your favourite text editor, and saving it with the .COB extention you have a choise of evoking the compiler from CLI/shell or Workbench.

From CLI/SHELL

NRCOBOL <sourcefile>.COB [-l] [-d]

where the optional flags:

- l produces a simple listing <sourcefile>.LST
- d produces a debug listing <sourcefile>.DBG

A successfull compilation produces <sourcefile>.INT to run type... RUNCOB <sourcefile>.INT

From Workbench

Use of the program from workbench is self explanatory, but for those too dim.

```
+-----+
|                                     |
|                                     | [.] [#]
|+-----+|
||Titles                             || | <-- What When Who
||                                     || |
||                                     || |
|+-----+|
|File: |                             || | <-- current sourcefile. ←
|   COB                               || | |
|   +-----+| |
|_____|| |
||                                     || |
|| IDENTIFICATION                       *   || | <-- status block
|| ENVIRONMENT                           *   || |
|                                     || |
|                                     || |
|                                     || |
|                                     || |
```

*'s replaced by

```

|| DATA * || | PASSED or FAILED
|| PROCEDURE * || |
|| || |
|| || | <-- current line
|+-----+| |
|+-----+ +----+ +---+ +-----+| | <-- ancillary output ↵
files
|| | List +----+ Debug +---+ | || | as .LST and/or . ↵
DBG
|| COMPILE | +-----+ +-----+ | EDITOR || | | | | |
|| | | LOAD | | EXIT | | ||_|
|+-----+ +-----+ +-----+ +-----+|=|
+-----+
^ ^ ^ ^
| | | |
| | | | Edit sourcefile.COB/LST/DBG
| | filerequester | |
Make it so #1 Exit program or
Abort compilation or
RUN succesful compilation (
whoops!
)

```

The Editor option allows one to view sourcefile.COB, sourcefile.LST or sourcefile.DBG. The file editor defaults to 'ED' as supplied in sys:c. You can however, set TEXTEDITOR in NRcobol's icon to launch your favourite editor. eg TEXTEDITOR=c:ced . Note no space between = and editor named.

In both cases an ICON is produced named <sourcefile>.INT to execute just double click.

The debug option produces a file which you may find useful in locating runtime errors, which sadly is normal for cobol. Usually this is to do with uninitialised variables, you know, finding alpha data in a numeric field.

Runtime errors usually return a value in the form 0x????, where 0x???? is an offset into the data space allocated in the Data Division. Stradling values, ie value between offset values indicates a postion within a table, (a subscripted identifier).

1.5 syntax

Syntax

I may at a later date add the syntax, in the mean time the following books are recommended:

- COBOL for the IBM Personal Computer by kip R Irvine
- COBOL for the 80's by J Wayne Spence
- Structured Programming in COBOL by B J Holmes
- COBOL 85 for programmers by Jim Inglis

OK go on then..

1.6 ok

The basics

The Syntax of the COBOL Language

At the moment COBOL has strict rules on where different parts of the program should be, a legacy of the days of coding sheets. So lets start with the ruler ←

The COBOL Ruler and the COBOL Source File

Program Syntax

```

          1           2           3           4           5           6 ←
          7
123456789012345678901234567890123456789012345678901234567890123456789
  ^ ^
  ||
  |

```

The A Margin

The B Margin

The seventh column is reserved for several special purposes. Those are: * / - D

When the 7th column contains an asterisk, the entire line is taken to be a comment.

A forward slash in the 7th column behaves in much the same way as an asterisk, in that the entire line is taken to be a comment. The only difference is that in the compiler listing [if one is produced] there will be a page break before the comment is printed. This can be used to enhance the readability of the source listing. See examples.

When the 7th column contains a hyphen, the line is taken to be a continuation of the previous line.
See examples.

The letter "D" in the 7th column flags the line as a debugging line. The line will be conditionally compiled for debugging. If the source-computer paragraph has the phrase WITH DEBUGGING MODE, the line will be compiled. This feature is used by some compilers to provide a program trace.

See example

.

1.7 a

The A Margin

Positions 8-11 is used in general for Division, Section and Paragraph names. The exception to this being the 01 level label in the data division.

See example

.

1.8 b

The B Margin

Positions 12-72 used for data and program code.
Note comment entry are allowed in some compilers from position 73+.

See example

1.9 programsyntax

Program Syntax

Any COBOL program is composed of exactly four divisions: identification, environment, data and procedure. These divisions appear explicitly in the COBOL source in the form:

division name DIVISION.

Each division may be further subdivided into sections and paragraphs.
For further information, select from the following list of divisions:

The IDENTIFICATION Division

The ENVIRONMENT division

The DATA division

The PROCEDURE division

If you have a problem remembering the exact order in which the divisions

must appear in the COBOL source file, try the mnemonic: "I Enjoy Data Processing".

See example

1.10 identification

The IDENTIFICATION Division

The identification division provides information about the program. Of this information, only the name of the program, specified in the PROGRAM-ID paragraph, is essential.

```
IDENTIFICATION DIVISION.
PROGRAM-ID. program-name.
[AUTHOR. comment.]
[INSTALLATION. comment]
[DATE-WRITTEN. comment]
[DATE-COMPILED. comment]
[SECURITY. comment]
```

The program name is just a user-defined name. It doesn't seem to have much of a use, and therefore some authors recommend that if at all possible [given OS constraints], the program name should be chosen to equal the name of the source file.

The compilation date paragraph contains the date of the most recent compilation.

Supposedly some COBOL compilers [DOES ANYONE KNOW WHICH??] will automatically update the comment in this paragraph each time the program is re-compiled.

A user-defined name is a sequence of characters satisfying the following conditions:

The length of a user-defined name cannot exceed 30 characters.

- A user defined name can contain letters, digits, or the hyphen.
- A user-defined name cannot begin with a hyphen.
- A user-defined name must contain at least one letter.

See example

1.11 environment

The ENVIRONMENT Division

The purpose of the environment division is to abstract out of the program all of its system-dependent aspects. Some of those aspects are:

Information for humans, on the type of computers used for development and execution of the given program.

Information as to the character set assumed throughout the program, as well as any special aliases the programmer might want to use.

Comment fields provide information on the development and target computers.

```
ENVIRONMENT DIVISION.  
[CONFIGURATION section]  
[INPUT-OUTPUT section]
```

See example

1.12 data

The DATA Division

```
DATA DIVISION.  
[SUB-SCHEMA SECTION.]  
[FILE SECTION.]  
[WORKING-STORAGE SECTION.]  
[LINKAGE SECTION.]  
[COMMUNICATION SECTION.]  
[REPORT SECTION.]
```

See example

1.13 procedure

The PROCEDURE Division

PROCEDURE DIVISION.

...

ACCEPT

ADD

CALL

CANCEL

CLOSE

COMPUTE

CONTINUE

DELETE

DISPLAY

DIVIDE

EVALUATE

EXIT

GO

IF
INITIALIZE
INSPECT
MERGE
MOVE
MULTIPLY
OPEN
PERFORM
READ
RELEASE
RETURN
REWRITE
SEARCH
SET
SORT
START
STOP
STRING
SUBTRACT
UNSTRING
USE
WRITE
See example

1.14 example

Example Cobol source

OK, this is short hello to cobol for more involed examples see the files (extension .cob) accompanying this release. The key thing to

note is the requirement of the verbs STOP RUN and the use of 'fullstops' to define the scope of a COBOL statement. LIN and COL are system defined.

```

*ROUTINE TO DISPLAY A MULTIPLICATION GRID.
~~~~~ IDENTIFICATION DIVISION.
~~~~~ PROGRAM-ID.  SECOND1.
~~~~~ AUTHOR.  C HARRIOTT.
~~~~~*
~~~~~ ENVIRONMENT DIVISION.
~~~~~*
~~~~~ DATA DIVISION.
~~~~~ WORKING-STORAGE SECTION.
~~~~~ 01  WS-X-SCREEN-DFT      PIC 99 VALUE  10.
~~~~~ 01  WS-Y-SCREEN-DFT      PIC 99 VALUE  05.
~~~~~ 01  WS-XY-PRODUCT        PIC 9(3).

      01  WS-COUNTERS.
          05  WS-Y-ROW          PIC 99.
~~~~~      05  WS-X-COLUMN        PIC 99.
~~~~~      05  WS-X-POS          PIC 99.
~~~~~      05  WS-Y-POS          PIC 99.

~~~~~*
~~~~~*
~~~~~ PROCEDURE DIVISION.
~~~~~ 001-MAIN.
          DISPLAY (1 , 10)
          "Routine to CALCULATE and DISPLAY a MULTIPLICATION TABLE".
          MOVE ZEROS TO WS-COUNTERS.
          PERFORM 002-ROWS
          VARYING WS-Y-ROW FROM 1 BY 1 UNTIL WS-Y-ROW > 10.
          STOP RUN.

~~~~~*
~~~~~*
~~~~~ 002-ROWS.
          MOVE ZEROS TO WS-X-COLUMN.
          PERFORM 003-COLUMNS
          VARYING WS-X-COLUMN FROM 1 BY 1 UNTIL WS-X-COLUMN > 10.

~~~~~*
~~~~~*
~~~~~ 003-COLUMNS.
          MOVE ZEROS TO WS-X-POS.
          MULTIPLY WS-X-COLUMN BY 5 GIVING WS-X-POS.
          ADD WS-X-SCREEN-DFT TO WS-X-POS.

~~~~~*
          MOVE ZEROS TO WS-Y-POS.
          MOVE WS-Y-ROW TO WS-Y-POS.
          ADD WS-Y-SCREEN-DFT TO WS-Y-POS.

~~~~~*
          MULTIPLY WS-Y-ROW BY WS-X-COLUMN
          GIVING WS-XY-PRODUCT.
          PERFORM 004-DISPLAY-PRODUCT.

      *
~~~~~*
~~~~~ 004-DISPLAY-PRODUCT.

```

```

        MOVE WS-Y-POS  TO LIN.
        MOVE WS-X-POS  TO COL.
~~~~~  DISPLAY (LIN , COL) WS-XY-PRODUCT.
        ~*
~~~~~*****END*****

```

1.15 later!

```

LATER!
*****

```

To be continued, but who knows when?

1.16 implemented?

```

                Implemented?
*****

IDENTIFICATION DIVISION.
        The program-id is required to be the same as the filename bar ↔
        the
        .COB file extention. This allows for consistent CALLs to subprogram
        and nested programs ie via their program-ids.

ENVIRONMENT    DIVISION.
        The only options supported in SPECIAL-NAMES are CURRENCY and ↔
        DECIMAL-POINT.
I-O-CONTROL SECTION. recognised but not implemented!

DATA          DIVISION.
        There is no support for the REPORT SECTION or communications.
        BLOCK and RECORD clauses recognised but not implemented!
        USAGE IS default always ie no support yet for PACKED / BINARY etc..
        SYNCHRONISED is recognised but not implemented!
        SCREEN SECTION is as defined by Microsoft 2.XX , i may add some supports
        for basic BUTTON GADGETS later!

PROCEDURE     DIVISION.
        .

```

All keywords above have been implemented!

CORR/CORRESPONDING not supported.

reference modification pending!

Note: if the program is evoked via the COMPILER's RUN gadget and the program has CALLS to a subprogram, then there is a chance the program may not find the subprogram in the 'current directory'. To avoid this 'EXCEPTION ERROR', either compile the program with the full path to the subprogram (ie CALL "path:currentdir/subprogram") or place the compiler NRCOBOL in the same directory as the program and its subprograms. The alternative to the above is simply to compile as normal and then double click on the created icon to evoke the program.

1.17 history

History

```

1.00  July      97 - first release CU Amiga Magazine
1.0a  August    97 - CU Amiga Magazine / Aminet
                    resort to pre 'reference moddification' version, since
                    last minute addition of reference modification option
                    introduced strange/terminal errors ...sorry!
1.0b  November  97 - font sensitive main program
                    NRCOBOL compiled under GNU C 270 ,just!
                    still trying with runcob (NorthC1.3 + A68k)
                    support only for A1200 or better
                    ...unless i get feedback from pre A1200 users :)
                    MOVE ALL ...
                    HIGH-VALUE(S) LOW-VALUE(S) sign sensitive
                    blank when zero should now work
                    fixed oversight with full stop now allowed in zone/margine ↔
                    A

                    relaxed LEVEL 02-49 positioning, now allowed in zone/ ↔
                    margine A

```

1.18 todo

Todo

- font sensitive main program
- font sensitive runtime screen
- own rather than system input
- Totally ANSI 85 see Implemented?
- Intrinsic functions ANSI 89
- support for SQL trainer module
- OBJECT????? Hmmmm :-)
- suggestions welcomed

Feedback

- fix reference modification for next release

1.19 bugs

Bugs

1.20 feedback

Feedback

- 'things aint wot they ougt a'
- unexpected output or syntax ... well you never know
- congratulations
- bribes not to continue
- buy outs / takeovers
- Donations of money / hardware.. down to last 7Mb 'again' of 120Mb HD ;-)
- or even a post card from what ever part of the planet you and your Amiga reside. Amiga dead? Long live Amiga.
- OBJECT COBOL example code???
- Spare copy of latest ANSI COBOL standard??
this comes via limited access to Ansi74 plus the stuff at the back of most books on COBOL
- Offers of work :-)

email : HARRIOTTTC@cardiff.ac.uk (new!)
email : HA.ROT@bigfoot.com (re-router/lifetime email curtesy of
<http://WWW.bigfoot.com/>)

snail mail

C Harriott
14a Splott Road
Splott
Cardiff
Cf2 2BZ
UK

Salutations

1.21 contacts

Contacts

For further information on the COBOL scene you might try the FAQ created and maintained by

jfid@mfltd.co.uk (James Fidell)
<http://www.mfltd.co.uk/~jfid/>
<http://www.mfltd.co.uk/FAQ/cobol-faq.html>

1.22 greetings

Hellos

A big 10:4 to the staff at ITEC (cardiff) especially S&M you know who your. Also thanks to Mike Evans at Cardiff Uni for the email account, however long it lasts.

Hello to Nick (Newport) 'i will get round to returning that HD cradle'

ADVERTISEMENT FOLLOWS 8-^

support your indepent local retailers
greetings to the staff at:

COMPUTER SUPERCENTRES LTD
31 Queens Arcade
Queen Street
Cardiff

15 David Street
Cardiff
Tel (01222) 255432

Where you're not just another customer...

1.23 distribution

Distribution

Distribution is granted to anyone via floppy and internet/ BBS provided documentation remains as is, only a nominal fee is charged for copying

and no attempt is made to reverse engineer the program.
Distribution is also granted to anyone compiling a CDROM provided the
above conditions are met AND i am sent a copy of the CDROM. Please??

1.24 credits

Credits

Would you believe i wrote this with the public domain 'C' compiler
NorthC 1.3 ...if you're out there S Hawtin thanks dude. Although i have to
say it makes for lazy programing. Hopefully GNU C will induce disipline!
Ouch!

Thanks expressed to Thore Böckelmann for the use of his disclaimer (quite
frankley i couldn't have put it better myself) also thanks for gifting
the Amiga community his EXCELENT CyberAVI and CyberQT. A real babe of a
program. SHoWING?? Now how about some avi/mov files AF/CUCD/ANOther?

Thanks to Markus Hillenbrand

Author

for the excellent GUICreator,

which was used to create (after much tweeking) the newface of NRCOBOL.
Please note the difference in compilation speeds is due solely to the use
of gadtools ;(

Thanks also to those unknown warriors who gave their all to bring us the
freedom to program in assembler.. the authors of A68K and BLINK
without which..... RESPECT!

1.25 disclaimer

Disclaimer

No warranty, either express or implied, is made with respect to the fitness
or merchantability of NRCOBOL.

chArRiOTt (referred to as "the author"), reserves the right to not
develop any future versions of NRCOBOL.

The author will try to make a good faith attempt at correcting any problems
if any are discovered, but is in no way required, nor bound to correct them.

The author neither assumes nor accepts any responsibility for the use or misuse of these programs. He will also not be held liable for damages or any compensation due to loss of profit or any other damages arising out of the use, or inability to use this program.

The author will not be liable for any damage arising from the failure of this program to perform as described, or any destruction of other programs or data residing on a system attempting to run the programs.

The user of this program uses it at his or her own risk.

1.26 1.5 Author

GUICreator1.5 Author

Address:

Markus Hillenbrand
Hauptstra_e 6
36124 Eichenzell-Lvschenrod
Germany

Phone: -49/6659/3176 (Sa-Su) and -49/631/99658 (Mo-Fr)
(Germany)

EMail: m_hillen@informatik.uni-kl.de
(not during holidays)

Banking:
Raiffeisenbank Eichenzell (BLZ 53060180)
Account # 2103850

This program was developed on an A4000/040.

I'd be grateful for:

- presents
- SCSI hard drives
- SCSI tape drives
- lottery winnings

and of course

- bug reports
- suggestions for improvement
- comments

and most of all

- REGISTRATIONS

1.27 sorry!

Sorry!

My apologies to anyone who tried and failed to make contact at my old email address. At around 18th september my account was suddenly and without warning terminated. Imagine my suprise to find i no longer exist, to find my account ↵
 fini,
 to find this parrot's dead. No longer on this mortal coil, no longer with us, gone to software heaven, frankly needing more than a magnetic medium to get in touch with our dearly departed data :((KO'ed at HQ by HO.

This ment not only the loss of pending email, but records on all contacts made to date. If that wasn't bad enough, i had just, that week, been published on ↵
 the
 CDROM of the CU Amiga magazine (CUCD15), which ment all those potential ↵
 contacts
 are now lost.

Anyway, due to an act of devine intervention, (new JAVA course) i am now back ↵
 on
 line. Existing from day to day, suddenly aware of my own mortality, unsure of ↵
 the
 integrity of the system, but gradually rekindling my trust... bare with me.
 Willing to forgive but never forget!

If you were one of those who tried and failed, ...do call again y'all :*)

Feedback

1.28 thank-you!

Thanks!

Special thanks to the following who had made contact, and are fondly remembered ↵
 :

Groetjes, JAVE. ...pointing out fullstop zone error resulting in
 bad compilation for paragraph/section names less then 4 ↵
 characters
 request for relaxation of level 02-49 zone 'B' done!

...microFocus COBOL85 info? still treading water?)

Charles Elliott ...still beta testing? trace environment directives?

Claus Luethje ...offer of docs, nice to know my prog, from its humble ↔
origins
has found such a grand home. Wow the stuff dreams are ↔
made of !))

i am afraid the rest of you went down with the great crash :((
do call again!

Feedback
