OCic

An Interactive OpenCOBOL Compiler Front End

OCic created and documented by:
Gary Cutler
CutlerGL@gmail.com

Contents

OCic 1

Introduction To OCic 1

OCic Switches 5

OCic Switches and Corresponding COBC Arguments........ 6

The OCic Screen 8

Introduction To OCic

The OCic program is an OpenCOBOL Interactive Compiler “front-end” to the standard OpenCOBOL “cobc” compiler.

YOU DO NOT HAVE TO USE OCic TO COMPILE YOUR PROGRAMS - you remain free to use the standard OpenCOBOL **cobc** compiler command, and even if you do decide to use OC there's nothing preventing you from using **cobc** as well.   An advantage to using OCic to compile your programs is its ability to generate source and/or cross-reference listings of your programs.

Source listings generated by OCic will show the original source code of your programs, with all indentation and comments preserved. Additionally, any COPYed code will be included in the listing immediately following the COPY statement that triggered its inclusion into your program.

Figure 1 shows two pages from a source listing.

Cross-reference listings will show all user-defined data items and procedures as well as intrinsic function and special register references. In addition to showing the line numbers at which items were defined and referenced, those references that MODIFY the contents of the data item will have an asterisk appended to them.

The columns of information found on a cross-reference listing are as follows:

Column	Meaning
PROGRAM-ID	The PROGRAM-ID of the program unit that the data-item reference was found in.
Identifier/Register/Function	The name of the user-defined data name or procedure name, built-in register or intrinsic function that was referenced.
Defn	The source line number where the item was defined in the original input source file. Items defined within a copybook will all have the same “Defn” line number (observe the various “COB-SCR-xxx” items), and that line number will be the source line number where the COPY occurs.
Where Defined	This indicates the area in the program unit where the definition took place. Possible values are:

Column	Meaning	
	CONFIGURATION	CONFIGURATION SECTION of the ENVIRONMENT DIVISION
	FILE	FILE SECTION of the DATA DIVISION
	INPUT-OUTPUT	INPUT-OUTPUT SECTION of the ENVIRONMENT DIVISION
	LINKAGE	LINKAGE SECTION of the DATA DIVISION
	LOCAL-STORAGE	LOCAL-STORAGE SECTION of the DATA DIVISION
	PROCEDURE	PROCEDURE DIVISION
	SCREEN	SCREEN SECTION of the DATA DIVISION
	WORKING-STORAGE	WORKING-STORAGE SECTION of the DATA DIVISION
	[xxxxxxxxxxxxx]	Defined within copybook "xxxxxxxxxxxxx"
References	All source line numbers within the program unit where the item is referenced. If the line number has an asterisk next to it (*), **the item WILL BE MODIFIED at that line!** If the same line number appears multiple times for the same item, that item is referenced multiple times on that line.	

Figure 2 shows a sample page from a cross-reference listing.

Source and/or cross-reference listings will be written to a single file in the same folder in which the program being compiled resides. The filename will be the same as that of the compiled program and the extension will be "**.lst**".

Use of the "**>>SOURCE FORMAT IS FIXED**" and/or "**>>SOURCE FORMAT IS FREE**" directives within program source may cause line-number references to be incorrect in the source or cross-reference listings. This turns out to be caused by the "cobc" compiler's occasional introduction of an extra blank line in the pre-processed intermediate source file (xxxxx.i) when these directives are used. These directives may be used freely within COPYed code, however.

Source to OCic is provided in the **x:\OpenCOBOL\Samples** folder in case you'd like to modify it or if you just want to see how "things are done" in OpenCOBOL programs. The compiled version is available in the "**x:\OpenCOBOL\bin**" folder along with the other executables in the distribution. A full listing of the OCic program is also included in the **OpenCOBOL Programmers Guide**.

Figure 1 - A Sample from an OCIC Source Listing

```
OpenCOBOL V1.1 12MAR2010 Source Listing - OCic Copyright (C) 2009-2010, Gary L. Cutler, GPL                                  2010/04/12
                                                                                                      E:/OpenCOBOL/Samples/STREAMIO.cbl
Line   Statement
====== ===================================================================================================================================
   337      **        assumed.  The filename will be STREAMIO-nnnnnnnn.dat  **
   338      **        where "nnnnnnnn" is a random number.                  **
   339      **                                                              **
   340      ** .      If you specify only a dot (period) as the filename,   **
   341      **        the behavior will be the same as with a value of      **
   342      **        SPACES except there will be no ".dat" at the end of   **
   343      **        the generated filename.                               **
   344      **                                                              **
   345      ** .ext   If you specify a filename extension prefixed with a   **
   346      **        dot (period), the behavior will be the same as if a   **
   347      **        value of SPACES were specified, except that the given **
   348      **        extension will be used instead of ".dat".  Note that  **
   349      **        if you are using a Unix/Cygwin implementation of      **
   350      **        OpenCOBOL and you'd like to specify a hidden file in  **
   351      **        the current directory as the SCB-Filename, you MUST   **
   352      **        code the filename as "./.xxxxx" to avoid having it    **
   353      **        treated as this special name.                         **
   354      **                                                              **
   355      ******************************************************************
   356       ENVIRONMENT DIVISION.
   357       CONFIGURATION SECTION.
   358       REPOSITORY.
   359           FUNCTION ALL INTRINSIC.
   360       DATA DIVISION.
   361       WORKING-STORAGE SECTION.
   362       01  Access-Mode                 PIC X(1) COMP-X.
   363       01  Arg-Length                  PIC X(4) COMP-X.
   364       01  Buffer                      PIC X(256).
   365       01  Delimiter-Buffer            PIC X(2).
   366       01  Env-Temp                    PIC X(256).
   367       01  Slash                       PIC X(1).
   368       01  Tally                       USAGE BINARY-LONG.
   369       01  Temp-9-8                    PIC 9(8).
   370       01  Temp-X-256                  PIC X(256).
   371       LINKAGE SECTION.
   372       COPY STREAMIOcb.
       01 Streamio-CB.
        05 SCB-Handle PIC X(4) COMP-X.
        05 SCB-Mode PIC X(1).
        88 Streamio-MODE-Input VALUE 'I' 'i'.
        88 Streamio-MODE-Output VALUE 'O' 'o'.
        88 Streamio-MODE-Both VALUE 'B' 'b'.
        05 SCB-Function PIC X(2).
        88 Streamio-FUNC-CLOSE VALUE 'C ' 'c '.
        88 Streamio-FUNC-DELETE VALUE 'D ' 'd '.
        88 Streamio-FUNC-OPEN VALUE 'O ' 'o '.
        88 Streamio-FUNC-READ VALUE 'R ' 'r '.
        88 Streamio-FUNC-READ-Delimited VALUE 'RD' 'rd'
        'rD' 'Rd'.
        88 Streamio-FUNC-WRITE VALUE 'W ' 'w '.
        88 Streamio-FUNC-WRITE-Delimited VALUE 'WD' 'wd'
        'wD' 'Wd'.
        05 SCB-Delimiter-Mode PIC X(1).
        88 Streamio-DELIM-Unix VALUE 'U' 'u'.
        88 Streamio-DELIM-Windows VALUE 'W' 'w'.
        05 SCB-Offset PIC X(8) COMP-X.
OpenCOBOL V1.1 12MAR2010 Source Listing - OCic Copyright (C) 2009-2010, Gary L. Cutler, GPL                                  2010/04/12
                                                                                                      E:/OpenCOBOL/Samples/STREAMIO.cbl
Line   Statement
====== ===================================================================================================================================
        05 SCB-Error-Routine USAGE PROGRAM-POINTER.
        05 SCB-Error-Routine-Num REDEFINES SCB-Error-Routine
        USAGE BINARY-LONG.
        05 SCB-Return-Code USAGE BINARY-LONG.
        05 SCB-Filename PIC X(256).
   373       01  Arg2                        PIC X ANY LENGTH.
   374       PROCEDURE DIVISION USING Streamio-CB, Arg2.
   375       010-Main.
   376           ENTRY "streamio" USING Streamio-CB, Arg2.
   377           MOVE 00 TO SCB-Return-Code
   378           EVALUATE TRUE
   379               WHEN Streamio-FUNC-CLOSE
   380                   PERFORM 030-Validate-Handle-NonZero
   381                   PERFORM 200-CLOSE
   382               WHEN Streamio-FUNC-DELETE
   383                   CALL "CBL_DELETE_FILE"
   384                       USING SCB-Filename
   385                   END-CALL
   386               WHEN Streamio-FUNC-OPEN
   387                   PERFORM 020-Validate-Handle-Zero
   388                   PERFORM 100-OPEN
   389               WHEN Streamio-FUNC-READ
   390                   PERFORM 030-Validate-Handle-NonZero
   391                   PERFORM 400-READ
   392               WHEN Streamio-FUNC-READ-Delimited
   393                   PERFORM 030-Validate-Handle-NonZero
   394                   PERFORM 500-READ-Delimited
   395               WHEN Streamio-FUNC-WRITE
   396                   PERFORM 030-Validate-Handle-NonZero
   397                   PERFORM 300-WRITE
   398               WHEN Streamio-FUNC-WRITE-Delimited
   399                   EVALUATE TRUE
   400                       WHEN Streamio-DELIM-Unix
   401                           PERFORM 030-Validate-Handle-NonZero
   402                           PERFORM 300-WRITE
   403                           MOVE 1 TO Arg-Length
   404                           MOVE X"0A" TO Delimiter-Buffer
   405                       WHEN Streamio-DELIM-Windows
   406                           PERFORM 030-Validate-Handle-NonZero
   407                           PERFORM 300-WRITE
   408                           MOVE 2 TO Arg-Length
   409                           MOVE X"0D0A" TO Delimiter-Buffer
   410                      WHEN OTHER
   411                           MOVE -4 TO SCB-Return-Code
   412                           PERFORM 099-ERROR-Return
   413                   END-EVALUATE
   414                   CALL "CBL_WRITE_FILE"
   415                       USING SCB-Handle
   416                             SCB-Offset
   417                             Arg-Length
   418                             0
   419                             Delimiter-Buffer
   420                   END-CALL
   421                   PERFORM 040-Check-WRITE-SCB-Return-Code
   422                   ADD Arg-Length TO SCB-Offset
   423               WHEN OTHER
```

Figure 2 - A Sample Cross-Reference Page

```
OpenCOBOL V1.1 12MAR2010 Cross-Reference Listing - OCic Copyright (C) 2009-2010, Gary L. Cutler, GPL                             2010/04/12
                                                                                                      E:/OpenCOBOL/Samples/STREAMIO.cbl
PROGRAM-ID      Identifier/Register/Function     Defn   Where Defined   References (* = Updated)
=============== ================================ ====== =============== ===============================================================
STREAMIO        010-Main                            375 PROCEDURE
STREAMIO        020-Validate-Handle-Zero            429 PROCEDURE          387
STREAMIO        030-Validate-Handle-NonZero         435 PROCEDURE          380     390     393     396     401     406
STREAMIO        040-Check-WRITE-SCB-Return-Code     441 PROCEDURE          421     578
STREAMIO        050-Check-READ-SCB-Return-Code      452 PROCEDURE          594     610
STREAMIO        060-Identify-TEMP                   463 PROCEDURE          491     506
STREAMIO        099-ERROR-Return                    475 PROCEDURE          412     425     432     438     444     448     455     487
                                                                           535     546     550     561     631
STREAMIO        100-OPEN                            483 PROCEDURE          388
STREAMIO        200-CLOSE                           555 PROCEDURE          381
STREAMIO        300-WRITE                           566 PROCEDURE          397     402     407
STREAMIO        400-READ                            581 PROCEDURE          391
STREAMIO        500-READ-Delimited                  597 PROCEDURE          394
STREAMIO        Access-Mode                         362 WORKING-STORAGE    528*    530*    532*    539
STREAMIO        Arg-Length                          363 WORKING-STORAGE    403*    408*    417     422     570*    574     579     585*
                                                                           586     590     595     601*    602     606     612     615
                                                                           616     618     620*    625     652     657*    657     659*
                                                                           659     662     664
STREAMIO        Arg2                                373 LINKAGE            376     576     586*    592     602*    608     615     616*
                                                                           656     662*    664*
STREAMIO        Buffer                              364 WORKING-STORAGE    622*    627
STREAMIO        Delimiter-Buffer                    365 WORKING-STORAGE    404*    409*    419
STREAMIO        Env-Temp                            366 WORKING-STORAGE    464*    466     468     471*    497     517
STREAMIO        LENGTH                                  LINKAGE            373
STREAMIO        RANDOM                                  PROCEDURE          494     514
STREAMIO        RETURN-CODE                             PROCEDURE          442     446     453     457     544     548     559     570
                                                                           585     601     629     633
STREAMIO        SCB-Delimiter-Mode                  372 [STREAMIOcb   ]
STREAMIO        SCB-Error-Routine                   372 [STREAMIOcb   ]    372     477
STREAMIO        SCB-Error-Routine-Num               372 [STREAMIOcb   ]    476
STREAMIO        SCB-Filename                        372 [STREAMIOcb   ]    384     485     490     492*    503*    505     507     510
                                                                           512*    523*    538
STREAMIO        SCB-Function                        372 [STREAMIOcb   ]
STREAMIO        SCB-Handle                          372 [STREAMIOcb   ]    415     430     436     542     557     564*    572     588
                                                                           604     623
STREAMIO        SCB-Mode                            372 [STREAMIOcb   ]
STREAMIO        SCB-Offset                          372 [STREAMIOcb   ]    416     422*    553*    573     579*    589     595*    605
                                                                           618*    624     640*    646*    654*
STREAMIO        SCB-Return-Code                     372 [STREAMIOcb   ]    377*    411*    424*    431*    437*    443*    447*    450*
                                                                           454*    458*    461*    486*    534*    545*    549*    552*
                                                                           560*    563*    619*    630*
STREAMIO        SECONDS-PAST-MIDNIGHT                   PROCEDURE          494     514
STREAMIO        Slash                               367 WORKING-STORAGE    467*    469*    472*    498     518
STREAMIO        Streamio-CB                         372 [STREAMIOcb   ]    376
STREAMIO        Streamio-DELIM-Unix                 372 [STREAMIOcb   ]    400
STREAMIO        Streamio-DELIM-Windows              372 [STREAMIOcb   ]    405
STREAMIO        Streamio-FUNC-CLOSE                 372 [STREAMIOcb   ]    379
STREAMIO        Streamio-FUNC-DELETE                372 [STREAMIOcb   ]    382
STREAMIO        Streamio-FUNC-OPEN                  372 [STREAMIOcb   ]    386
STREAMIO        Streamio-FUNC-READ                  372 [STREAMIOcb   ]    389
STREAMIO        Streamio-FUNC-READ-Delimited        372 [STREAMIOcb   ]    392
STREAMIO        Streamio-FUNC-WRITE                 372 [STREAMIOcb   ]    395
STREAMIO        Streamio-FUNC-WRITE-Delimited       372 [STREAMIOcb   ]    398
STREAMIO        Streamio-MODE-Both                  372 [STREAMIOcb   ]    484     531
STREAMIO        Streamio-MODE-Input                 372 [STREAMIOcb   ]    484     527
STREAMIO        Streamio-MODE-Output                372 [STREAMIOcb   ]    529
```

If you do decide to use OCic, it will present you with a TUI (Textual User Interface) display with which various compilation options may be selected. When the user presses the ENTER key, a **cobc** command will be generated and executed.

If desired, the user may have selected that the newly-compiled program should be automatically executed upon a successful compilation.

The OCic program makes a perfect means of integrating OpenCOBOL program compilations and test executions into text editing packages such as Helios Software's "Textpad" utility. Of course, it is also suitable for use directly from a command window.

This program's execution syntax is as follows:

ocic <program-path-and-filename> [<switch>...]

Any number of switches may be specified, in any combination of upper- and/or lower-case. If multiple switches are supplied, they must be separated from one another by at least one space. The intent of the command-line switches is to give the user the ability to custom-specify the switch settings YOU want to have as defaults, thus overriding the built-in defaults. While not terribly practical for the user invoking OCic from the command line, this capability is of greater value if you are building an OCic command into a text-editing and/or development framework of some sort, where you only need to enter the "default" switch settings once! Users of OC will quickly see it's easy to change switch settings in OC once it's running, so you don't need to use switches when running OC manually from a console window.

OCic Switches

Most switch names and values can be abbreviated - the valid abbreviations are shown via underlining. For example, the switch "**@DEBUG=YES**" could be abbreviated as "**@D=Y**".

The built-in default switch settings are shown in boldface and are double-underlined (for example, "**NO**" is the default setting for the "**DEBUG**" switch).

Remember that these switches are actually specifying the option selection settings that will be in-effect when the OCic screen is presented (see "**The OCic Screen**").

@CONFIG=BS2000**|**COBOL85**|**COBOL2002**|DEFAULT|**IBM**|**MF**|**MVS

This switch specifies the default **cobc** compiler configuration file to be used

@DEBUG=YES**|NO**

This switch specifies whether (YES) or not (NO) debugging lines (those with a "D" in column 7) will be compiled.

@DLL=YES**|NO**

Use this switch to force ALL compiled programs to be built as DLLs ("**@DLL=YES**"). When main programs are built as DLLs they must be executed using the **cobcrun** utility. When "**@DLL=NO**" is in effect, main programs are generated as actual "**exe**" files and only subprograms will be generated as DLLs.

@EXECUTE=YES**|NO**

This switch specifies whether ("**@EXECUTE=YES**") or not ("**@EXECUTE=NO**") the program will be executed after it is successfully compiled.

@EXTRA=*extra cobc argument(s)*

This switch allows you to specify additional **cobc** arguments that aren't managed by the other OC switches. If used, this must be the last switch specified on the command line, as everything that follows the "=" will be placed on the **cobc** command generated by OC.

@NOTRUNC=YES|NO

This switch specifies whether (YES) or not (NO) the suppression of binary field truncation will occur. If a PIC 99 COMP field (one byte of storage), for example, is given the value 123, it may have its value truncated to 23 when DISPLAYed. Regardless of the NOTRUNC setting, internally the full precision of the field (allowing a maximum value of 255) will be preserved.

Even though truncation – if it does occur – would appear to have a minimal disruption on program operation, it has a significant effect on program run-time speed. The "**Samples**" folder includes two programs – "**bintest**" and "**mathtest**" that can illustrate the truncation and speed aspects of this switch nicely. Try each of them compiled with and without the **cobc** "**-fnotrunc**" argument (a "**@NOTRUNC=YES**" switch on **OC** becomes a "**-fnotrunc**" argument to **cobc**).

@SOURCE = YES | **NO**

If set to YES, this switch controls whether or not a source listing will be generated after a successful compilation.

@TRACE=YES**|NO|**ALL

This switch controls whether or not code will be added to the object program to produce execution-time logic traces. A specification of "**/TRACE=NO**" means no such code will be produced. By specifying "**/TRACE=YES**", code will be generated to display procedure names as they are entered. A "**@TRACE=ALL**" specification will generate not only procedure traces (as "**@TRACE=YES**" would) but also statement-level traces too!

All trace output is written to **STDERR**, so adding a "**2>***file*" to the execution of the program will pipe the trace output to a file. You may find it valuable to add your own DISPLAY statements to the debugging output via "**DISPLAY … UPON SYSERR.**" The **SYSERR** device corresponds to the Windows **STDERR** device and will therefore honor any "**2>***file*" placed at the end of your program's execution. Add a "D" in column 7 and you can control the generation or ignoring of these DISPLAY statements via the "**@DEBUG**" switch.

@XREF= YES | **NO**

If set to YES, this switch controls whether or not a cross-reference listing will be generated after a successful compilation. OCic generates its own cross-reference directly from the intermediate source file produced by the cobc compiler, and does not use any other routines.

OCic Switches and Corresponding COBC Arguments

The following chart shows how the various OCic switches will generate arguments to **cobc**:

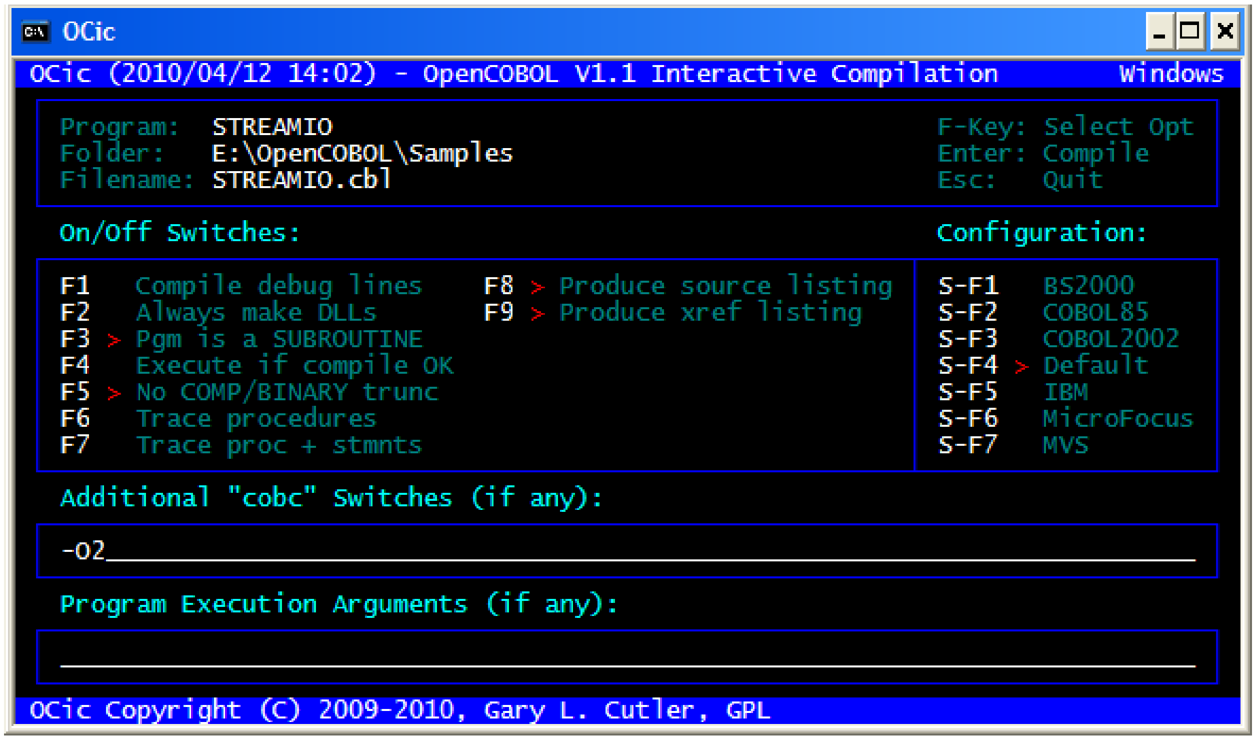
OCic Switch	OCic Switch Value	Corresponding Function Key or Screen Input Area	Corresponding cobc Argument
@CONFIG=*value*	*name*	**shift-F1** thru **shift-F7**	**-conf=%OCPATH%\config\***name***.conf**
@DEBUG= *value*	YES	**F1** (toggles yes/no each time pressed)	**-fdebugging-line**
	NO		None – "D" lines are ignored
@DLL= *value*	YES	**F2** (toggles yes/no each time pressed)	**-m** (used for ALL programs)
	NO		**-m** (used for subroutines) **-x** (used for main programs)
@EXECUTE= *value*	YES	**F4** (toggles yes/no each time pressed) Additionally, program command-line arguments	There is no **cobc** equivalent to this switch – this is a feature meaningful only to OCic
	NO		

OCic Switch	OCic Switch Value	Corresponding Function Key or Screen Input Area	Corresponding cobc Argument
		may be specified in the '**Program Execution Arguments**' area	
@EXTRA= *value*	*extra cobc argument(s)*	specify arguments in '**Additional "cobc" Switches**' area	*extra cobc argument(s)*
@NOTRUNC= *value*	YES	**F5** (toggles yes/no each time pressed)	**-fnotrunc**
	NO		None – binary truncation will be in effect
@TRACE= *value*	YES	**F6** (toggles yes/no each time pressed)	**-ftrace** (will trace just entry to procedures)
	NO	Neither **F6** nor **F7** are "yes"	None – there will be no tracing
	ALL	**F7** (toggles yes/no each time pressed)	**-ftraceall** (will trace entry to procedures and statements)
@SOURCE=*value*	YES	**F8** (toggles yes/no each time pressed)	The cobc compiler will be run <u>twice</u> – once to generate do the actual compilation and (assuming that was successful) a second time to save the intermediate source file (using the cobc "**-E**" option). Note that if both @SOURCE=Y and @XREF=Y are in effect, cobc is still run twice.
	NO		
@XREF=*value*	YES	**F9** (toggles yes/no each time pressed)	The cobc compiler will be run <u>twice</u> – once to generate do the actual compilation and (assuming that was successful) a second time to save the intermediate source file (using the cobc "**-save-temps**" option). Note that if both @SOURCE=Y and @XREF=Y are in effect, cobc is still run twice.
	NO		None – cobc will be run only once

The OCic Screen

The OCic screen will resemble the following:

Figure 3 - The OCic Screen



```
OCic
OCic (2010/04/12 14:02) - OpenCOBOL V1.1 Interactive Compilation        Windows
  Program:   STREAMIO                                         F-Key: Select Opt
  Folder:    E:\OpenCOBOL\Samples                             Enter: Compile
  Filename:  STREAMIO.cbl                                     Esc:   Quit
  On/Off Switches:                                            Configuration:
  F1    Compile debug lines        F8 > Produce source listing  S-F1    BS2000
  F2    Always make DLLs           F9 > Produce xref listing    S-F2    COBOL85
  F3  > Pgm is a SUBROUTINE                                     S-F3    COBOL2002
  F4    Execute if compile OK                                   S-F4  > Default
  F5  > No COMP/BINARY trunc                                    S-F5    IBM
  F6    Trace procedures                                        S-F6    MicroFocus
  F7    Trace proc + stmnts                                     S-F7    MVS
  Additional "cobc" Switches (if any):
  -O2_____________________________________________________________________
  Program Execution Arguments (if any):
  ________________________________________________________________________
OCic Copyright (C) 2009-2010, Gary L. Cutler, GPL
```

You may use the **TAB** key to tab between the "Additional Switches" and "Program Execution Arguments" text-entry fields. Use the function keys named on the screen to control the setting or clearing of various switches or to select the desired compiler configuration. When "set" (equivalent to a "yes" setting of the corresponding command-line switch), a caret (">") will appear between the function key name and the descriptive text on the screen.

Once you're ready, press the **ENTER** key to initiate compilation. You may also quit by pressing either the **F12** or **ESC** keys.

All compiler messages are redirected to a file in your %TEMP% folder named "OC-Messages.txt". This file will be automatically loaded into your system-default text editor (Notepad, Textpad, …) when compilation completes.

If the compilation failed, you'll see the messages generated by the compiler in your text editor. The OCic window will also disappear automatically after a few seconds.

If compilation was successful, a message to that effect will be issued to the OC-Messages.txt file and it will be loaded into your default text editor. Whether or not the OCic window disappears automatically at this point depends on whether you selected the "Execute" switch. If not, the OCic window will disappear.

If your program is to be executed, the appropriate command to do so will be generated and submitted to Windows. This command will be executed in a new window and the OCic window will automatically disappear.

When your program executes, you may find the window dimensions insufficient to properly display the program's output the first time you run it. If that's the case, just select "Properties" from the window's context menu and resize it as desired. If you're using Windows XP, remember to select the "Save properties for future windows with the same title" switch (Vista and Windows 7 do this automatically).