

Installing and Running XXICC

Revision 0.0a © 2011 John F. Beetem.

This work is licensed under the Creative Commons Attribution-ShareAlike 3.0 Unported License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/3.0/>. No warranty is expressed or implied.

This document describes how to install and run XXICC on your computer, either by compiling XXICC source code or installing binaries. These are procedures the author uses, but he may be using a different version of the C compiler or its environment. If you have problems and cannot resolve them yourself, please use the Google Groups mailing list to ask for help.

The current version of XXICC is written in both C and GalaxC. XXICC source code uses the following file types:

- .c ANSI C source code.
- .h C header file.
- .w C source code containing PDF character width information, #included in `pdfwidth.c`.
- .gal GalaxC source code containing ASCII characters and perhaps format control characters as described in *Programming in the GalaxC Language*, section “Format Control”. You may be able to edit .gal files using your favorite text editor. Otherwise, use the XXICC Object Editor (XOE).
- .xoe GalaxC source code containing text and graphics. Edit these files using XOE.

1. Compiling XXICC using Cygwin

The author usually compiles XXICC using the excellent Cygwin (<http://en.wikipedia.org/wiki/Cygwin>, www.cygwin.com) on a machine running Windows 2000 or later. Cygwin provides a GNU/Linux look and feel on a Windows machine. It provides the GNU make and gcc tools for compiling the C modules of XXICC for either Win32 or X Windows, and provides an X Windows implementation for testing X11 executables. It also provides useful GNU tools like grep, diff, and sed.

At the present time, you can compile XXICC for the following targets:

- Microsoft Windows: the resulting executable has been run on Windows 98, 2000, XP, Vista, and 7. It uses the standard Win32 libraries `kernel32.dll`, `gdi32.dll`, and `user32.dll`. For C library functions, XXICC uses either `cygwin1.dll` (if compiled using Cygwin) or `cw32.dll` (if compiled using BorlandC -- see §2). Windows should have all the font files needed by XXICC.
- Cygwin with X Windows: the resulting executable runs Cygwin’s X Windows implementation. This is primarily for testing XXICC’s X11 version of G-SWIM (GalaxC Simplified Windows Manager) and to serve as a model for compiling XXICC for other GNU/Linux environments. The X11 version uses Xft (<http://en.wikipedia.org/wiki/Xft>) to provide FreeType scalable fonts, so `libXft` must be available. For best results, copy the Windows fonts used by XXICC to a directory where Xft can find them. These fonts are `cour*.ttf`, `times*.ttf`, `arial*.ttf`, and `symbol.ttf`. Except for `symbol`, these are all Core Fonts for the Web (http://en.wikipedia.org/wiki/Core_fonts_for_the_Web) which are available for non-commercial use. They are all standard PDF fonts and should be present on any system that has a PDF reader.
- BeagleBoard (<http://beagleboard.org>) under the Ångström GNU/Linux distribution: this version demonstrates XXICC on an ARM processor. It also uses X11 and Xft. For best results, use the

Windows fonts. You may need to purchase `symbol.ttf`, but the Core Fonts for the Web are available for non-commercial use.

BeagleBoard is compiled using CodeSourcery's gratis version of `gcc`.

Here is the procedure for compiling XXICC under Cygwin on a Windows machine.

1. Install Cygwin from www.cygwin.com.
2. Create a directory for XXICC, which we'll refer to as `XXICC`.
3. Copy the desired release file -- e.g., `code00.zip` -- to `XXICC` and expand it.
4. Give the command "`make T=xxx`" in the `XXICC` directory. This will compile XXICC for target `xxx`, which can be:
 - `T=Win32`: compile XXICC for Win32, using `cygwin1.dll` for C library functions. The resulting executable `xxicw.exe` can be run from a Cygwin window or from anywhere else in Windows provided that `PATH` can find `cygwin1.dll`. You must create subdirectory `XXICC/cygw32.od` for object modules.
 - `T=CygX11`: compile XXICC for Cygwin's X Windows. The resulting executable `xxicx.exe` can be run from a Cygwin `xterm` window. You must create subdirectory `XXICC/cygx11.od` for object modules.
 - `T=Beagle`: cross-compile XXICC for the ARM-based BeagleBoard under Ångström GNU/Linux. The resulting executable `xxibea` can be copied to a BeagleBoard and run under Ångström. You must create a subdirectory `XXICC/beagle.od` for object modules. You will probably need to make minor changes to `Makefile` to tell `gcc` where to look for `.so` files.

"`make T=xxx`" compiles all the C modules and links them into one of the above executables. We have tried to make `Makefile` as clear as possible so that you can adapt it to other environments.

5. You now have an executable ready to compile GalaxC code. The next step is to compile the XXICC Object Editor in `XXICC` so that you can edit and compile your own GalaxC programs. Do this using the command for your environment:

<code>xxicw -force xoe</code>	Execute in a Windows command window.
<code>xxicx -force xoe</code>	Execute in a Cygwin or other X11 <code>xterm</code> window.
<code>xxibea -force xoe</code>	Execute in <code>XXICC</code> directory on your BeagleBoard in an <code>xterm</code> window.

XXICC quickly compiles and links all the `.gal` and `.xoe` files needed for XOE. They are listed in `main.c`. GalaxC object code files have a `.gi` extension.

When XXICC is finished compiling, you can open a XOE window and create or edit GalaxC programs or other documents. On Windows, press `ctrl-O` in the Output Window to get a file selection dialog box. On X11, XXICC will automatically open the dialog.

Once the `.gi` files are created, you can run XXICC with the `xxicw`, `xxicx`, or `xxibea` command by itself. Currently you must run XXICC in the directory that contains the `.gi` files.

2. Compiling XXICC using Borland C

You can also compile XXICC using Borland C. This uses Borland make file `xxibor.mak` instead of `Makefile`, since the syntax is somewhat different.

1. Create a directory for XXICC, which we'll refer to as XXICC.
2. Copy the desired release file -- e.g., `code00.zip` -- to XXICC and expand it.
3. Give the command "`make -f xxibor.mak`" in the XXICC directory, assuming Borland C executables are found in `PATH` before any other version of `make.exe`. This will compile XXICC into target `xxibor.exe` for Win32, using Borland's `cw32.dll` for C library functions. `xxibor.exe` can be run from anywhere Windows provided that `PATH` can find `cw32.dll`. You must create a subdirectory `XXICC/borland.od` for object modules.
4. You now have an executable ready to compile GalaxC code. The next step is to compile the XXICC Object Editor in XXICC so that you can edit and compile your own GalaxC programs. Use the command "`xxibor -force xoe`". XXICC quickly compiles and links all the `.gal` and `.xoe` files needed for XOE. They are listed in `main.c`. GalaxC object code files have a `.gi` extension.

When XXICC is finished compiling, you can open a XOE window and create or edit GalaxC programs or other documents. Press `ctl-O` in the Output Window to get a file selection dialog box.

Once the `.gi` files are created, you can run XXICC with the `xxibor` command by itself. Currently you must run XXICC in the directory that contains the `.gi` files.

3. Installing XXICC in Binary Form

One of the huge advantages of FLOSS is that you can see the source code and compile it with your own compiler, minimizing the possibility that you are bringing malware into your system. However, as each compiler has its own quirks, it can be time-consuming to do this. So we may provide pre-compiled versions of `xxicw`, `xxibea`, and others as appropriate.

XXICC executables rely on library files, as described in §1. Some of these are already present in your operating system, e.g., all tested versions of Windows have `kernel32.dll`. Other binaries such as Xft may not be present by default and you'll have to get them from your OS distributor or elsewhere on the Internet. GPL does not allow redistribution of binaries without their source code, so we don't include library binaries.

If you want to run `xxicw.exe`, you'll need `cygwin1.dll`. You can get this by installing Cygwin on your machine using the instructions at www.cygwin.com. If you don't want to install Cygwin, you can extract just `cygwin1.dll` from a release file at one of the many HTTP mirror sites listed at <http://cygwin.com/mirrors.html>. Look in `release/cygwin` (e.g., <http://mirrors.kernel.org/sourceware/cygwin/release/cygwin>) for a file like `cygwin-1.7.9-1.tar.bz2`. This is release 1.7.9 in the form of a compressed `.tar` (tape archive) file. `cygwin1.dll` is buried in this file in subdirectory `/usr/bin`. You can extract just this file using `tar` or 7-Zip and put `cygwin1.dll` at some convenient place where `PATH` can find it, or just put it in your XXICC directory.

Here is the procedure for installing XXICC in binary form:

1. Create a directory for XXICC, which we'll refer to it as XXICC.
2. Copy the desired release file -- e.g., `code00.zip` -- to XXICC and expand it. Even though you don't need the `.c` and `.h` files, you do need the `.gal` and `.xoe` files.
3. Copy the desired XXICC binary executable -- e.g., `xxicw00.zip` -- to XXICC and expand it.
4. Make sure you have the library files (`cygwin1.dll`, `libXft.so`) and fonts you need for your executable.
5. Run your executable to compile the XXICC Object Editor as in §1 step 5 or §2 step 4.

This document will be updated to handle other environments such as other versions of GNU/Linux. At some point it may also make sense to use the package installation procedures offered by some distributions, but currently our procedures are so simple that it hardly seems worth the bother. At least with our procedures you know what's going on and have a clue as to how to fix problems if they occur.

4. Running XXICC

We have already examples of running XXICC, both for compiling XOE and for running XOE. This section describes additional options. The general form for running XXICC is:

`xxicc options target`

Where `xxicc` is one of the XXICC executables listed in Sections 1 and 2.

Here are the possible *options*, which can be used in most combinations. They are separated by spaces:

`-llib.dll`

`-llib.so`

Include a Windows Dynamic Load Library `lib.dll` or its GNU/Linux equivalent Share Object library `lib.so` when XXICC starts up. XXICC assumes that any `PATH` or equivalent environment variable has been set up correctly to find the library. The default libraries needed for Windows or X11 have already been included and do not need to be listed on the `xxicc` command line.

XXICC does not need to know about `lib` when it is compiled and linked at the C level. It loads `lib` and looks up function names during GalaxC compilation as it compiles `cdecl` and `stdcall` inline function calls. In fact, it does not check whether `lib` even exists until it compiles `inlines`. This is truly dynamic linking.

`-force`

This forces recompilation of all GalaxC code that is needed to compile or load *target*. Otherwise XXICC uses a Make procedure by checking source and object file time stamps. For details, see *Compiling GalaxC Programs*.

`-nocall`

`-nogm`

These are special Make options primarily for the Borland C version. For details, see *Compiling GalaxC Programs*.

File *target* is a source or object file that XXICC should either compile or load. If it has a `.gi` extension,

XXICC loads it, automatically performing Make. Otherwise, XXICC assumes it's a ".gal" or ".xoe" source file and compiles it, stopping after each stage for the user to press RET or ENTER. This was used early on when debugging the compiler and is not generally useful.

If target is xoe, XXICC recompiles all of XOE using a Make procedure. The `-force` option forces recompilation of all XOE files instead of just checking time stamps. It is equivalent to deleting all the .gi files.

If there is no target, XXICC loads and runs XOE without checking any time stamps. Use this if you haven't changed XOE.