Notes on Rebuilding XFree86/OS2 from Scratch

Holger Veit

Last modified March 8th, 2000

1. Preface

X11 and XFree86 were initially developed on Unix-based systems. Usually Unix systems provide a rich number of tools and utilities to get certain things done. Under OS/2, these tools are not installed, but ports are available which are sometimes functionally equivalent to Unix utilities with the same name, but also differ sometimes in a subtle way. This guide will give you hints if you intend to rebuild the system from scratch under OS/2.

Please also read README.OS2 for end-user information, and set at least the environment variables described there.

At the current time, the most recent version available is XFree86-4.0. This is a full and unrestricted version which comes with complete source code. 4.0 is a highly experimental release, so many features that might have worked in earlier versions, may now no longer work, or work differently. Be aware that for OS/2, XFree86-4.0 is considered to be alpha software.

If you want to join the XFree86 developer team, e.g. to add support for certain hardware, please send a request to <BOD@XFree86.org>. Please think about such a step carefully before, though, since much work is involved. Please use the XFree86-4.0 source code as a test example how to compile the system. The ability to manage that is a basic requirement for becoming a developer.

2. Tools required

I have tried to reduce the number of external tools, but when looking back it seems I were not very successful. At least I managed to get everything working with the native CMD.EXE shell only. However, there is still plenty of software required. Most of this software is available from `hobbes.nmsu.edu` or `ftp.leo.org` via anonymous FTP. The following shopping list shows what you will need:

- gcc EMX/gcc emx 0.9C patch4 or later (0.9d preferred!)
- gzip GNU zip/unzip
- tar GNU tar
- patch Larry Wall's patch utility (attention: incompatible tool with same name in OS/2)
- install BSD/GNU install
- rm,mv,cp GNU file utilities
- tee,.. GNU shell utilities

- groff GNU nroff/troff
- sed GNU sed stream editor
- grep GNU grep
- gawk GNU awk
- make GNU make 3.71/3.72 (use the one from Xprog.zip!)
- flex GNU flex
- bison GNU bison
- m4 GNU m4
- find GNU find (attention: incompatible tool with the same name in OS/2)

If there is no version number given, any new version will do. Particularly critical is only EMX/gcc and GNU make. Note that the second GCC implementation which might still be available from some archives is NOT compatible.

Furthermore, you need the XFree86 sources. These are available from the common XFree86 repositories. Look into a directory which is often named /pub/XFree86/4.0/source.

3. Compiling and Installing

You need about 300MB of free HPFS space for the whole system. This does not include space for the postscript and troff documentation files. I have never installed them. Nor did I install the test subtree.

1. Install all the above utilities. Refer to the corresponding documentation. Verify that everything works well, particularly EMX.
2. It is a good idea to use the same or a similar structure I have. I have made a directory `\x11` on the partition for compiling and have put everything below this tree. I found that a clean tree occupies less than the half space of the disk, this gives me the opportunity to rename this tree to `\x11old` and copy a new version to the same disk to produce diffs. Last time the complete tree was arranged under the root directory `xc`, this would become `\x11\xc` then.
3. To unpack the files you would usually execute the command

   ```
   gzip -dc file.tar.gz | tar xvf -
   ```

   in the `\x11` directory. At the end you will usually see the irritating, but non-fatal message "gzip: stdout Broken pipe". Ignore it.
4. After that, is is likely necessary to apply some patches, either from the XConsortium or from the XFree86 project. Before you do this, enter

   ```
   chmod -R a+rw \x11\xc
   ```

   to make certain files in the tree writable.
5. There should be a file `added-XXX` accompanying the patch file which lists the files that are newly created. The patch program has a problem with creating new directories, so we need to create them on advance. For each `added-XXX` file you find, execute from `\x11`

   ```
   xc\config\util\added added-XXX
   ```

   If there is no `added-XXX` file available, you can make one with the following instructions:

```
grep "\*\*\* xc/" patchfile >added-file
```

Edit `added-file` with a text editor and remove the `***` at the beginning and the time stamp at the end (search for a TAB and erase to the end of the line). You get a list of file paths, one in a line, which is the input to the added utility.

6. After that you can apply the patches in the right order. Usually this is done by a command

   ```
   patch -p -E <patchfile 2>&1 | tee patchlog
   ```

   from the `\x11` directory. Be aware to use the right patch - OS/2 has a utility with the same name and different functionality. Don't use the recommended `-s` option, this makes `patch` quiet, and you won't see problems in the patchlog file. Use

   ```
   find \x11 -name *.rej -print
   find \x11 -name *# -print
   ```

   to find any rejects and unapplied patches (attention: yet another OS/2 program with wrong functionality). Normally there shouldn't be any problems of this kind, else you have made a mistake. Finally remove the original files with

   ```
   find \x11 -name *.orig -print -exec rm {} ;
   ```

7. Go to the `xc/config/cf` directory and edit the `xf86site.def` file to match your requirements (you probably don't want to compile all X servers). Certain changes must be set to the following values:
   - Disable if not already done any PC98 server; PC98 (Japanese XFree86) does not work yet. Porters from Japan are welcome!
   - `#define WacomSupport NO #define ElographicsSupport NO` Both options are not yet supported.
   - Tcl* and Tk* don't need to be set explicitly. Reasonable defaults are in the other config files, provided you have a complete XFree86/OS2 binary tree with the tcl/tk runtime support installed.
   - `#define BuildDynamicLoading NO` This does not work.
8. Go to the directory `xc\util\compress` and `make compress.exe` there. Install the program produced there in your path. I stumbled more than once on half-ported compress programs on OS/2 ftp servers that are defective w.r.t. reading and writing stdin/stdout. In some stage (font compression) otherwise you will get a core dump of mkfontdir, because all compressed fonts are corrupt.
9. Set the environment variable `X11ROOT` to something different than it is; otherwise the installation process will overwrite your original XFree86/OS2 installation. If you have not set this variable, go back to the prefix section of this document: you have forgotten something.
10. Copy the file `xc/programs/Xserver/hw/xfree86/etc/bindist/OS2/host.def.os2` to the location `xc/config/cf/host.def`. Use this file to do any specific modifications to imake variables, rather than editing the file xfree86.cf, imake.tmpl, or os2.cf directly.

11. Copy the file `xc/config/util/buildos2.cmd` into the `xc` directory. If this is a second or later attempt, you might need to copy the saved toplevel Makefile.os2 back to Makefile.
12. Execute this `buildos2.cmd` command in the `xc` directory; it will produce a logfile `buildxc.log` in this directory.
13. Go have a bucket of coffee, or better, buy new coffee - in Colombia! The compile will need between 2 and 20 hours, depending on your selections, and the horse power of your hardware.
14. When finished, view the logfile for errors, and fix the problems if there are some. I have managed to compile the whole system flawlessly, so there is at least one configuration that works.
15. Finally, from the `xc` dir, execute

```
xmake install
xmake install.man
```

16. There are a few minor glitches in the installation:
    1. The xdm and linkkit directories will fail in compile and installation. This is no problem and has no effect on the rest of the system.
    2. The imake.exe which is installed in `\XFree86\bin` is usually defective. The one which was built initially and installed in the root directory of the drive where you have the source tree is okay. So simply copy this `\imake.exe` to the `\XFree86\bin` directory manually. Some day this might be fixed.
    3. `XF86Setup` is not ported yet and won't work with the tcl/tk port available for XFree86/OS2. My idea was to replace this by some native installation tool, which I didn't find the time to do yet. Feel free to spend a bit of time to play with XF86Setup if you like.

Well, you see, this was quite easy :-)

CONTENTS

$XFree86: xc/programs/Xserver/hw/xfree86/doc/sgml/OS2Notes.sgml,v 1.2 2003/01/04 04:20:23 dawes Exp $