

# Bijlage 18

## The DJGPP port of web2c

Gilbert van den Dobbelsteen  
gilbert@login.iaf.nl

### abstract

Web2c is getting popular and popular. Why? Because it is widespread. I'll discuss some of the aspects of the port to the MS-DOS environment. I used the variant on the 4AllT<sub>E</sub>X4 CD-ROM.

### Introduction

So what's so good about web2c<sup>1</sup>? Since I didn't know, I decided to find out. I know nothing about web2c — I still don't — except its wide-spread use on Linux systems and other unix based operating systems. The system is ported to win32 (win95 and NT) and various other platforms. There is also a win95/NT and a DOS port. This article focusses mainly on the DOS port.

Since I use emT<sub>E</sub>X, I was interested in getting a combination of things. This involved the usual T<sub>E</sub>X programs from the web2c distribution, and the dviscr viewer from emT<sub>E</sub>X. I also needed dvihplj because it's a good program which I know.

### What is DJGPP?

DJGPP is a port of the GNU C/C++ compiler and development tools to the 32-bit protected mode environment on Intel 32-bit CPUs running MS-DOS and compatible operating systems. The port was made by D.J. Delorie and friends. Early versions of the compiler needed extenders to run the generated executables. With the coming of version 2.0 DJGPP programs do not need a separate extender program, only a DPMServer to run. DJGPP includes a free 32-bit DPMServer so you don't have to worry about that.

### Extenders and DPMServer

So now you say 'what is an extender?' I can imagine that question. Traditionally DOS (in specific MS-DOS) is a 16 bit operating system<sup>2</sup> with no 32-bit support at all. So how can you run 32-bit programs? Some years ago a number of companies made a standard called DPMServer. It stands for Dos Protected Mode Interface. Under DOS a 32-bit intel CPU (like the 386, 486, pentium) runs in 8086 mode. Ok, it's a lot faster than a 8086 but it is still limited in some aspects.

To gain the advantages of the modern processors, you (I mean the software guru's here) can switch the processor into something called 'protected mode'<sup>3</sup>. Protected mode has several advantages. To name a few:

- You can run more programs simultaneous. Windows 3.x does that. Each DOS box you open is a separate process.
- Protection. Programs cannot interfere with each other. This subject needs some attention. Implementation of this protection is usually different for different operating systems. For example a DOS box in Windows 95 can easily crash the system, where on the other hand a similar DOS box on a OS/2 platform does not. OS/2 is somewhat superior in this aspect. Linux and other unix variants are far superior in crash protection. It's been told that Windows NT is far better in this but I don't tend to trust Microsoft on such advertising stuff.
- There is something called virtual 8086 mode. This is where the DOS boxes are. In this mode the CPU emulates a 8086 and re-routes specific functions to protected mode functions residing in the 32-bit kernel.
- Simple DOS boxes and multi-megabyte applications can co-exist. On Windows 95 this is very obvious, you can have DOS boxes, 32 bit applications and Windows 3.x applications all at the same time. Each program has its separate program space.

There are also disadvantages:

- Communication between tasks is difficult. The microsoft way is to use DDE/OLE but have you actually successfully used that? Modern programs use ActiveX and COM. I won't discuss these.
- Slower. A multitasking system is usually slower because more things need to be administrated. The operating system simply has to do more administration and must protect itself if more tasks make simultaneous system calls.
- It's harder for the software guys to create programs for these systems. Especially if these programs must interact with each other.

1. web2c: pronounce as web-to-see.

2. Some say it's hardly an operating system, but merely a shell with some tools.

3. Windows 3.x calls this Enhanced mode

## Installing web2c

So how to install the stuff? I assume you already have a working emTeX installation.

### kpathsea

What? I don't know where this stands for but this is the tool all programs in the distribution use to find files. As you probably know TeX is a small program but you usually need an endless amount of files stored in another endless amount of subdirectories. Installation and maintainance is a difficult job and kpathsea gives you the option to store files in as many directories as you like. So you decide where to put the stuff and kpathsea finds out where it is.

What does it do? kpathsea maintains a database of files present in your TeX system. This database is not automatically maintained. You can decide not to use the database functionality but for the best, use it.

Each time you install new files you must update the database. There is a command for doing so `MakeTeXls-R`.

There is also the `kpsewhich` command which tries to find a file in the database. If you type:

```
kpsewhich cmr10.tfm
```

the program responds with:

```
c:/tex/share/texmf/fonts/tfm/public/cm/cmr10.tfm
```

If it doesn't, something is wrong and you should check the `texmf.cnf` file which is located in the `c:/tex/share/texmf/web2c` directory.

Change settings in this file as you see fit and experiment. I only changed `prefix` to `$SELFAUTODIR`.

### Merging with the emTeX stuff

To merge in the emTeX stuff you have three options:

1. Merge some of the emTeX files into the new directory tree.
2. Maintain a separate emTeX tree.
3. Mingle with configuration files until you get it right.

I wouldn't advise the first method unless you have configured emTeX lots of times and know what the ins and outs are.

If you choose to do the second, there won't be much of a problem. Beware of this though: you shouldn't have any TeX related environment variables like `%texinputs%`. This confuses the web2c stuff. So make sure your installation is standard and doesn't use environment settings<sup>4</sup>. Disadvantage of this solution: You'll need more disk-space for PK fonts and TFM files since both distributions need them. This means all TFM and PK files must be present twice: once

for the emTeX tree, and one for the web2c tree. Though this is not a problem, suppose you got a new version of a font and you only updated the emTeX version. Things look fine on the viewer, but as soon as you use dvips from web2c to create postscript the old font shows up.

I chose the third method. I left all the files where they are, deleted the unnecessary ones and changed the `texmf.cnf` file so it also looks into the emTeX tree for information.

### Changing texmf.cnf for emTeX

Apply the following changes:

1. Delete all TFM-files from the web2c tree.
2. Move all TFM-files from your emtex distribution into the `/emtex/tfm` directory. I'll explain this later. Make sure all TFM files are in a *single* directory.
3. Change `texmf.cnf`. Change the variable `TFM FONTS` and `PK FONTS`. You can use the `dvidrvfonts` environment variable for the latter.
4. Run `MakeTeXls-R` to update the file database.
5. rename the `xxxDPI` directories in the `PK FONTS` to `DPIxxx`.
6. You probably need to change some mfjob parameter files as well. Get into the `/emtex/mfjob` directory and edit `modes.mfj`. Look for the lines at the end. They say something like `def output_lj= ...`. Change the `@Rdpi` into `dpi@R`. This is how web2c likes to have the PK files.
7. Rename the directory `pixel.lj` to the mode you are using (e.g. `laserjet` or something like that). Modify the `.CNF` file for `dvips` to use this directory too.

Item 2 says you should move all you TFM files into a single directory. This is because web2c maintains the same directory structure when generating PK fonts. So if `cmr10.tfm` is in a subdirectory (`cm`), the PK file will also be in that subdirectory. This is different from emTeX.

The idea of having subdirectories is handy, but is not compatible between web2c and emTeX. In emTeX you can use the font `cm/cmr10` by saying:

```
\font\f=cm/cmr10
```

but emTeX has no way of making a difference when using searching for `cmr10.pk`. In web2c this TeX feature doesn't work<sup>5</sup>.

4. This is not an easy task.

5. This is very unlucky because in emTeX you could easily maintain different encodings of the same font, and still use the same file name. This is an aspect where TeX is not very portable across systems, a missed opportunity.

You probably want to remove the emTeX metafont program. It is best to use only one metafont program. mfjob can call the METAFONT program that comes with web2c, but it's a bit tricky. Experiment here (clear out the mfjobopt in your environment).

Now start checking things out. Make sure TeX can find the font files. Simply run `testfont.tex` through TeX and try to create a font-sample from `cmsy10` or something like that. If this checks out fine, you have to check the viewer. If the Mattes viewer works, try to delete a PK file that is used (you can find this in `dviscr.dlg`). When `dviscr` says *Run mfjob to generate the font?* type Y, and check if the font will be generated and positioned in the proper directory.

Don't forget: Your `mode.defs` in METAFONT must be the same for emTeX and for web2c.

Also check if dvips (from web2c) generates the correct PK fonts. Make sure the files end up in the correct directory. After a few hours of experimenting you'll have the TeX system you want, and you've learned a lot about configuring web2c.

## Advantages of web2c

Here are some advantages of web2c:

- If you use web2c you will notice it is *extremely* flexible in configuration. Furthermore most programs related to TeX are available in web2c.
- One of the great advantages is the configuration. Although the `texmf.cnf` configuration file is not very easy to understand, once you master it you'll see the flexibility and power. On my old system (using emTeX) I had several environment variables and several configuration files depending on the person's name using TeX, are we running under windows, and so on. With web2c I can have a single installation on the network and provide stuff for all network users. Even the format files (`latex.fmt` and so on) are portable across platforms. So you can use a single installation and serve all users independent of the platform, whether it is DOS, win95/NT or even Linux. And if there's a port to the MAC, I wouldn't be surprised if it worked right away.
- When you are finally through configuring all the stuff (this takes an endless amount of time), the updating is pretty easy. If you get a new executable<sup>6</sup> you can overwrite the older version with the new one, re-generate the formats and you're done.
- Your users can take advantages of different platforms. The win95 version supports long filenames where the DOS version does not.
- Cross platform stuff. The format files are usually cross platform. An example: Suppose you have a file-server with a web2c installation. You only have to generate the

format files once and then they can be used on win95, MS-DOS and linux. There are no separate format files needed for other platforms. Even endian conversion is done by web2c. This is real cool and very maintainable. Since you know you need only one format file you won't have different behaviour on different systems.

- And there is also pdfTeX. This is a TeX variant that has pdf files as output. So if you use TeX mainly to create pdf files (as I do) then this is a time-saver. You don't need to use dvips and distiller anymore. There is a drawback to pdfTeX: the graphics support is limited. You can't include fancy EPS files. You must convert them using GhostScript.
- The DOS version of web2c has another nice advantage. I always had problems running huge emTeX in a DOS box under Windows. The DOS port of web2c runs under DOS, win3.x, win95 and OS/2. It even runs under DESQview.
- TeX input files. In web2c you can specify which paths are used for `\input`. You can say for example: 'Hey TeX if you are using the latex format, look only in these directories for input files'. And for another format you can use a different set of directories. This comes in very handy if you have different formats each with their own set of input files. You don't want you L<sup>A</sup>TeX 2<sub>ε</sub> input files to mix up with input files of another package.
- Custom options. Example: Copy the program `tex.exe` in the bin directory to `latex.exe`. When you start the `latex.exe` it automatically loads the format `latex.fmt`. Ok, that's nice but you can do more than that. in the `texmf.cnf` configuration file you can say: `main_memory.latex = 500000`, which means: 'Dear TeX if the program name is latex.exe make sure there are 500000 main memory words available. This way you can customize various settings for different variants of the same TeX compiler.

## Drawbacks of web2c

Are there any? Yes. First of all it is a huge package. The full installation requires some 20 megabytes (or more) and you have to administrate some things. Second, the configuration. Since web2c embodies the spirit of TeX, it is highly flexible. This results in lots of errors being made when configuring the system. Every distribution I used (win32, DOS and linux) didn't work right out of the box. There's always some variable wrong.

Currently there is no viewer for web2c. The viewer supplied is a quick hack and lacks lots of options. The best viewer for DOS is still `dviscr`, and it doesn't come easy

6. Although unlikely, suppose DEK found a bug in TeX and makes an update. It's more likely that a bug is fixed in the web2c distribution

to get this installed properly. The port for win95 includes xdvi. This is a good previewer and moreover it is well integrated into the web2c environment. At the time of this writing the port should be available as beta.

## Conclusion

web2c is a decent  $\text{T}_{\text{E}}\text{X}$  implementation and it is rather complete.

Installation and configuration is not easy, especially if you intend to use  $\text{emT}_{\text{E}}\text{X}$  and web2c together. The way web2c uses directories is too complex for  $\text{emT}_{\text{E}}\text{X}$  to handle. You must simplify things to them working.

I didn't talk about multiple directory trees, but it is possible to have more than one configuration file and let web2c use them simultaneously. Find some docs on that, there are plenty. As usual with  $\text{T}_{\text{E}}\text{X}$  products, configuration and customizing things is not easy and takes time.