

NeXT stop, the Web

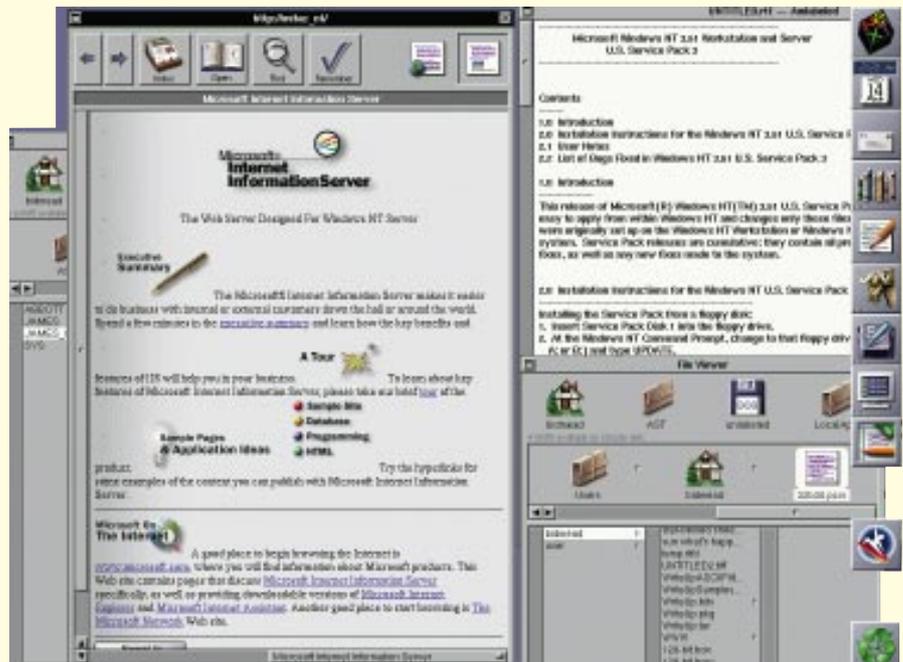
Chris Bidmead on the shortcomings of Warp's Internet Access Kit, the demise of Taligent, and why .rtfd is the way forward.

Last month I was whingeing about some of the shortcomings of IBM's Web Explorer. My opinion of the Warp Internet Access Kit has gone down a couple more notches since then, although the blame for this lies not with IBM, but with NeXT and the new PEP machine I told you about. Messing around with Internet-connected NeXTStep has opened my eyes to how this World Wide Web stuff ought to work in a properly designed 32-bit computing environment.

As I said last month, there are several browsers on the NeXT machine. The one I've been using mostly is Netsurfer. It has some very important features that you won't find with WebExplorer, which up to now, despite my criticisms, has been my favourite way of getting onto the World Wide Web. I've already mentioned the pain of not being able to highlight a selection of text from the browsers and copy it across to some other application. There are other annoyances:

1. You have to wait for a page to fill (or explicitly stop it filling) before you can spring off from one of its links on to the next. Given that at least 50 percent of the pages you visit are only stepping-stones, you waste a lot of mouse clicks aborting pages in mid-fill by clicking on the animated icon at the top left-hand corner, then clicking again on the link you want to jump off. Just clicking on a link should be enough to tell the browser that it's time to move on.

2. When you move on to a new link, you don't necessarily want it to overwrite the old one. The page you're about to step off from might have several links you want to explore. Why won't WebExplorer let you



click-click-click on three different links and spawn pages for each of those links, retaining the original page? OS/2 is a multi-tasking, multithreaded operating system underneath, so why can't we see some of this carried through into the application?

3. How do you save a page that you want to refer to again? Yes, WebExplorer lets you file it, but when you bring it back offline all you get is the raw HTML-formatted text. The graphics have all fallen off. For months now, I've just been printing pages that I wanted to keep intact. But not everybody has a colour printer to hand, and those who do may not want to print pages that may only be read once and then binned.

None of this is a problem on the PEP NeXT box with Netsurfer. For a start, clicking on a link jumps you to the target page whether the current page has finished filling or not. Simple — NetScape does this too. But look how Netsurfer behaves when you click on a link and hold down the NeXT

Netsurfer on the Next machine reads the Windows NT Web server across my "intranet". Sharp-eyed readers will notice the word "Microsoftfi", which should read "Microsoft®". The "fi" is a single-ligatured character in the NeXTStep character set where Microsoft puts ®, one of the minor problems of multi-platform computing

Command key (usually Alt-Left, but mappable). It holds on to your current page (and goes on filling it if it's in mid-fill), and spawns a new page for the link you're springing off to.

Copying and pasting a section of text from the surface of Netsurfer works the way it does in NetScape and Microsoft Explorer. A little disappointingly, the NeXTStep interface shows its age here. By modern lights you ought to be able to highlight a section of text and drag it across to an editor; instead you have to copy and paste explicitly with the standard Command-c Command-v keystrokes or by

pulling down the Netsurfer menu and selecting Edit/Copy. A very similar menu appears in all NeXTStep applications, by default at the top left of the screen, but power users arrange the application menu to be invisible, only popping up when you click the right-hand button.

So that's copy and paste — don't forget that this NeXT stuff was designed ten years ago. That's why it came as a real surprise when I used one of the NeXT standard mechanisms to save my first Netsurfer page. Needless to say, I hadn't bothered reading the online help, but I knew from messing around with NeXTStep that when you see an icon in a "well", you can usually expect to do some drag and drop with it. An icon well is a square frame, shadowed to look like a shallow indentation in the surface. If you look at my screenshot of Netsurfer (opposite), you'll notice that while the main icons running from left of screen are on raised buttons, the icon at the far right is sitting in a well. The icon shows a page with text and a picture, and represents the whole document you're looking at — technically it's called a "proxy". So one way of saving a page is simply to drag the proxy into a folder shown by the viewer.

And here's the surprise. NeXT works with three basic text types: plain ASCII, Rich Text Format and PostScript, respectively named .txt, .rtf and .ps. Maybe I hadn't been paying attention, but the drag-and-drop save from the icon well turned out to produce a file type that was new to me, an .rtfd file. When I recalled one of these files into NeXT's standard editor, Edit, I was amazed to see that it came back complete, with the text, pictures and diagrams all perfectly in place.

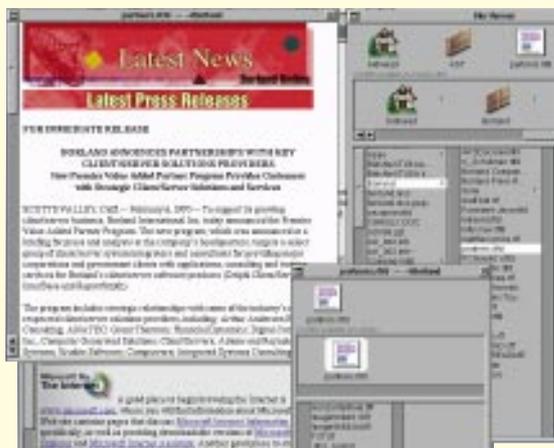
The amazement, of course, was a function of my previous experience with other Web browsers. Any ordinary Joe sitting

down at a computer would expect no less. But how does the decade-old NeXTStep interface do this? If it were Microsoft, it would be using OLE 2.0 Structured Storage. OpenDoc — supposedly arriving this year — will have a similar compound file storage mechanism called Bento, named after the multiple-compartment dishes used to serve up Japanese food. Was NeXT really this much ahead of the game?

The joy of .rtfd

Forget OLE 2.0 Structured Storage. An .rtfd file is so simple it makes me weep (with joy that people still remember how to keep it simple). There's a neat trick that NeXTStep uses to shape its applications. An application tends to be a whole mess of different executables, resource and data files, so to keep them all neatly together, NeXTStep stores them all under a single directory, then makes that directory behave as through it were the sole executable. All you see from the NeXTStep desktop is a single icon, on which you click to launch the app. The icon actually belongs to the directory, but you never get to see the underlying files unless you explicitly use the Workspace menu's File/Open As Folder command.

This is something you can also do with OS/2, as we discussed in this column once. And it's the same way that the Netsurfer saved pages retain their pictures. The .rtfd file is in fact a directory (hence the d), iconised to handle like a file. Inside the directory is the main rich text file, and this holds pointers to the picture files, which are also contained in the directory. Drag the page from the browser into a folder, and the .rtfd file is created on the spot. Double-click on the .rtfd file and it loads into the Edit app, NeXTStep's standard editor, and is displayed like the original browser page. 



Borland's WWW versions of its press releases are far more attractive than the drab typed pages that arrive through the post. NeXTStep saves them as .rtfd "files", but as the Viewer shows here, they're really directories that hold the pictures and text necessary to rebuild the original page. Digital Librarian, NeXTStep's standard text retrieval system, works fine with this .rtfd construction. It just ignores the pictures and indexes on the main TXT.RTF file

Windows NT on the Web

In mid-February, Microsoft released the final version of its Web server package, called the Internet Information Server (IIS). It's downloadable over the Internet, and so is the seven-disk Service Pack 3 upgrade you need to run it on your Windows NT 3.15 server. If, however, you have a slow link, it could take you the best part of a day. This column was already overdue by this time, so I'm very grateful to Barry Richards at Product Support Services, Microsoft UK, for biking over the Service Pack and the newly minted IIS on a CD-ROM they cut specially for me.

The service pack is commendably easy to apply, and the IIS install off the CD-ROM was even easier. The only executive decisions you're asked to make during the install is whether you want to allow anonymous ftp, and whether you want extra features such as a gopher server. I just said yes to everything, and within minutes I had a working Web server.

By default you start with a set of ready-made Web pages that give you an executive summary of the new software and a tour of the features. The moment I pointed browsers from the Windows NT Workstation, the NeXT machine and the Warp connect box at http://mitac_nt, the hostname I'd given to the Windows NT server, each picked up the home page across my network (which somewhat grandly gets to be called an "intranet" once you start doing this Web stuff on it). Having gone through a similar exercise last year with Linux-FT, which comes ready equipped with a built-in Web server, I didn't get quite the same initial thrill at making the connection, but I still find it amazing that three or four quite different operating systems can share the same world view so harmoniously.

More about this next month, I hope, when I get properly stuck in.

on your network, you'll need to rename them to <something>.tif, without the extra f. The DOS command to do this on an entire directory is simple:

```
ren *.tiff *.tif
```

The equivalent command in Unix is mv (it stands for move), but unfortunately if you try

```
mv *.tiff *.tif
```

inside an xterm window on the NeXT machine, you get a syntax error. (My copy of Linux-FT is more helpful. It throws up the error message "When moving multiple files, last argument must be a directory".)

In other words the Unix command is more flexible, because it moves files as well as renames them. But because it moves them, it can't handle renaming in batches.

Here's the neatest way I've found of doing a batch rename under Unix. The NeXT machine's shell by default is csh, not the most handy of shells, but you can get away with this, entered directly onto the command line:

```
> foreach f ( *.tiff )  
? mv $f $f:r.tif  
? end
```

The closest equivalent in DOS to the foreach command is FOR, but notice here that although foreach is entered at the command line, it's actually a multi-line command, setting up a loop that is terminated by end. When you hit Enter at the end of the first line, instead of executing the command, the tsh shell evaluates it (it will complain if it's garbage) and then gives you a continuation prompt in the form of the question mark. Now you enter the command line or lines to go inside the loop. The end line completes the loop, so when you hit the next Enter, the loop executes.

The foreach line expands the wildcard filename inside the brackets and feeds the results one at a time into the arbitrary variable f. Any name will do here. The line inside the loop simply passes each filename through the f variable to the mv command. The NAME of the variable is f, so in Unix \$f represents the VALUE of the variable. We use this twice, once in full (\$f) and once truncating the filename down to its root (cutting off the .tiff — that's what the :r does) and adding the new suffix .tif.

Taligent tails off into the distance

A couple of years ago at Comdex, round about the time I started this column, I was shut in a room with Joe Guglielmi, CEO of Taligent. For those of you not familiar with Taligent, it is the spin-off company that Apple and IBM spawned back in 1991 and Hewlett-Packard bought into a year later. Guglielmi was telling me that NeXT had been great in its day, but of course it was all over now. Microsoft's OLE and Cairo would be a distraction, OpenDoc would be the stepping stone, and Taligent would be the true culmination of modern, 32-bit, truly object-oriented computing technology.

Since the Apple connection was severed at the end of last year and the company was rolled back into IBM, Taligent is widely believed to be pining for the fjords. What was initially trumpeted as a revolutionary operating system discreetly evolved into a "multiplatform development environment", and is now lying quietly on its back at the bottom of its cage posing no trouble to anybody. Guglielmi left the company last September, and the Taligent Web page at <http://www.taligent.com>, under Latest News (last updated November 27, 1995) says: "No current job opportunities".

Meanwhile, NeXT hasn't taken over the

world, but it's certainly continuing to deliver, and its own Web page at <http://www.next.com> is on fire with founder Steve Jobs' latest preoccupation, WebObjects. The NeXTStep operating system, having evolved onto the major processor platforms (Intel, Sparc, HP-PA) is now positioning itself under the new name of OpenStep to become that "multiplatform development environment" (including DEC Alpha and PowerPC) that Taligent aspired to be. It's not 100 percent clear how this is all going to pan out, but from where I'm sitting the whole operation looks pretty buoyant.

Coincidentally, unless you believe in fate, Jobs himself has bounced back into the industry as a major player. Round about the time Taligent was posting "No current job opportunities" on its Web page, he made a public offering of another company of his, Pixar, and walked away from the deal with an estimated billion dollars in his pocket. I was talking to one analyst last week who seriously believes there's an outside chance he might get together with some friends and buy back Apple. Borland, thinks my analyst, is another possibility. Borland's Delphi development system certainly seems to owe a lot of its inspiration from NeXTStep's InterfaceBuilders

PCW Contacts

Chris Bidmead is a consultant and commentator on advanced technology. He can be contacted on bidmead@cix.compulink.co.uk