Exploring the NetWare Web Server, Part 3: A Complete Innerweb Solution

Articles and Tips: article

ALAN MARK
Corporate Integration Manager
Systems Engineering Division

01 Sep 1996

Describes how to use the NetWare Web Server 2.5 along with Novell's InnerWeb Publisher to set up a complete innerweb environment.

Introduction

Novell recently released version 2.5 of the NetWare Web Server and is including it with every copy of NetWare 4.11 ("Green River"). The Web Server has also been bundled with a number of related tools to form a product called Novell InnerWeb Publisher. This product is designed to give you everything you need to turn your existing NetWare network into an intranet. Whereas my previous two AppNotes dealt with just the NetWare Web Server (version 2.0), I thought it would be useful to wait until the release of version 2.5 and the InnerWeb Publisher to continue the series. This way I can cover the other components included in InnerWeb Publisher and show how they can be used along with the NetWare Web Server to create a complete innerweb solution.

Parts 1 and 2 of this series provided a foundational overview of setting up a web environment, covering such topics as installing and configuring the web server, using graphics, and writing simple scripts. (See "Exploring the NetWare Web Server" in the February 1996 issue and "Exploring the NetWare Web Server: Part 2" in the March 1996 issue of *Novell Application Notes*.)

The outline for this third AppNote is as follows:

- Introduce the components of InnerWeb Publisher, highlighting the new features in the NetWare Web Server version 2.5.
- Discuss some additional concepts relating to the use of NetWare Web Server in an innerweb environment, specifically in regards to integration with the NetWare operating system and services.
- Pull everything together by describing an example web site that demonstrates many of the ideas presented in all three AppNotes. You can download all of the sample files from the Systems Engineering Web Server at http://137.65.96.136:11111.

This AppNote is geared toward system administrators and "webmasters" who are looking for ideas on how to effectively integrate an internal web environment into their existing NetWare network.

Before proceeding, let me clarify my use of the terms "innerweb" and "intranet". Both words describe the same thing: a LAN that uses Internet protocols, such as HTTP (HyperText Transfer Protocol), to serve data. LAN clients receive data via an HTML (HyperText Markup Language) browser such as Netscape or Mosaic. I prefer the term "innerweb" to "intranet" because it's less likely to be confused with the word "internet"--especially in conversation.

Introducing Novell Innerweb Publisher

Novell Innerweb Publisher provides tools to help you create and publish HTML documents on an innerweb or on the Internet. The package includes:

- NetWare Web Server 2.5
- NetWare 4.1 Runtime
- Netscape Navigator 2.01 (50-user license)
- Softquad's HoTMetaL Lite (5-user license)
- QuarterDeck's I-Ware Lite, an IP-IPX Gateway (1-server license)

For more information on NetWare Web Server performance, see "NetWare on One Server Outperforms Windows NT on Four CPUs" in the June 1996 issue, and benchmark results from Shiloh Consulting at http://www.novell.com/.

NetWare Web Server 2.5

This second release of the NetWare Web Server has been optimized to take advantage of NetWare's superior file service engine. Besides being an ultra-fast HTTP server, version 2.5 supports the OS/2 name space, which means it now supports the downloading of Java applets. It runs as a NetWare Loadable Module (NLM) under the NetWare 4.1 operating system.

Another key feature of NetWare Web Server 2.5 is its ability to browse Novell Directory Services (NDS) objects through any World Wide Web browser. A user can explore an organization's NDS tree and its objects from any innerweb or Internet connection, using the standard Web browser interface. We'll discuss this capability in more detail later in this AppNote. For now, suffice it to say that this is the first step in integrating Novell's Directory with the World Wide Web. Future plans call for an NDS-to-LDAP (Lightweight Directory Access Protocol) gateway, which will permit NDS users to perform directory functions on another directory service.

NetWare 4.1 Runtime

The NetWare Web Server software can be installed on any existing NetWare 4.1 server. But in case you want to create a dedicated innerweb server, InnerWeb Publisher includes a runtime version of NetWare 4.1. This is a fully operational, two-user version of NetWare that acts as the base operating system for your innerweb server. The two-user license only applies to users actually logging in to the file server using NetWare Core Protocols (NCPs); you can have an *unlimited* number of users accessing the Web Server via HTTP.

HoTMetaL Light

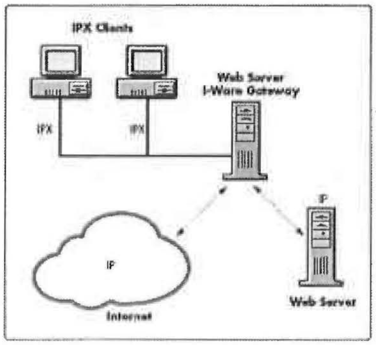
SoftQuad's HoTMetaL Light is a web authoring tool that lets you create HTML documents without having to know all of those esoteric HTML commands. It supports the HTML 2.0 specification, as well as many of the advanced HTML 3.0 and Netscape extensions. InnerWeb Publisher includes both the 16-bit and 32-bit versions of HoTMetaL Light so you can run it on Windows 3.1, Windows 95, or Windows NT workstations. I'll explain more about HoTMetaL when I discuss the example web site.

Quarterdeck IWare Lite

IWare is an IP-IPX Gateway that converts IPX packets into IP packets, and vice versa. It resolves a problem many system administrators are facing as they expand their LANs into intranets. Most NetWare networks use IPX as the standard protocol. However, you need TCP/IP to access web servers, and TCP/IP is more difficult to administer and maintain than IPX. The solution is to use an IP-IPX gateway to eliminate the need for TCP/IP to ever reach the LAN.

IWare has both a client and a server component. On NetWare workstations, the IWare Client translates TCP/IP packets generated by a web browser into IPX packets. When these IPX packets arrive at the NetWare Web Server, the IWare Server component (running as an NLM) converts them back into TCP/IP. From there, the packets are sent to the appropriate Web server, either within your own intranet or out on the public Internet (see Figure 1).

Figure 1: Iware acts as an IPX-to-IP gateway, translating IPX packets into TCP/IP packets to be sent to the appropriate web server, and vice versa.



IWare eliminates the need to install a TCP/IP stack on every client. Only one TCP/IP address is needed for the NetWare 4.1 server running the NetWare Web Server and the IWare Server component. Of course, you can continue to use TCP/IP on your workstations if you already have TCP/IP stacks loaded on them. With Novell InnerWeb Publisher, you can have any mixture of IPX and TCP/IP workstations on your intranet.

Since TCP/IP packets are only forwarded to those clients which requested the data, IWare also acts as a "firewall" to the external Internet. Hackers cannot explore a LAN that uses IWare as its TCP/IP router.

Netscape Navigator

Netscape is the leading web browser on the market. InnerWeb Publisher comes with a 50-user license for Netscape Navigator version 2.x, which supports Java and JavaScript, as well as a variety of enhancements to HTML that make web pages better looking. Netscape's plug-in architecture allows third-party vendors to create modules which provides extra functionality, such as displaying animation and playing real-time audio within the browser.

Innerweb Concepts

Most of the time and money expended on a web site goes toward keeping it operational and current. The ultimate goal of any web site is to have as many web pages as possible linked to real-time sources where content is updated automatically. Today, the amount of true automation that you can implement is limited, unless you use sophisticated back-end database links and clever scripting. In the near future, technologies such as Java promise to make real-time information easier to provide in the web environment.

Another goal for a web site is to provide links to existing data. Imagine being able to design an HTML document that integrates data residing on other NetWare servers, and that only allows access if the user who requests the data has rights to that data.

Expanding on this concept, wouldn't it be great if the links to the network data were dynamic, so that the underlying HTML document rarely needed to be updated?

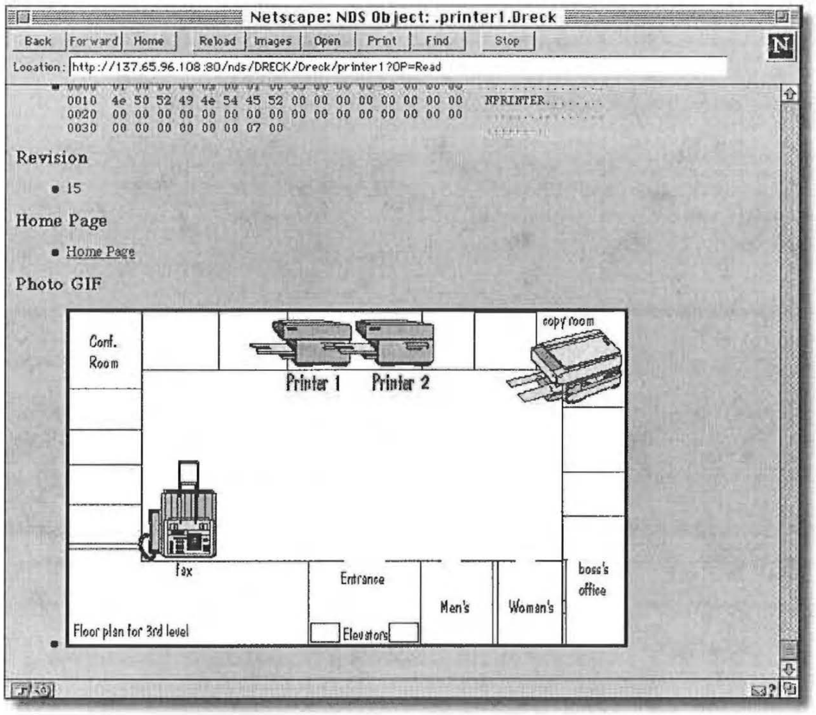
These types of implementations are possible with the NetWare Web Server, thanks to its ability to integrate with NetWare. Details on these and other concepts are provided in this section. Keep in mind, however, that some of the following ideas are applicable only to innerweb environments, or to global internetwork solutions that route NDS services across a WAN.

Browsing NDS

Why would someone want to browse NDS objects? One big reason is that NDS is a real-time database that is the source of nearly all network information. Up-to-date user information such as phone numbers, addresses, and even photographs are available to anyone with a browser. No separate database is necessary.

Another attraction of this NDS browsing feature is the ability to explore any object. Suppose you have set up a network printer and named it "Printer1". Users who want to print to that printer have no indication from the name as to the printer's location. But if you were to store a GIF file representing a floor plan as part of the printer's object in NDS, users could browse the object to discover its location (see Figure 2).

Figure 2: The NetWare Web Server allows you to store information about an object in NDS and then access that information from a standard web browser.



Netscape: NDS Object: .printer1.Dreck

Location: http://137.65.96.108:80/nds/DRECK/Dreck/printer1?OP=Read

```
0010  4e 50 52 49 4e 54 45 52 00 00 00 00 00 00 00 00   NPRINTER........
0020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0030  00 00 00 00 00 00 07 00
```

Revision

- 15

Home Page

- Home Page

Photo GIF

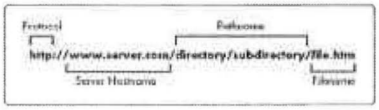
With the WEB_SNAP.DLL snap-in to the NWAdmin utility, you can store such things as photographs and home pages in NDS. The snap-in automatically extends the NDS schema to allow for the extra properties.

Incorporating NetWare Filenames in a URL

Suppose you are the network administrator for a department that has three NetWare file servers. You want to implement a web server so that people in other departments can easily find and access your data. But you don't want just anyone looking at your department's files--only those who have rights to do so. NetWare's file system security will take care of the access control, but how do you point a web browser to files that exist on different NetWare servers?

To answer this question, let's look at the structure of a Uniform Resource Locator (URL), as illustrated in Figure 3.

Figure 3: A URL is nothing more than a way to request a particular file on a host server.



Basically, a URL is just a fancy way of asking for data. It's not much different than asking for a file on a network drive. In DOS, a similar request looks like this:

```
drive:\directory\filename (for example, F:\PUBLIC\EDLIN.COM)
```

Using the Universal Naming Convention (UNC), a filename looks like this:

```
//server/volume/directory/filename
```

So in UNC, the name of a file in my home directory might be

The FILE Protocol. You can encapsulate a UNC name within a URL, but to do so the URL requires a special "protocol" called FILE. The syntax for using FILE is similar to using HTTP, FTP (File Transfer Protocol), Gopher, and Usenet. When the FILE protocol is used, the browser looks for a file or directory on a local or network file system.

Here is an example of a URL that points to an HTML file on the network:

```
file:////skime/sys/alan/projects/current.htm
```

where "skime" is the server name and "sys" is the SYS volume.

A file reference within an HTML document might look like this:

```
<a href="file:////skime/sys/alan/projects/current.htm"<<
For current projects, click here.</a<
```

This syntax is valid for a Windows 95 or Windows 3.x workstation running the NetWare Client 32 client software and Netscape v2.x. If the client is authenticated to the NDS tree, it will auto-connect to the file server. You do not have to map a drive prior to making the URL request.

For a Windows 3.x workstation running the NetWare DOS Requester (VLM) client software, the URL looks like this:

```
file:///F|/alan/projects/current.htm
```

where F| indicates drive F. In this case, a drive mapping is required prior to the URL request.

Supporting Multiple Desktop Operating Systems. For other types of NetWare clients besides DOS/Windows, the proper URL syntax is determined by how the desktop operating system interprets the server and volume name. For instance, on my Macintosh the URL for the file referenced above looks like this:

```
file:///SKIME.SYS/ALAN/PROJECTS/CURRENT.HTM
```

In the MacOS, server and volume names are combined into one desktop reference ("skime.sys"). Note also that the file is only accessible if the user is currently attached and authenticated to file server Skime.

For non-DOS filenames, spaces and non-standard characters are converted to their hexadecimal equivalents (for example, "%20" for space). For example, when I choose the "Open file..." option in my Macintosh web browser and select a local file, the URL is displayed as follows:

```
file:///Macintosh%20HD/Data/Novell%20%C4/NOVLOG1.GIF
```

Since many people find this extra clutter confusing, it's best to use only numbers and letters for file and directory names.

When supporting diverse desktop operating systems, you may need to provide multiple references to the same physical location. Initial HTTP requests to the server would include the browser type and client OS, found in the environment variable HTTP_USER_AGENT. For example:

```
HTTP_USER_AGENT = Mozilla/2.02 (Macintosh; I; PPC)
```

A script could then return a properly-syntaxed URL corresponding to the client platform detected in this environment variable. Here is a slice of Perl code that would perform this function:

```
$user_agent = "$ENV{'HTTP_USER_AGENT}";

  if ($user_agent =~ /Mozilla/) {

     #** Netscape **

  }

  if ($user_agent =~ /Macintosh/) {

     #** using Macintosh **

     $url = "file:///SKIME.SYS/ALAN/PROJECTS/CURRENT.HTM";

  }

  elsif ($user_agent =~ /Win16/) {

     #** using Windows 3.1**

     url$ = "file:///F|/alan/projects/current.htm";

  }

  elsif ($user_agent =~ /Win32/) {

     #** using Windows 95**
```

```
    $url = "file:////skime/sys/alan/projects/current.htm";
 }
```

Unfortunately, it is impossible to automatically determine network client shell information (VLM or Client32) from an HTTP request. The method shown above only works if you know that all workstation types are running the same NetWare client software; for instance, that all Windows 3.1 workstations are running the VLM client.

Using Directory Map Objects for Dynamic Links

One of the biggest headaches in maintaining a web site is making sure that hypertext links--those references that specify other documents or locations --are valid. When the document referenced by a hypertext link is moved or deleted, you see this all-too-familiar error message:

404 Not Found The requested URL /LONGFILO.HTM was not found on this server.

Changes to hypertext links affect browser bookmarks, scripts, and HTML files. In other words, they affect users and administrators alike.

Since NDS is a real-time database, it lends itself well to storing key link information. In fact, NDS already provides an object that fits the bill quite nicely: the Directory Map object.

Background on Directory Maps. Directory Map objects have been around since the release of NetWare 4.0 in 1993. The purpose of these objects is to store pointers to directories on any server on the network, and allow names to refer to server directories. Directory Maps are normally used in login scripts, batch files, and MAP commands. They use simple names instead of whole paths. For example, using a Directory Map object, the MAP command

```
MAP J:=APPSERVER/APPS:WORDPROC/WP60
```

becomes

```
MAP J:=WORDPERFECT
```

This simplified MAP command is then placed in users' login scripts and batch files. When the site upgrades from WordPerfect 6.0 to 6.1, all the administrator has to do is update the Directory Map object to point to "APPSERVER/APPS:WORDPROC/WP61". No changes need to be made to the users' login scripts or batch files. Best of all, the users don't have to know the directory structure on the server.

Using Directory Maps for URLs. Applying this concept to the web, a Directory Map object can be referenced as a URL. That way, whenever a reference changes, you only need to update one object in NDS. The syntax is:

```
file:////tree/Org/OU/DMap
```

where treeis the NDS tree name, Orgis the Organization, OU is the Organization Unit, and DMap is the Directory Map object name. For example, to reference files in the "wordperfect" directory on an application server, the URL might be:

```
file:////oak/skigreat/downhillers/wordperfect
```

Note: This syntax is valid only for Windows 95 or Windows 3.xworkstations running NetWare Client 32 and Netscape v2.x.

Remember that the referenced data is only accessible if the user is currently authenticated to tree "oak" and to server "appserver." There is no need to have a drive mapped to the server.

Another advantage to using NDS Directory Map objects as web references is that you can authorize specific users to change those objects without giving them rights to change the base web documents (which should be controlled by the web administrator). This scenario also relieves the web administrator from having to manage those references.

To have a user control a Directory Map, simply make the user a trustee of the object with property rights RWC (read, write, compare) to the Path property. If you grant the user Supervisor rights, he or she will be able to delete the Directory Map object.

Note: Currently, the URL syntax requires using the FILE protocol.Therefore, you cannot specify scripts or Server Side Includes (SSIs)as documents. In the future, complete URLs will be stored in NDS sothat any location or document type can be referenced.

Enabling NDS Browsing

With the NetWare Web Server, the NDS browsing capability is turned off by default. You enable it by launching the WEBMGR utility, specifying which server to access, and clicking on the "Enable NDS browsing" button. A line is created in the SYS:WEB/CONFIG/SRM.CFGfile that reads:

```
LoadableModule /nds/ sys:web/docs/cgi/ndsobj/ndsobj.nlm
```

You must restart the HTTP process in order for the changes to take effect.

To store images and home pages in NDS, the NWAdmin utility must load WEB_SNAP.DLL when it starts. This is done by adding a line in the NWADMIN.INI file that looks something like this (depending on where the DLL was installed):

```
[Snapin Object DLLs]

Websnap = c:\netware\web_snap.dll
```

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.

WHAT WILL YOU BUILD? | **sales@docketalarm.com** | **1-866-77-FASTCASE**

fastcase® Smarter legal research.