

## Perl 5.8 on NetWare®

# Novell® Developer Kit

June 21, 2006

[www.novell.com](http://www.novell.com)



## Legal Notices

Novell, Inc. makes no representations or warranties with respect to the contents or use of this documentation, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Novell, Inc. reserves the right to revise this publication and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes.

Further, Novell, Inc. makes no representations or warranties with respect to any software, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Novell, Inc. reserves the right to make changes to any and all parts of Novell software, at any time, without any obligation to notify any person or entity of such changes.

You may not export or re-export this product in violation of any applicable laws or regulations including, without limitation, U.S. export regulations or the laws of the country in which you reside.

Copyright © 1993-2003 Novell, Inc. All rights reserved. No part of this publication may be reproduced, photocopied, stored on a retrieval system, or transmitted without the express written consent of the publisher.

U.S. Patent Nos 5,553,139; 5,553,143; 5,677,851; 5,758,069; 5,784,560; 5,818,936; 5,864,865; 5,903,650; 5,905,860; 5,910,803 and other Patents Pending.

Novell, Inc.  
404 Wyman Street, Suite 500  
Waltham, MA 02451  
U.S.A.  
[www.novell.com](http://www.novell.com)

*Online Documentation:* To access the online documentation for this and other Novell developer products, and to get updates, see [developer.novell.com/ndk](http://developer.novell.com/ndk). To access online documentation for Novell products, see [www.novell.com/documentation](http://www.novell.com/documentation).

## **Novell Trademarks**

For a list of Novell Trademarks, see [Novell Trademark and Service Mark list \(http://www.novell.com/company/legal/trademarks/tmlist.html\)](http://www.novell.com/company/legal/trademarks/tmlist.html)

## **Third-Party Materials**

All third-party trademarks are the property of their respective owners.



# Contents

<b>Preface</b>	<b>7</b>
<b>1 Getting Started</b>	<b>11</b>
1.1 Setting Up	11
1.2 Configure Apache 2.0 Web Server	11
1.3 Configure Apache1.3 Web Server	12
1.4 Configure NetWare Enterprise Web Server	12
1.4.1 Configure the CGI Directory	13
1.4.2 Configure and test your database connection - Oracle	14
1.5 Test Your Perl Setup	17
1.6 Executing a Oracle Database Sample	18
1.7 Executing Perl Scripts from the System Console	18
1.8 Perl Debugger on NetWare	19
1.9 Using Perl for CGI Scripting	19
1.10 Using Perl for System Administration	19
1.11 Accessing Pervasive from Perl	20
1.12 Accessing MySQL from Perl	20
1.12.1 Executing a MYSQL Database Sample	21
1.13 Accessing UCS From Perl	23
1.13.1 Get and Set	23
1.13.2 Example	24
<b>2 Functions and Extensions</b>	<b>27</b>
2.1 Supported Functions and Extensions	27
2.1.1 Remarks	27
2.2 Unsupported Functions and Extensions	27
2.2.1 Standard Unsupported Functions and Extensions	27
2.2.2 System V Unsupported Functions	28
<b>3 Perl_LDAP Modules</b>	<b>29</b>
<b>4 Perl Links</b>	<b>31</b>
4.1 Web Pages	31
4.2 Online Guides	31
4.3 Publishers' Web Pages	31
<b>A Sample mod_perl Configuration File</b>	<b>33</b>
A.1 mod_perl Configuration File	33
A.2 Description of Configuration File	33

<b>B</b>	<b>mod_perl Sample</b>	<b>35</b>
<b>C</b>	<b>Secure Scripting</b>	<b>37</b>
C.1	Introduction .....	37
C.2	Script Operations and Security .....	37
C.3	Remote File System Support .....	38
C.4	Configuration .....	40
C.5	Changes Required in the Scripting Applications .....	40
C.6	Limitations .....	42
<b>D</b>	<b>Specifications for Perl for NetWare in CPAN Site</b>	<b>45</b>
<b>E</b>	<b>Building Perl Extensions</b>	<b>47</b>
E.1	Software Requirements .....	47
E.2	Building Perl NLM .....	47
E.2.1	Setting up the Environment for Building Perl NLM .....	47
E.2.2	Make Process .....	48
E.2.3	Building Perl Interpreter .....	48
E.3	Installing Perl on NetWare .....	48
E.4	Building Perl Extensions .....	49
<b>F</b>	<b>Revision History</b>	<b>51</b>
F.1	June 2006 .....	51
F.2	October 2003 .....	51
F.3	March 2003 .....	51
F.4	November 2002 .....	51
F.5	September 2002 .....	51
F.6	May 2002 .....	51
F.7	February 2002 .....	52
F.8	June 2001 .....	52

# Preface

This documentation describes the Perl programming language and how Novell has implemented it on NetWare.

## Introduction

The Practical Extraction and Report Language (Perl) was created by Larry Wall and is designed specifically to process text. It was originally founded on the UNIX platform, and later on it was ported onto many platforms. The porting effort required for moving Perl scripts between different platforms is very minimal.

It is a very simple language to learn as Types and Structures are easy to understand. It borrows most of its syntax from C and can be extended in a modular fashion. As a text processor, Perl is especially useful in parsing and report generation. Perl is valuable for both dynamic Web page generation and server task automation.

Perl is categorized as an *open-source language* since the sources are freely available to the programming public. This means that the Perl source is available to anyone who wants to access it. You can download the latest version of Perl from the [Perl Home Page \(http://www.cpan.org/ports/index.html#netware\)](http://www.cpan.org/ports/index.html#netware).

The Perl language is maintained and updated by a team of core developers known as the Perl Porters. You can learn about the latest Perl Porters developments by visiting [The Perl Porters Page \(http://www.xray.mpe.mpg.de/mailling-lists/perl5-porters\)](http://www.xray.mpe.mpg.de/mailling-lists/perl5-porters). To subscribe to the Perl Porters mailing list, send a request to [perl5-porters-request@perl.org](mailto:perl5-porters-request@perl.org).

Perl 5.8 on NetWare is based on the corresponding Perl open-source code base. Because the team that ported Perl to NetWare took care to maintain standard Perl functionality, programmers can base their Perl for NetWare development on industry-standard Perl reference sources.

## So Why is Perl valuable to NetWare?

Perl offers two main advantages on a NetWare system: CGI (Common Gateway Interface) Scripting and System Administration.

CGI scripting is the most common way for a Web server to interact with users. You can use CGI scripting to increase the sophistication and functionality of your Web pages. See [Section 1.9, “Using Perl for CGI Scripting,” on page 19](#).

Because Perl is such a powerful text processor, it has proved to be a valuable system administration tool on NetWare systems. Procedures that require manual interaction in NWAdmin can be automated using Perl. See [Section 1.10, “Using Perl for System Administration,” on page 19](#).

## mod\_perl 2.0 for NetWare

mod\_perl is an Apache extension written by Doug MacEachern for embedding the Perl interpreter in Apache Web server. It integrates the power of Perl with the stability of the Apache Web server and provides Web applications with high performance. As the interpreter is embedded in the Web server,

the startup and compilation time for any HTTP request comprising a Perl script is reduced. All the scripts and the modules can be compiled once when the server is started and rerun many times when the requests come.

`mod_perl` provides full access to the Web server and enables intervening and customizing the various phases of HTTP request processing like:

- ♦ URL to filename translation
- ♦ Authorization
- ♦ Response Generation
- ♦ Logging

## Perl Extensions for NetWare

The three types of Perl extensions are:

- ♦ [“Standard modules” on page 8](#)
- ♦ [“NetWare specific modules” on page 8](#)
- ♦ [“Other public domain modules” on page 8](#)

### Standard modules

All Perl modules shipped with the standard distribution are included in the NetWare. Some of the standard modules that get shipped with NetWare are `Fcntl`, `IO` and `SDBM_FILE`.

### NetWare specific modules

The NetWare specific modules are specific to NetWare and will not work on any other platform.

The following NetWare specific modules are included in Perl 5.8 on NetWare:

- ♦ [“CGI2Perl” on page 8](#)
- ♦ [“Perl2UCS” on page 8](#)

#### CGI2Perl

This is a CGI interface for running Perl on NetWare. The very purpose of porting Perl onto NetWare is to aid common gateway interface (CGI) programmers. CGI extension allows you to use CGI scripting to interact with users on Web servers.

#### Perl2UCS

You can call Novell's Universal Component System (UCS) using Perl2UCS extension. UCS has its strength in its reusability, and it allows you to develop applications that use software components on any network machine. Using a single interface, it can consume Java, Java Beans and Classes, Perl, C, and C++ programs to provide Web scripting and server side functionality.

### Other public domain modules

[DBI](#) and [DBD::ODBC](#) are the public domain modules that are available with Perl 5.8 on NetWare.



In addition, there are other public domain modules like LDAP, XML etc. that can be downloaded from CPAN site and easily built for NetWare.

## Database Support in Perl 5.8 on NetWare

**DBD::ODBC** driver, provided as a Perl extension, enables the database support in Perl5 for NetWare. This driver uses **DBI** as an interface between an application and one or more database drivers.

### DBI

Database independent interface (DBI) is the database access application programming interface (API) for Perl. DBI defines a set of functions and variables that provide a consistent database interface independent of the actual database that is being used. DBI also provides a standard interface and framework for the drivers to operate. The drivers contain an implementation of the DBI methods written using the private interface functions of the corresponding database.

The DBI extension for NetWare defines a set of functions and variables that provide a consistent database interface independent of the actual database being used. DBI is just an interface while the drivers such as **DBD::ODBC** do the real work.

### DBD::ODBC

DBD::ODBC is the database driver for ODBC. To access a database, a data source name (DSN) must be created. See [Section 1.4, “Configure NetWare Enterprise Web Server,” on page 12](#) for information on setting up a DSN. Either Oracle or [Pervasive.SQL 2000 \(http://developer.novell.com/wiki/index.php/Pervasive.SQL\\_Software\\_Developer\\_Kit\)](http://developer.novell.com/wiki/index.php/Pervasive.SQL_Software_Developer_Kit) can be used as a database.

For more information on accessing Oracle, see [“Configure and test your database connection - Oracle” on page 14](#).

### DBD::MYSQL

DBD::MySQL is a database interface for accessing the MySQL database from Perl 5.8. It is based on the DBD-MySQL-2.1017 version available on CPAN.

For more information on accessing MySQL database see, [Section 1.12, “Accessing MySQL from Perl,” on page 20](#).



# Getting Started

# 1

This section describes the setup procedures for Perl 5.8, and mod\_perl 2.0 for NetWare®. These procedures only provide general guidelines. Most of the steps consist of suggested uses for Perl 5.8 on NetWare and are intended to be viewed as launching points for your Perl projects rather than the solutions to your system needs.

---

**NOTE:** mod\_perl 2.0 will work only with Perl 5.8 for NetWare.

---

## 1.1 Setting Up

You need to meet the following system requirements before implementing the tasks.

**Table 1-1** System Requirements for Perl 5.8 for NetWare

Hardware/SoftWare	Version
Operating system	NetWare 5.x and above
Web Server	NetWare Enterprise server or Apache Web server
Perl	Perl 5.8 for NetWare

**Table 1-2** System Requirements for mod\_perl 2.0 for NetWare

Hardware/SoftWare	Version
Perl	Perl 5.8 for NetWare
Web Server	Apache Web server 2.0.47 for NetWare. This can be downloaded from the <a href="http://www.apache.org/dist/httpd/binaries/netware/">Apache site (http://www.apache.org/dist/httpd/binaries/netware/)</a>

Refer to readme.txt available in the installed folder before proceeding further.

## 1.2 Configure Apache 2.0 Web Server

Proceed with the following steps to configure Apache 2.0 Web server for Perl and mod\_perl.

- 1 Open the file `sys:\apache2\conf\httpd.conf`.
- 2 To configure mod\_perl, enter the following configuration details in httpd.conf file.  

```
include sys:\apache2\conf\mod_perl.conf
```

Refer to “[Sample mod\\_perl Configuration File](#)” on page 33 to see a sample mod\_perl.conf file.1‘
- 3 Restart the Web server and test the setup by typing the URL similar to  
`http://server_name:port_number/perl/samples/hello.pl`

The Perl script (hello.pl) available in the alias\samples directory is executed through mod\_perl and greeting is displayed in the browser. To invoke the Apache extension for Perl, enter the following URL in the Web browser.

```
http://server_name:port_number/hello
```

Refer to “[mod\\_perl Sample](#)” on page 35, for a sample using mod\_perl.

## 1.3 Configure Apache1.3 Web Server

Proceed with the following steps to configure Apache Web server for Perl 5.8 on NetWare.

- 1 Open the file sys:\apache\conf\httpd.conf.
- 2 Enter the following configuration details in httpd.conf file.

```
LoadModule lcgimodule modules/mod_lcgimodule.nlm
<IfModule mod_lcgimodule.c>
  AddHandler lcgimodule-script pl
  LCGIModuleMap sys:\perl\lcm\cgi2perl.nlm .pl /perl
  AddEnvVar PERL_ROOT sys:\perl\web

  ScriptAlias /perl sys:\perl\web
</IfModule>
```

---

**NOTE:** If the Perl scripts are kept in a different directory enter the corresponding path for ScriptAlias and AddEnvVar.

---

- 3 Restart the Web server and test the setup by typing the URI similar to

```
http://servername/samples/env.pl
```

or

```
http://servername/perl/samples/env.pl
```

## 1.4 Configure NetWare Enterprise Web Server

By default, the Web server will be configured for Perl in NetWare5.1 and above. If you need to reconfigure the Web server Complete the following steps to enable Perl support for the NetWare Enterprise server.

- 1 Open the NetWare Web Manager in your browser. The syntax for this URL is typically:

```
https://server_name:adminport_number
```

- 2 Log in to the server as an administrator and the server administration screen is displayed.



3 Click <Web server name> to display the NetWare Enterprise Web Server administration screen.



4 Then, [Configure the CGI Directory](#).

---

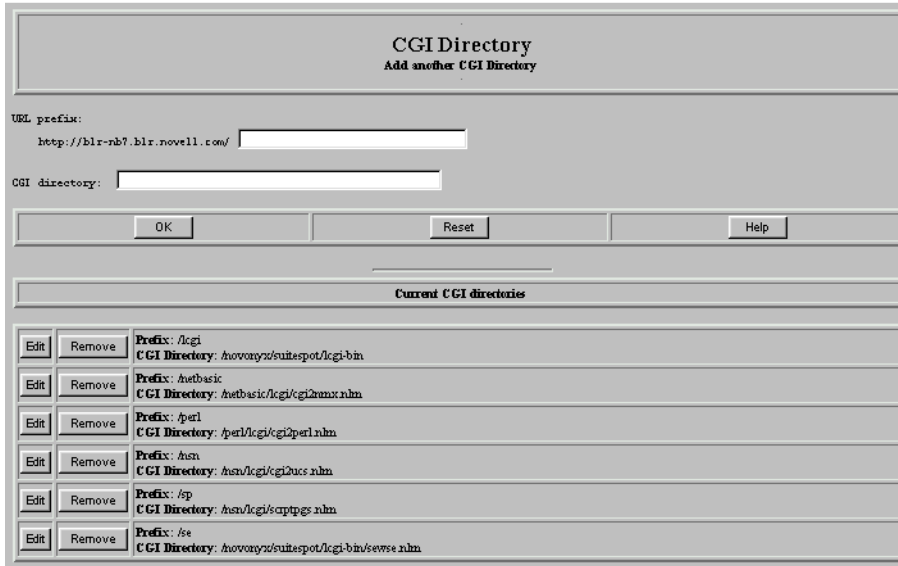
**TIP:** The ON/OFF indicator must be set to On for executing this procedure.

---

### 1.4.1 Configure the CGI Directory

Before Perl can be used with the your Web Server, ensure that the CGI Directory is properly configured.

1 From the NetWare Administration screen, click Programs, then click CGI Directory. Add New CGI directory screen is displayed.



- 2 Click the Edit Button next to the line that reads *Prefix:/perl*.
- 3 Ensure that the CGI Directory for Perl is set to `/Perl/lcgi/cgi2perl.nlm` and the URI is `/perl`. If the values are not set, change them and then click OK.
- 4 The `Obj.conf` file resides in the `sys:\Novonyx\suitespot\https-servername\config` directory. Enter the following line defining `PERL_ROOT`, in the `Obj.conf` file using a text editor.

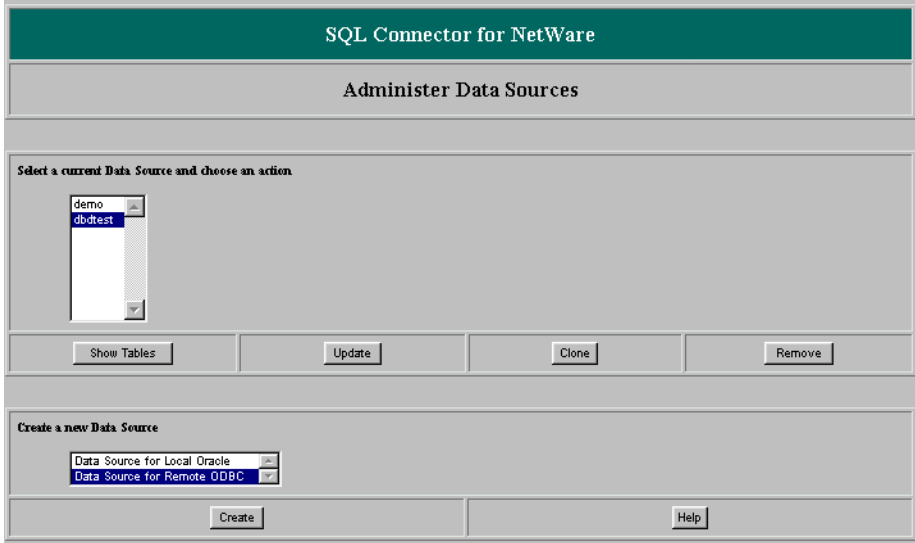
```
Init fn="lcgi_map_init" PERL_ROOT="sys:\perl\web"
```

Since this path defines the directory that is searched for scripts when running Perl from the Web server, ensure that the specified directory exists and is correctly named.
- 5 Then [Configure and test your database connection - Oracle](#).

## 1.4.2 Configure and test your database connection - Oracle

The ODBC connection can be set up through the specified Web server using SQL connection manager.

- 1 Click Programs from the NetWare Enterprise Web server administration screen.
- 2 Then, click Server-Side JavaScript.
- 3 Select Yes to Activate the Server-Side JavaScript application environment. This will allow you to test the database connection after it has been created. Click OK.
- 4 Click Novell Servlet gateway.
- 5 Select Yes to Activate the Novell Servlet Gateway. This will allow you to run the servlet that helps to create your Data sources. Click OK.
- 6 Click ODBC Data sources.
- 7 Click Administer, then log in as the Administrator. The SQL Connector for NetWare screen is displayed.




---

**NOTE:** A list of existing Data Sources is displayed only if configured.

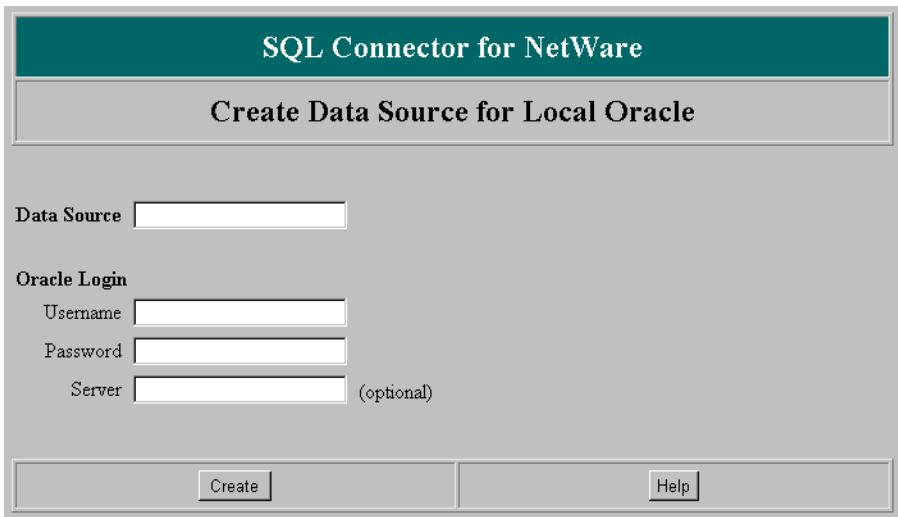
---

- 8 Highlight the data source type and click Create. The system assumes that you are connecting to an Oracle database on your NetWare server. Create Data Source for Local Oracle screen is displayed.

---

**NOTE:** Choose Remote ODBC if you are connecting to a database on NT server.

---



- 9 Fill the fields based on the following field descriptions

**Data Source:** The name you want to assign to your database.

**UserName:** The username assigned by Oracle.

**Password:** The password assigned by Oracle.

**Server:** (Optional). The Oracle TNS connection name for accessing a remote Oracle database, such as ora\_world. The Oracle database administrator defines the TNS names.

---

**NOTE:** If you choose Remote ODBC as your Data source type in [Step 9 on page 15](#), this screen will require both Windows NT login information and ODBC data driver information. In the Data source field, enter the name you want to assign to your Windows NT database connection. In the ODBC Data Source Name field, enter the name of an existing ODBC data source. For the list of existing data sources click Control panel > ODBC in the NT machine.

---

**10** Click Create to create the data source.

A monitor window gives details about the program's progress as it creates the data source. When the data source creation is completed, a status window will appear.

**11** Click OK to return to the Administer Data Sources screen. The new data source is displayed in the Data Sources list.

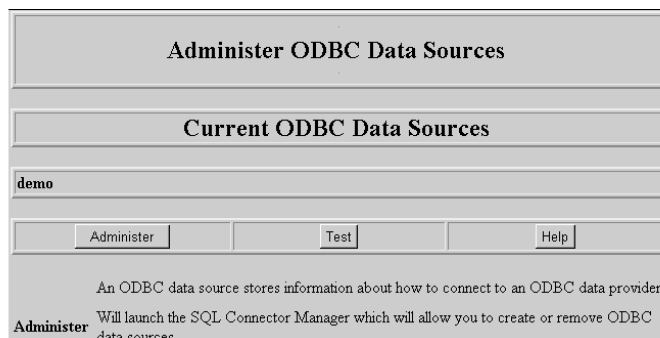
**12** Highlight the data source and click Show Tables to display a list of tables in the data source. These tables are useful to test the database connection.

For more information about database connectivity, see:

- ♦ [Netware 5.1 Documentation](http://www.novell.com/documentation/lg/nw51/docui/index.html). (<http://www.novell.com/documentation/lg/nw51/docui/index.html>). Click Database Services > Novell SQL Connector.
- ♦ [Netscape Enterprise Web Server Administration Guide](http://www.novell.com/documentation/lg/nes4nw/docui/index.html). (<http://www.novell.com/documentation/lg/nes4nw/docui/index.html>)
- ♦ [Novell® ODBC drivers for Novell eDirectory™](http://developer.novell.com/ndk/doc/odbc/odbc_enu/data/hvdyg1lg.html). ([http://developer.novell.com/ndk/doc/odbc/odbc\\_enu/data/hvdyg1lg.html](http://developer.novell.com/ndk/doc/odbc/odbc_enu/data/hvdyg1lg.html))
- ♦ [SQL integrator: A Data Request Broker for heterogeneous Data Access](http://developer.novell.com/research/appnotes/1998/may/04) (<http://developer.novell.com/research/appnotes/1998/may/04>) (*Novell App Notes, May 1998*)

Once the database connection is configured it should be tested to ensure proper functioning of the database. Proceed with the following steps on your NetWare Enterprise Web Server administration screen, to test the configuration.

**1** Click ODBC Data Sources. Administer ODBC Data Sources screen is displayed.



**2** Click Test to launch the dbadmin JavaScript utility. Connect To Database screen is displayed.

---

**IMPORTANT:** The Server-Side JavaScript should be activated first, to execute this procedure properly. See step 2 under [“Configure and test your database connection - Oracle” on page 14](#).

---



PARAMETER	INPUT	DESCRIPTION
Server Type:	<input type="text" value="ORACLE"/>	Select the type of database server for this connection. Some types may not be available for certain platforms. See the <a href="#">documentation</a> for more information.
Server Identifier:	<input type="text" value="your_server_id"/>	This parameter identifies the database server. The information required for this parameter is vendor specific. <ul style="list-style-type: none"> <li>• Informix - <i>Informix Server</i> from INFORMIXSERVER environment variable</li> <li>• Oracle - <i>Service Name</i> from tnsnames.ora file</li> <li>• Sybase - <i>Server Name</i> from sql.ini or interfaces file</li> <li>• ODBC - <i>DSN</i>(Data Source Name)</li> <li>• DB2 - <i>Local Database Alias</i></li> </ul>
User ID:	<input type="text" value="your_user_id"/>	This is a username for accessing the database server.
Password:	<input type="password" value="*****"/>	The password for this database server users.
Database:	<input type="text"/>	This should be blank unless your database server uses the concept of individually named databases within the database server. For example, Informix and Sybase require this, however, it must be left blank for Oracle, DB2 or ODBC.
		<p>If the connection succeeds, the next page will be the "Execute Query" page. If it fails, the next page will be an error page which will display information to help resolve the problem.</p> <p style="text-align: right;"><input type="button" value="Connect"/></p>

**3** Enter the data specific to your database connection, and click Connect To Test your Connection.

---

**TIP:** If you have an ODBC server type, enter the newly created Data Source Name in the Server Identifier field.

---

If the connection is correct, the *Execute Query* screen will be displayed. Queries can be issued against the tables that were displayed when show Table was selected in step 13 of “[Configure and test your database connection - Oracle](#)” on page 14.

If the connection fails, an error page containing the solution for the corresponding problem is displayed.

## 1.5 Test Your Perl Setup

### From the NetWare Enterprise Web Server

To test Perl setup from the NetWare Enterprise Server for NetWare, type the following URL in your Web browser:

```
http://servername/perl/samples/echo.pl?var1=a&var2=b
```

echo.pl is a Perl script that ships with NetWare Enterprise Server for NetWare and is located in the sys:perl\web\samples directory. This script returns the CGI parameters back to your browser.

### From the NetWare System Console

To test Perl setup from the command line, type the following command at the system console:

```
Perl Script name
```

where *Script name* is the name of the Perl program you want to run.

For more details refer to [Section 1.7, “Executing Perl Scripts from the System Console,”](#) on page 18.

---

**NOTE:** On NetWare5.x server, typing Load at the command line is optional.

---

## 1.6 Executing a Oracle Database Sample

To execute the sample scripts, copy the perldocs directory to the following locations:

- ♦ **For Enterprise Web server:** sys:\novonyx\suitespts directory
- ♦ **For Apache Web server:** sys:\apache\htdocs directory

Follow these steps to run a database sample on your NetWare server.

1. Ensure that both Oracle and SQL Connector are running.
2. At the NetWare System console command prompt, type

```
vtxhome sys:
```

to set VORTEX\_HOME to the SYS: volume.

The sample script perlbdb.pl under sys:\perl\script uses ODBC to access an Oracle database. For web based access, use the perlbdbquery.htm sample in perldocs\database.

## 1.7 Executing Perl Scripts from the System Console

You can run Perl scripts from the system console of NetWare 5.x and later versions.

### NetWare 5.x and Later

You can run the Perl scripts from the directory of your choice, provided you set the environment variable PERL\_ROOT. Type the following command at the console prompt:

```
env PERL_ROOT=SYS:\USER_DIRECTORY
```

where, *USER\_DIRECTORY* is the directory you have placed your scripts.

By default, the Perl scripts are placed in sys:\perl\scripts directory.

Each time you run a Perl script from the console prompt, a new screen comprising the results of the script execution appears. To close, press Enter after executing the script.

To prevent a new screen from popping up, run Perl scripts with the following command:

```
Perl --noscreen <Perl script name>
```

---

**NOTE:** *noscreen* succeeds two - (Hyphen)

---

To automatically go to the console prompt without pressing Enter from the new screen, run Perl scripts with the following command:

```
Perl --autodestroy <Perl script name>
```

---

**NOTE:** *autodestroy* succeeds two - (Hyphen)

---

To redirect the standard input and output using the greater-than (>) and lesser-than symbols (<) respectively, run Perl scripts with the following command:

```
Perl Perl script name < ascii file name  
Perl Perl script name > ascii file name
```

To redirect the standard error output using the `2>` symbol, run Perl scripts with the following command:

```
Perl Perl script name 2> ascii file name
```

To redirect both the stdout and stderr using the `&>` symbol, run Perl scripts with the following command:

```
Perl Perl script name &> ascii file name
```

For more options, run Perl with the following command:

```
Perl -h
```

## 1.8 Perl Debugger on NetWare

The Perl debugger allows you to detect error in your program logic by viewing the Perl script, setting break points and viewing stack information. You can invoke the debugger on NetWare using `-d` switch at the command prompt as follows:

```
Perl -d Script name
```

Before you invoke the `-d` switch, you should ensure that the script you have written is free from syntax errors and that it complies correctly.

After successfully invoking the debugger, the following debugger prompt appears on the console:

```
DB<1>
```

Type `h h` at the debugger prompt to display the summary of the commands and their usage instructions.

## 1.9 Using Perl for CGI Scripting

CGI (Common Gateway Interface) scripting is the most common way for a Web server to interact with users. You can use CGI scripting to increase the sophistication and functionality of your Web pages. For example, CGI scripts are frequently used to

- ♦ Access information such as server status and time from a Web server and display it on a Web page.
- ♦ Increment a counter by 1 each time a Web page is accessed.
- ♦ Process information entered in HTML forms by accessing a database using the Web server and generating a table that contains the query data.

More information about using Perl with CGI scripting is available in [“Perl Links” on page 31](#).

## 1.10 Using Perl for System Administration

Because Perl is such a powerful text processor, it has proved to be a valuable system administration tool on NetWare systems. Procedures that require manual interaction in NWAdmin can now be automated using Perl.

For example Perl can be used for setting up a University enrollment and registration system. Through NWAdmin, you would manually assign a student ID and password for each enrollee. With Perl for NetWare, you could write a Perl routine that interacts with the student to automatically generate a student ID and password when the student fills out an HTML form. In addition, the Perl program could allow students to change their class schedules as often as necessary at any time of the day.

In short, Perl for NetWare offers you the luxury of creating custom Web-based solutions to handle heavy workflow while easing the burden of network administration.

Other Perl programming examples:

- ♦ Change file ownership
- ♦ Generate random passwords for users
- ♦ List all mounted volumes and the available disk space on each volume.
- ♦ Report disk space used by each user

“[Perl Links](#)” on page 31 provides more information about using Perl.

## 1.11 Accessing Pervasive from Perl

Database can be accessed from Perl using DBI and DBD::ODBC or using the NDO UCX component. The database can be Pervasive.SQL 2000, MS-Access, or MySQL on NetWare. To access a database, a Data Source Name (DSN) has to be created.

The sample scripts createtab.pl and querydb.pl in the sys:\perl\scripts directory and perlbdbquery.htm in the perldocs\database directory show how to access a database using DBI and DBD::ODBC.

The sample script displaytab.pl in sys:\perl\scripts shows how to access a database using NDO UCX component.

Additionally, database running on the Windows machine can be accessed using Remote ActiveX provider. See [UCS documentation \(http://developer.novell.com/ndk/ucs.htm\)](http://developer.novell.com/ndk/ucs.htm) for details on using Remote ActiveX provider.

For more details about accessing a Pervasive.SQL 2000 database, refer to the [NDK site \(http://developer.novell.com/ndk/psqlsdk.htm\)](http://developer.novell.com/ndk/psqlsdk.htm).

## 1.12 Accessing MySQL from Perl

To access MySQL from Perl, the search path should be set to the bin directory containing the libmysql.nlm file, by entering the following command at the server console:

```
search add sys:\mysql\bin
```

Ensure that libmysql.nlm is available in the bin directory. The search path can also be set by adding the above line in the autoexec.ncf file.

Then, execute the perl-mysql sample (mysql.pl), available in sys:\perl\scripts directory by entering a valid username and password, and the database information.

## 1.12.1 Executing a MYSQL Database Sample

To execute the sample scripts, copy the perldocs directory to the following locations:

- ♦ **For Enterprise Web server:** sys:\novonyx\suitespot directory
- ♦ **For Apache 2.0 Web server:** sys:\apache\htdocs directory

The following Web based Perl sample will be used to access the MySQL database which resides on the NetWare Server.

```

# Sample Example code for accessing MySQL database from Perl
using the perl-MySQL module
use DBI;
use CGI;
my $cgiobject=new CGI;
print $cgiobject->header();

print "<html>";
print "<BODY TEXT=\"\#000000\" LINK=\"\#FF0000\">";
print "<FONT FACE=ARIAL>";
print "<html><head><title>Perl-MySQL</title></head>\n";
print "<body><h1>Perl-MySQL</h1>";

# Database Name
$databse="user_databse";

# HostName of the MySQL Server. HostName of the Remote
MySQL server also can be used.
$hostname="hostname";
# Port name on which MySQL server is running.
$port=3336;
# User Name who has the access to manipulate the mentioned
database.
$user="user";
$password="";
$DSN =
"DBI:mysql:database=$databse;host=$hostname;port=$port";

$DB_HANDLE = DBI->connect($DSN, $user, $password) or die "Not
able to Connect. Check the UserName and Password";
$drh = DBI->install_driver("mysql");
$sth = $DB_HANDLE->prepare("SELECT * FROM Emp");
$sth->execute;

print "Number of Rows    ::  ";
$numRows = $sth->rows;
print $numRows;
print "<br>";
print "Number of Fields    ::  ";
$numFields = $sth->{'NUM_OF_FIELDS'};
print $numFields;
print "<br>";

print "<br> The Values are"."<br>";
while (my $ref = $sth->fetchrow_hashref()) {
# Change the Field Name based on the Requirement.
    print "<br>";
    print "Host = $ref->{'Name'}: Sex = $ref->{'Sex'}:
User = $ref->{'Prof'} " ;
    print "<br>";
}
$sth->finish;

print "<hr>";
print "</body></html>\n";

```

## 1.13 Accessing UCS From Perl

The Universal Component System (UCS) provides access to tools and services outside of Perl (For more details visit the [the NDK site \(http://developer.novell.com/ndk/ucs.htm\)](http://developer.novell.com/ndk/ucs.htm)). When writing a Perl script that uses UCS, you must include the Perl Module `Perl2UCS.pm` (formerly `UCSExt`) and locate the file. On a NetWare server, make sure this file resides in the directory `sys:\perl\lib`.

Along with your other Perl Modules, you must include the `perl2ucs.pm` module as follows

```
use Perl2UCS;
```

Use the new method to create a Perl object for a UCS object. Preface this call with the Perl extension (for example, `Perl2UCS -> new`).

```
$server = Perl2UCS->new("UCX:Server");
```

Make sure that `Perl2UCS.nlp` resides in the directory, `sys:\perl\lib\auto\perl2ucs`. Otherwise an error message indicating that the UCS Extension module (`perl2ucs.nlp`) was not found, is displayed.

---

**NOTE:** Perl install does not include UCS and UCX components. For download of UCS and UCX components visit [the NDK site \(http://developer.novell.com/ndk/ucs.htm\)](http://developer.novell.com/ndk/ucs.htm)

---

### 1.13.1 Get and Set

To get or set a property for a UCS object refer to the following examples.

```
#get example
$retVal = $server->{"LanBoards"};
#set example
$server->{"LoginStatus"}=1;
```

To make method calls on the UCS object you have created, make the call as shown below:

```
$retVal = $dir->login(username,password);
```

The constants exposed by the UCX components are available to the Perl script. The constants are invoked in the same way as getting a property.

See the following code snippet that allows to access NWDIR UCX component from a Perl script and list the contents of the directory under the specific context.

## 1.13.2 Example

```
use Perl2UCS;

#This will pull in the definitions and declarations
#from the module file. Also this loads the necessary
#module into memory if not yet loaded. In this case it
#imports definitions and declarations from Perl2UCS.pm
#present in sys:\perl\lib directory and also loads
#Perl2UCS.nlp which is an NLM from
#sys:\perl\lib\auto\ucsext directory. Also this loads
#UCSCore.nlm, which is in sys:\system directory.

$Dir = UCSExt->new("UCX:NWDIR") or die "Unable to create Directory object \n";

#This instantiates the directory object, and stores it
#as a Perl object for later use. This loads ucs2ucx.nlm
#and ucxmgr.nlm from sys:\system directory. It also loads
#_nwdir.nlm which is the directory UCX component from #sys:\ucs\ucx
directory.

$Dir->Login("username", "password") or die "Failed to login.
Please check the username and password.\n";

#This will call the login method of the directory
#object and logs into the tree with the specified
#user name and password. If the user is in a different
#context, set the 'FullName' property with the correct
#context before calling the login method.
#$Dir->{"FullName"} = "nds:\\\treename\\context";
#Also if you want to enumerate the contents of
#a different context, set the 'FullName' property
#before starting the enumeration.

$Entries = $Dir->{"Entries"};

#This gets the entries object, which is a property in
#the directory object.

$Entries->Reset();

#This calls the 'reset' method of the entries object,
#which initializes for a search of all entry objects.
#The following loop, gets all the entry objects, by
#calling the next method of the entries object. Also
#this prints the 'shortname' which is a property on
#the entry object.

while($Entries->HasMoreElements()) {
    $Entry = $Entries->Next();
    $ShortName = $Entry->{"ShortName"};
    print "$ShortName \n";
}

#The following calls the 'logout' method of the
#directory object, which logs out the user out
#of the directory service.
```



```
$Dir->Logout();
```

---

**NOTE:** Installing the latest UCS copies more samples into the default directories perl\scripts and perl\web\sample directories. The samples can also be stored in docs\perlroot\sample. The environment variable PERL\_ROOT in the Obj.conf file should be changed accordingly, using a text editor.

[Appendix \(http://developer.novell.com/ndk/doc/ucs/ucs\\_\\_enu/data/a8shmq7.html#a8shmq7\)](http://developer.novell.com/ndk/doc/ucs/ucs__enu/data/a8shmq7.html#a8shmq7) in UCS documentation provides more samples on using UCS in Perl.

---



# Functions and Extensions

# 2

This chapter lists Perl functions and extensions applicable to the NetWare environment.

The latest Perl Specification can be got from [distributed Perl documentation page \(http://www.perl.com/CPAN-local/doc/manual/html/index.html\)](http://www.perl.com/CPAN-local/doc/manual/html/index.html).

## 2.1 Supported Functions and Extensions

All supported functions and extensions shipping with standard distribution work for Perl 5.8 on NetWare.

### 2.1.1 Remarks

- ♦ The built-in truncate procedure does not function reliably.
- ♦ Because NetWare does not support symbolic links, the built-in readlink procedure always returns the undefined value.
- ♦ NetWare does support the built-in system procedure. However, the shell it invokes is the standard NetWare console, which does not support pipes, I/O redirection, variable expansion, or filename globbing.

## 2.2 Unsupported Functions and Extensions

Some of the standard Perl functions and extensions are not available in the NetWare implementation. Each function listed below is either unavailable or inapplicable in the NetWare environment.

### 2.2.1 Standard Unsupported Functions and Extensions

---

chown	getpgrp	setgrent
chroot	getppid	setpgrp
endgrent	getpriority	setpriority
endpwent	getpwent	setpwent
fork	getpwnam	symlink
getgrent	getpwuid	syscall
getgrgid	link	teldir
getgrnam	rewinddir	times

---

## 2.2.2 System V Unsupported Functions

---

msgctl	shmctl	semctl
msgget	shmget	semget
msgrcv	shmread	semop
msgsnd	shmwrite	

---

# Perl\_LDAP Modules

# 3

Perl\_ldap modules provide programmers with an interface to perform complex operations on LDAP directories with minimal Perl code.

The existing Perl\_LDAP module available from CPAN can perform only the standard LDAP operations and does not support operations specific to Novell<sup>®</sup> eDirectory<sup>™</sup> extensions. The Perl\_LDAP modules for eDirectory extensions provide easy interfaces to access the eDirectory extensions such as [AbortPartitionOperationRequest](http://developer.novell.com/ndk/doc/jldap/jldapenu/api/com/novell/ldap/extensions/AbortPartitionOperationRequest.html) (<http://developer.novell.com/ndk/doc/jldap/jldapenu/api/com/novell/ldap/extensions/AbortPartitionOperationRequest.html>) and [GetReplicaInfoResponse](http://developer.novell.com/ndk/doc/jldap/jldapenu/api/com/novell/ldap/extensions/GetReplicaInfoResponse.html) (<http://developer.novell.com/ndk/doc/jldap/jldapenu/api/com/novell/ldap/extensions/GetReplicaInfoResponse.html>). For more details about the extensions provided by Perl\_LDAP modules see, [eDirectory extensions for Perl\\_LDAP module \(../perl\\_ldap/edirectory.html\)](http://developer.novell.com/ndk/doc/jldap/jldapenu/api/com/novell/ldap/extensions/GetReplicaInfoResponse.html).



# Perl Links

# 4

The popularity of Perl has given rise to an abundance of information about programming with Perl and using it with various products. The authoritative Perl text is [Programming Perl](http://www.oreilly.com/catalog/ppperl3) (<http://www.oreilly.com/catalog/ppperl3>) by Larry Wall, Tom Christiansen & Jon Orwant. Links to other resources are provided in the Web Pages section.

## 4.1 Web Pages

1. [The Perl Home Page](http://www.perl.com) (<http://www.perl.com>)
2. [What's new in Perl 5.8](http://www.perl.com/pub/2000/04/whatsnew.html) (<http://www.perl.com/pub/2000/04/whatsnew.html>)
3. [The Perl Manual](http://www.mit.edu/perl/perl.html) (<http://www.mit.edu/perl/perl.html>)
4. [The Comprehensive Perl Archive Network \(CPAN\) Page](http://www.perl.com/CPAN-local/modules/index.html) (<http://www.perl.com/CPAN-local/modules/index.html>)
5. [CGI Made Really Easy](http://www.jmarshall.com/easy/cgi) (<http://www.jmarshall.com/easy/cgi>)
6. [Perl/CGI Resources and Scripts](http://www.speakeasy.org/~cgires) (<http://www.speakeasy.org/~cgires>)

## 4.2 Online Guides

[Johan Vromans' Perl 5 Reference guide transcribed to HTML by Rex Swain](http://www.rexswain.com/perl5.html) (<http://www.rexswain.com/perl5.html>)

[The Perl Reference Guide](http://www.squirrel.nl/pub/perlref-5.004.1.pdf) (<http://www.squirrel.nl/pub/perlref-5.004.1.pdf>) by Johan Vromans

## 4.3 Publishers' Web Pages

1. [Learning Perl](http://www.ora.com/catalog/lperl2) (<http://www.ora.com/catalog/lperl2>) by Randal L. Schwartz
2. [Perl Cookbook: Tips and Tricks for Perl Programmers](http://www.ora.com/catalog/cookbook/noframes.html) (<http://www.ora.com/catalog/cookbook/noframes.html>) by Tom Christiansen & Nathan Torkington
3. [The Perl Reference Guide](http://www.squirrel.nl/people/jvromans/perlref.html) (<http://www.squirrel.nl/people/jvromans/perlref.html>) by Johan Vromans
4. [Accessing Novell Services from Perl on NetWare](http://developer.novell.com/research/appnotes/2000/october/07/a001007.htm) (<http://developer.novell.com/research/appnotes/2000/october/07/a001007.htm>)
5. [Programming on Netware made easy with perl scripting](http://developer.novell.com/research/devnotes/1999/november/06/index.htm) (<http://developer.novell.com/research/devnotes/1999/november/06/index.htm>)





# Sample mod\_perl Configuration File



The following code snippet represents a sample mod\_perl configuration (mod\_perl) file.

## A.1 mod\_perl Configuration File

```
LoadModule perl_module modules/mod_perl.nlm

Alias /perl/ "sys:/perl/web/"

<Location /perl>
    SetHandler perl-script
    PerlResponseHandler ModPerl::Registry
    Options +ExecCGI
    PerlOptions +ParseHeaders
</Location>
PerlModule Apache::Hello

<Location /hello>
    SetHandler modperl
    PerlHandler Apache::hello
</Location>
```

## A.2 Description of Configuration File

**LoadModule perl\_module modules/mod\_perl.nlm:** Loads Perl module.

**Alias /perl/ "sys:/perl/web":** Searches for scripts with Perl as url prefix in sys:\perl\web directory.

**Location /perl:** Assigns the rules to be followed by the server when the URL of the requested location matches the location.

**Options +ExecCGI:** Informs the server that the file is a program and should be executed, instead of being displayed as a static file.

**PerlOptions +ParseHeaders:** Instructs mod\_perl to use ModPerl::Registry Perl module for the actual content generation.

**SetHandler perl-script:** Assigns the mod\_perl Apache module to handle the content generation phase.



# mod\_perl Sample

# B

The following sample displays the string *mod\_perl-2.0 is working*.

```
package Apache::Hello;
use strict;
use Apache::RequestRec (); # for $r->content_type
use Apache::RequestIO (); # for $r->puts
use Apache::Const -compile => ':common';
sub handler {
    my $r = shift;
    my $time = scalar localtime();
    my $package = __PACKAGE__;
    $r->content_type('text/html');
    $r->puts("mod_perl-2.0 is working");
    return Apache::OK;
}
1;
```



# Secure Scripting

# C

This section covers an introduction to the security features, the support for remote file systems, configuration details, the changes required in scripts, and the limitations.

## C.1 Introduction

Secure scripting environment is provided for NetWare with the implementation of various security features in Novell Script for NetWare, UCX components, and Perl 5.6 for NetWare.

The security features are implemented in such a way that it preserves the power of script execution from the server console, and it ensures that the Web-based scripts (server-side) can access only resources for which the user has appropriate rights.

The security feature uses Novell eDirectory. The Web-based scripts should now authenticate to eDirectory with appropriate rights for various operations.

## C.2 Script Operations and Security

The following table lists the various operations and the role of security.

Operations	Security Feature
<b>Local file system access</b> - Directory listing, opening files for reading/writing, creation, deletion, moving of directories and files, etc.	Operations are allowed based on the rights of the user.  If the script does not log in, these operations will return an error.
<b>Local file system</b> - Above file system activities under web document directory, such as SYS:PERLWEB.	These operations are allowed based on the rights of the user.  If the script does not log in (public script), read-only access is allowed to the Web documents.
<b>Remote file system access</b> - Directory listing, opening files for reading/writing, creation, deletion, moving of directories and files, etc.	These operations are allowed based on the rights of the user.
<b>Server management</b> - Access to server configuration parameters, loading modules and NCF files, bringing down the server, access to server hardware, and peripherals	Allowed to admin or console operator.
<b>Import functions</b> - Calling functions exported by NLMs	These function calls are performed under the context of the user. Based on the user rights, the call either succeeds or fails.

## C.3 Remote File System Support

Security feature is provided for the remote NetWare file system. Existing components are modified to support it. Scripts can access the file systems on other servers, by logging into eDirectory using UCX:NWDir component and the FileSystem object.

The resources on the remote NetWare file system are identified using the Universal Naming Convention (UNC). The strings in the UNC format have to be used as arguments for the method/property call, to access the remote file/directory.

Following is the syntax and examples.

Syntax 1: `\\ServerName\Volume:Path\name`

Ex: `\\My-Test-Server\DATA\User\mydata\account.txt`

Syntax 2: `\\IP Address\Volume:Path\name`

Ex: `\\1.2.3.4\DATA\User\mydata\account.txt`

Following example adds a folder to a remote NetWare volume and displays the contents of the directory.

```

# Script to create a directory on remote NetWare volume
# Load Perl2UCS or UCSExt based on the Perl version

if ($] < 5.006) {
    require UCSExt;
} else {
    require Perl2UCS;
}

if ($] < 5.006) {
    $fso = UCSExt->new("UCX:FileSystemObject") or die "Failed to
instantiate the File System Object component\n";
} else {
    $fso = Perl2UCS->new("UCX:FileSystemObject") or die "Failed to
instantiate the File System Object component\n";
}

if ($] < 5.006)
{
    $nwdir = UCSExt->new("UCX:NWDIR") or die "Failed to instantiate the
Directory component\n";
}
else
{
    $nwdir = Perl2UCS->new("UCX:NWDIR") or die "Failed to instantiate the
Directory component\n";
}
$nwdir ->{"FullName"} = "nds:\\\\remote_tree\\remote_context";
$nwdir ->login("UserName", "Password") or die "Failed to login. Check the
username and password\n";

$myFolder = $fso->GetFolder("\\\\remoteserver\\sys:\\tmp");

$subFolders = $myFolder->{"SubFolders"};

$subFolders->reset();

print "Remote Folder List: Prior to add\n";

while($subFolders->hasMoreElements())
{
    $sub_folder = $subFolders->Next();
    $FolderName = $sub_folder->{"Name"};
    print "The Folders Name is $FolderName" . "\n";
}

print "\n";

print "Remote Folder List: After adding \n";

$newfld=$subFolders->Add("Test");
$subFolders = $myFolder->{"SubFolders"};
$subFolders->reset();

while($subFolders->hasMoreElements())
{
    $sub_folder = $subFolders->Next();
    $FolderName = $sub_folder->{"Name"};
    print "The Folders Name is $FolderName" . "\n";
}

```

```
}  
  
$nwdir->Logout ();
```

## C.4 Configuration

By default, the security features are enabled on the system. However, it is an optional feature and you can choose to enable or disable it. Edit the `sys:\system\nwsec.ini` file to enable or disable the security features.

For example, to disable the security features for Perl, modify the `sys:\system\nwsec.ini`, to include the following line under the section [Languages].

```
PERL=OFF
```

---

**NOTE:** If the security features are turned off, public access is granted to all the system resources. The decision to disable this feature has to be taken only after carefully considering the consequences.

---

## C.5 Changes Required in the Scripting Applications

The changes required in the scripts with the introduction of the security features are minimal.

The following table lists the changes for the scripts.

Script Type	Changes
<b>Console-based scripts</b> - Scripts that are executed from the server console.	No changes.  Include script code to login to eDirectory using UCX:NWDIR object if access to remote file system is required.
<b>Web-based scripts</b> - Scripts that are executed as an LCGI program, or scripts embedded in HTML pages such as NSP or ASP.	Security features ensure that the users are allowed to perform operations for which they have the relevant rights.  For example, a script may not be able to list the content of the SYS volume, until the user is authenticated to eDirectory with appropriate rights.  Include script code to login to eDirectory using UCX:NWDIR object. Otherwise scripts have restricted access to resources.

Following is a simple Perl example to demonstrate the need for the script to authenticate with appropriate rights.

An existing Web script that is executed without any rights returns an error.



```

# Script to list the contents of SYS:NSN directory
# Load Perl2UCS or UCSExt based on the Perl version
use CGI;
$cgiobject=new CGI;
print $cgiobject->header;

if ($] < 5.006) {
    require UCSExt;
} else {
    require Perl2UCS;
}
if ($] < 5.006) {
    $fso = UCSExt->new("UCX:FileSystemObject") or die "Failed to
instantiate the File System Object component\n";
} else {
    $fso = Perl2UCS->new("UCX:FileSystemObject") or die "Failed to
instantiate the File System Object component\n";
}

print "<html>";
print "<BODY TEXT=\"#000000\" LINK=\"#FF0000\">";
print "<BODY BACKGROUND=/images/blue_pap.gif>";
print "<FONT FACE=ARIAL>";
print "<html><head><title>Directory Names</title></head>\n";
print "<body><h1>List of Folders</h1>";

$myFolder = $fso->GetFolder("sys:\\nsn");
$subFolders = $myFolder->{"SubFolders"};

$subFolders->reset();

while($subFolders->hasMoreElements())
{
    $sub_folder = $subFolders->Next();
    $FolderName = $sub_folder->{"Name"};
    print "$FolderName" . "\n";
    print "<br>";
}

```

The modified Web script does not throw an error as it logs into eDirectory with appropriate rights.

```

# Script to list the contents of SYS:NSN directory
# Load Perl2UCS or UCSExt based on the Perl version
use CGI;
$cgiobject=new CGI;
print $cgiobject->header;

if ($] < 5.006) {
    require UCSExt;
} else {
    require Perl2UCS;
}
if ($] < 5.006) {
    $fso = UCSExt->new("UCX:FileSystemObject") or die "Failed to
instantiate the File System Object component\n";
} else {
    $fso = Perl2UCS->new("UCX:FileSystemObject") or die "Failed to
instantiate the File System Object component\n";
}
if ($] < 5.006)
{
    $nwdir = UCSExt->new("UCX:NWDIR") or die "Failed to instantiate the
Directory component\n";
}
else
{
    $nwdir = Perl2UCS->new("UCX:NWDIR") or die "Failed to instantiate the
Directory component\n";
}
print "<html>";
print "<BODY TEXT=\`#\#000000\` LINK=\`#\#FF0000\`>";
print "<BODY BACKGROUND=/images/blue_pap.gif>";
print "<FONT FACE=ARIAL>";
print "<html><head><title>Directory Names</title></head>\n";
print "<body><h1>List of Folders</h1>";
$nwdir ->{"FullName"} = "nds:\\\\servername\\context";
$nwdir ->login("UserName", "PassWord") or die "Failed to login. Check the
username and password\n";

$myFolder = $fso->GetFolder("sys:\\nsn");
$subFolders = $myFolder->{"SubFolders"};

$subFolders->reset();

while($subFolders->hasMoreElements())
{
    $sub_folder = $subFolders->Next();
    $FolderName = $sub_folder->{"Name"};
    print "$FolderName" . "\n";
    print "<br>";
}
$nwdir->Logout();

```

## C.6 Limitations

Multiple tree support is not implemented for remote file system access. Once the script logs into an eDirectory tree, the script can access the files on any server within this tree. Simultaneous authentication to multiple trees, and accessing resources on different servers are not supported. If

there is a need to access different servers, it has to be done in a serial fashion. For example, log in to the tree A, access resources on server A1, and logout; again log in to tree B, access resources on server B1, log out, and so on.



# Specifications for Perl for NetWare in CPAN Site

# D

1. The source code for the NetWare<sup>®</sup> specific modules Perl2UCS and CGI2Perl are not available in the [CPAN site \(http://www.cpan.org/src/README.html\)](http://www.cpan.org/src/README.html). The binaries (ucscgi58.exe) for these modules can be downloaded from the [NDK support site \(http://developer.novell.com/wiki/index.php/TID102598\\_%28ucscgi58%29\\_Perl2UCS\\_%26\\_CGI2Perl\\_for\\_Perl\\_5.8.0\)](http://developer.novell.com/wiki/index.php/TID102598_%28ucscgi58%29_Perl2UCS_%26_CGI2Perl_for_Perl_5.8.0).

The downloaded zip file should be extracted to the SYS volume, after building Perl using the open source available in the CPAN site

2. Security features are not enabled for the open source available in the CPAN site.
3. NetWare specific Perl samples are not available in the CPAN site. The samples can be downloaded along with the Perl binaries from the [Perl home page \(http://developer.novell.com/ndk/perl5.htm\)](http://developer.novell.com/ndk/perl5.htm) in the NDK site.
4. External Perl modules can be built based on the information provided in “[Building Perl Extensions](#)” on page 47.

The information for building Perl modules is also available in the file readme.netware that is present with the NetWare sources in the CPAN site.



# Building Perl Extensions

# E

This section gives instructions for building Perl extensions for NetWare.

Complete the following steps to build Perl NLM™ (NetWare Loadable Module) and other associated modules.

- ♦ Meet the Tools and SDK requirements. See, [Section E.1, “Software Requirements,”](#) on page 47
- ♦ Build Perl NLM. See, [Section E.2, “Building Perl NLM,”](#) on page 47
- ♦ Install Perl on NetWare. See, [Section E.3, “Installing Perl on NetWare,”](#) on page 48
- ♦ Build Perl extensions. See, [Section E.4, “Building Perl Extensions,”](#) on page 49

## E.1 Software Requirements

Ensure to satisfy the following requirements before building Perl NLMs.

- CodeWarrior compiler and linker
- Microsoft\* Visual C++ version 4.2 or later
- NetWare SDK. This can be downloaded from the [NDK site \(http://developer.novell.com/ndk/downloadaz.htm\)](http://developer.novell.com/ndk/downloadaz.htm)
- NLM & NetWare Libraries for C. This can be downloaded from the [NDK site \(http://developer.novell.com/ndk/clib.htm\)](http://developer.novell.com/ndk/clib.htm)
- Server Protocol Libraries for C. This can be downloaded from the [NDK site \(http://developer.novell.com/ndk/nwprotlb.htm\)](http://developer.novell.com/ndk/nwprotlb.htm)

## E.2 Building Perl NLM

Complete the following steps to build Perl NLM:

- ♦ Setup the environment for building Perl NLM. See, [“Setting up the Environment for Building Perl NLM”](#) on page 47
- ♦ Perform the make process. See, [“Make Process”](#) on page 48
- ♦ Build Perl Interpreter. See, [“Building Perl Interpreter”](#) on page 48

### E.2.1 Setting up the Environment for Building Perl NLM

The build process is dependent on the location of the NetWare SDK. Once the Tools & SDK are installed, the build environment has to be setup. The following batch files setup the environment.

- ♦ setnwbl.bat - The Execution of this file requires the path to the NetWare sdk directory and the path to CodeWarrior Compiler & tools directory, as the input parameters. This file sets these paths and also sets the build type to Release by default.
- ♦ Buildtype.bat - This is used to set the build type to debug or release. Change the build type only after executing setnwbl.bat

Enter `buildtype d on` at the command prompt to set the buildtype to Debug type with D2 flag set.

Enter `buildtype d off` or `buildtype d` at the command prompt to set the buildtype to Debug type with D1 flag set.

Enter `buildtype r` at the command prompt to set the build type to Release Build type.

## E.2.2 Make Process

The make process runs only under Windows\* NT shell. The NetWare makefile located under the NetWare folder makes use of `miniperl.exe` to run some of the Perl scripts.

The make process creates the Perl extensions as `I<<Extension>.nlm>`.

To create `miniperl.exe`:

- 1 Set the required paths for Visual c++ compiler (specify `vcvars32` location) at the command prompt.
- 2 Execute `nmake` from `win32` folder through Windows NT command prompt.
- 3 You can stop the build process after `miniperl.exe` is created.

## E.2.3 Building Perl Interpreter

After creating `miniperl.exe`, run `nmake` from the NetWare folder. This will build the Perl interpreter for NetWare as `I<perl.nlm>`. The interpreter is copied under the `I<Release>` folder for release builds and `I<Debug>` folder for debug builds.

To build Perl Interpreter:

- 1 Execute `nmake` from NetWare folder through Windows NT command prompt.

Currently, the following build types are tested on NetWare:

- ♦ `USE_MULTI`, `USE_ITHREADS` and `USE_IMP_SYS` defined
- ♦ `USE_MULTI` and `USE_IMP_SYS` defined, and `USE_ITHREADS` not defined

## E.3 Installing Perl on NetWare

To install NetWare Perl on a NetWare server:

- 1 Map the SYS volume of a NetWare server to `I<i:>`, since the makefile by default sets the drive letter to `I<i:>`.
- 2 Enter `I<nmake nwininstall>` from the NetWare folder on the Windows NT command prompt. This will copy the binaries and module files to the NetWare server under `I<sys:\perl>` folder, and the Perl interpreter, `I<perl.nlm>`, to `I<sys:\perl\system>` folder.
- 3 Move the Perl interpreter to `sys:\system` folder.
- 4 Enter `nmake install`, to install the binaries on the local machine, under `c:\perl` folder.



## E.4 Building Perl Extensions

Perl for NetWare has to be installed on Windows along with Perl for Windows, to build extensions other than standard extensions. The sources for Perl for Windows can be downloaded from the [CPAN site \(http://www.cpan.org\)](http://www.cpan.org) and built and installed in the c:\ drive or, or can be directly downloaded from the [ActiveState site \(http://www.activestate.com\)](http://www.activestate.com).

Perl can be installed by typing `I<nmake install>` on a Windows NT command prompt after building Perl for NetWare by following steps given in [Section E.2, “Building Perl NLM,” on page 47](#). This will copy all the \*.pm files and other required files.

---

**NOTE:** Perl for Windows should be installed before installing Perl for NetWare.

---

To build extensions other than standard extensions:

- 1 Set the current directory to the extension directory that contains the source files.
- 2 Execute the following command at the command prompt.

```
perl -II<path_to_NetWare_lib_dir> -II<path_to_lib> Makefile.pl
```

For example, if Perl 5.6.1 binaries are installed the command executed will be

```
perl -Ic:/perl/5.6.1/lib/NetWare-x86-multi-thread -Ic:\perl\5.6.1\lib  
MakeFile.pl
```

If Perl 5.8 binaries are installed the command executed will be

```
perl -Ic:/perl/5.8.0/lib/NetWare-x86-multi-thread -Ic:\perl\5.8.0\lib  
MakeFile.pl
```

- 3 Execute `nmake` from Windows NT command prompt, to build the Perl extensions.
- 4 Execute `nmake install` from the Windows NT command prompt, to copy the libraries and the binaries of the built extension to the Windows machine in which Perl for NetWare is installed.

In this case, the libraries should be copied to the NetWare server, manually.

To copy libraries and binaries to the NetWare server, automatically:

- 1 The libraries can be copied to the NetWare server, automatically by executing the following commands at the Windows NT command prompt, as part of [Step 2 on page 49](#):

```
perl -II<path_to_NetWare_lib_dir> -II<path_to_lib> Makefile.pl
```

```
INSTALLSITELIB=i:\perl\lib
```

For example, if Perl 5.6.1 binaries are installed the command executed will be

```
perl -Ic:/perl/5.6.1/lib/NetWare-x86-multi-thread -Ic:\perl\5.6.1\lib  
MakeFile.pl
```

```
INSTALLSITELIB=i:\perl\lib
```

If Perl 5.8 binaries are installed the command executed will be

```
perl -Ic:/perl/5.8.0/lib/NetWare-x86-multi-thread -Ic:\perl\5.8.0\lib  
MakeFile.pl
```

```
INSTALLSITELIB=i:\perl\lib
```

- 2 Repeat [Step 3 on page 49](#) and [Step 4 on page 49](#).

---

**NOTE:** Perl modules downloaded from CPAN may require NetWare related API changes in the source, in order to build on NetWare.

---

# Revision History

# F

The following table outlines all the changes that have been made to this documentation (in reverse chronological order):

## F.1 June 2006

- ♦ Updated the Trademarks list to comply with revised Novell documentation standards

## F.2 October 2003

- ♦ Modified the version of Apache to 2.0.47 in [Table 1-2 on page 11](#) in [Chapter 1, “Getting Started,” on page 11](#).

## F.3 March 2003

- ♦ Added information about [“mod\\_perl 2.0 for NetWare” on page 7](#) in Preface.
- ♦ Included the [Table 1-2 on page 11](#) in the [Chapter 1, “Getting Started,” on page 11](#).
- ♦ Provided details about configuring Apache 2.0 Web server for Perl and mod\_perl.
- ♦ Included a [“Sample mod\\_perl Configuration File” on page 33](#).
- ♦ Included a [“mod\\_perl Sample” on page 35](#).

## F.4 November 2002

- ♦ Included information about [“DBD::MYSQL” on page 9](#)
- ♦ Included information about [Section 1.11, “Accessing Pervasive from Perl,” on page 20](#)
- ♦ Included information about [Section 1.12, “Accessing MySQL from Perl,” on page 20](#)

## F.5 September 2002

- ♦ Included information about [“Perl\\_LDAP Modules” on page 29](#)
- ♦ Included information about [“Building Perl Extensions” on page 47](#)

## F.6 May 2002

- ♦ Included [“Specifications for Perl for NetWare in CPAN Site” on page 45](#)
- ♦ Modified configuration script to [Section 1.2, “Configure Apache 2.0 Web Server,” on page 11](#)
- ♦ Modified the configuration command to [Section 1.4, “Configure NetWare Enterprise Web Server,” on page 12](#)

## **F.7 February 2002**

- ♦ Added [Appendix C, “Secure Scripting,”](#) on page 37
- ♦ Updated [Section 1.7, “Executing Perl Scripts from the System Console,”](#) on page 18 section

## **F.8 June 2001**

- ♦ This is the first edition of the book