

# Novell Developer Kit

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

June 2006

LAN TEST TOOLS



**Novell**<sup>®</sup>

## 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.

Any products or technical information provided under this Agreement may be subject to U.S. export controls and the trade laws of other countries. You agree to comply with all export control regulations and to obtain any required licenses or classification to export, re-export, or import deliverables. You agree not to export or re-export to entities on the current U.S. export exclusion lists or to any embargoed or terrorist countries as specified in the U.S. export laws. You agree to not use deliverables for prohibited nuclear, missile, or chemical biological weaponry end uses. Please refer to [www.novell.com/info/exports/](http://www.novell.com/info/exports/) for more information on exporting Novell software. Novell assumes no responsibility for your failure to obtain any necessary export approvals.

All files provided in this release are subject to the Novell Developer License Agreement, which can be found in the license.txt file provided in this download.

Copyright © 1993-2005 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.

Novell, Inc. has intellectual property rights relating to technology embodied in the product that is described in this document. In particular, and without limitation, these intellectual property rights may include one or more of the U.S. patents listed at <http://www.novell.com/company/legal/patents/> and one or more additional patents or pending patent applications in the U.S. and in other countries.

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

AppNotes is a registered trademark of Novell, Inc.

AppTester is a registered trademark of Novell, Inc. in the United States.

ASM is a trademark of Novell, Inc.

Beagle is a trademark of Novell, Inc.

BorderManager is a registered trademark of Novell, Inc.

BrainShare is a registered service mark of Novell, Inc. in the United States and other countries.

C3PO is a trademark of Novell, Inc.

Certified Novell Engineer is a service mark of Novell, Inc.

Client32 is a trademark of Novell, Inc.

CNE is a registered service mark of Novell, Inc.

ConsoleOne is a registered trademark of Novell, Inc.

Controlled Access Printer is a trademark of Novell, Inc.

Custom 3rd-Party Object is a trademark of Novell, Inc.

DeveloperNet is a registered trademark of Novell, Inc., in the United States and other countries.

DirXML is a registered trademark of Novell, Inc.

eDirectory is a trademark of Novell, Inc.

Exceleator is a trademark of Novell, Inc.

exteNd is a trademark of Novell, Inc.

exteNd Director is a trademark of Novell, Inc.

exteNd Workbench is a trademark of Novell, Inc.

FAN-OUT FAILOVER is a trademark of Novell, Inc.

GroupWise is a registered trademark of Novell, Inc., in the United States and other countries.

Hardware Specific Module is a trademark of Novell, Inc.

Hot Fix is a trademark of Novell, Inc.

Hula is a trademark of Novell, Inc.

iChain is a registered trademark of Novell, Inc.

Internetwork Packet Exchange is a trademark of Novell, Inc.

IPX is a trademark of Novell, Inc.

IPX/SPX is a trademark of Novell, Inc.

jBroker is a trademark of Novell, Inc.

Link Support Layer is a trademark of Novell, Inc.

LSL is a trademark of Novell, Inc.

ManageWise is a registered trademark of Novell, Inc., in the United States and other countries.

Mirrored Server Link is a trademark of Novell, Inc.

Mono is a registered trademark of Novell, Inc.

MSL is a trademark of Novell, Inc.

My World is a registered trademark of Novell, Inc., in the United States.

NCP is a trademark of Novell, Inc.

NDPS is a registered trademark of Novell, Inc.

NDS is a registered trademark of Novell, Inc., in the United States and other countries.

NDS Manager is a trademark of Novell, Inc.

NE2000 is a trademark of Novell, Inc.

NetMail is a registered trademark of Novell, Inc.

NetWare is a registered trademark of Novell, Inc., in the United States and other countries.

NetWare/IP is a trademark of Novell, Inc.

NetWare Core Protocol is a trademark of Novell, Inc.  
NetWare Loadable Module is a trademark of Novell, Inc.  
NetWare Management Portal is a trademark of Novell, Inc.  
NetWare Name Service is a trademark of Novell, Inc.  
NetWare Peripheral Architecture is a trademark of Novell, Inc.  
NetWare Requester is a trademark of Novell, Inc.  
NetWare SFT and NetWare SFT III are trademarks of Novell, Inc.  
NetWare SQL is a trademark of Novell, Inc.  
NetWare is a registered service mark of Novell, Inc., in the United States and other countries.  
NLM is a trademark of Novell, Inc.  
NMAS is a trademark of Novell, Inc.  
NMS is a trademark of Novell, Inc.  
Novell is a registered trademark of Novell, Inc., in the United States and other countries.  
Novell Application Launcher is a trademark of Novell, Inc.  
Novell Authorized Service Center is a service mark of Novell, Inc.  
Novell Certificate Server is a trademark of Novell, Inc.  
Novell Client is a trademark of Novell, Inc.  
Novell Cluster Services is a trademark of Novell, Inc.  
Novell Directory Services is a registered trademark of Novell, Inc.  
Novell Distributed Print Services is a trademark of Novell, Inc.  
Novell iFolder is a registered trademark of Novell, Inc.  
Novell Labs is a trademark of Novell, Inc.  
Novell SecretStore is a registered trademark of Novell, Inc.  
Novell Security Attributes is a trademark of Novell, Inc.  
Novell Storage Services is a trademark of Novell, Inc.  
Novell, Yes, Tested & Approved logo is a trademark of Novell, Inc.  
Nsure is a registered trademark of Novell, Inc.  
Nterprise is a registered trademark of Novell, Inc., in the United States.  
Nterprise Branch Office is a trademark of Novell, Inc.  
ODI is a trademark of Novell, Inc.  
Open Data-Link Interface is a trademark of Novell, Inc.  
Packet Burst is a trademark of Novell, Inc.  
PartnerNet is a registered service mark of Novell, Inc., in the United States and other countries.  
Printer Agent is a trademark of Novell, Inc.  
QuickFinder is a trademark of Novell, Inc.  
Red Box is a trademark of Novell, Inc.  
Red Carpet is a registered trademark of Novell, Inc., in the United States and other countries.  
Sequenced Packet Exchange is a trademark of Novell, Inc.  
SFT and SFT III are trademarks of Novell, Inc.  
SPX is a trademark of Novell, Inc.  
Storage Management Services is a trademark of Novell, Inc.  
SUSE is a registered trademark of Novell, Inc., in the United States and other countries.  
System V is a trademark of Novell, Inc.  
Topology Specific Module is a trademark of Novell, Inc.  
Transaction Tracking System is a trademark of Novell, Inc.  
TSM is a trademark of Novell, Inc.

TTS is a trademark of Novell, Inc.

Universal Component System is a registered trademark of Novell, Inc.

Virtual Loadable Module is a trademark of Novell, Inc.

VLM is a trademark of Novell, Inc.

Yes Certified is a trademark of Novell, Inc.

ZENworks is a registered trademark of Novell, Inc., in the United States and other countries.

### **Third-Party Materials**

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



# Contents

<b>About This Guide</b>	<b>11</b>
<b>Part I LAN Certification for SUSE LINUX</b>	<b>13</b>
<b>1 Introduction to the LAN Test Tools for SUSE LINUX</b>	<b>15</b>
1.1 Additional Testing Suggestions	15
1.2 Support	15
<b>2 Setting Up the Test Environment</b>	<b>17</b>
2.1 Reference Architecture Requirements	17
2.2 Software Requirements	17
2.3 Virtual Environments	17
2.4 Uninstalling the Test Tools	18
2.5 Installing the Test Tools	18
2.6 Setting Up	18
2.7 IP Addresses	20
2.8 Setting the Device Activation Mode	20
2.9 Troubleshooting	21
<b>3 Using TestConsole</b>	<b>23</b>
3.1 TestConsole Overview	23
3.1.1 Project Contents	24
3.1.2 Run Queue	25
3.1.3 Project Log	25
3.2 Managing Test Projects	26
3.2.1 Opening a New Test Project	26
3.2.2 Opening an Existing Test Project	26
3.2.3 Saving a Test Project	26
3.2.4 Closing a Test Project	26
3.3 Testing in Debug Mode	26
3.4 Using Loop Mode	27
3.4.1 Editing the Loop Test List	27
3.4.2 Running the Loop Test List	28
3.5 Reporting Test Results	28
3.5.1 Entering Product Information	28
3.5.2 Importing Product Information	29
3.5.3 Verifying Report Data	30
3.5.4 Generating a Novell Report	32
3.5.5 Viewing or Printing a Summary	32
3.5.6 Clearing a Report	33
<b>4 Testing Your Adapter</b>	<b>35</b>
4.1 Running the Tests	35

<b>5</b>	<b>Testing in a Virtual Environment</b>	<b>37</b>
5.1	Installing XEN Virtual Machine Host Server . . . . .	37
5.2	Creating the Virtual Machines . . . . .	37
5.3	Running the Tests . . . . .	38
<b>6</b>	<b>Submitting Your Test Results</b>	<b>39</b>
6.1	Additional Information . . . . .	39
6.2	Enter Product & Company Information . . . . .	39
6.3	Driver and Module Information . . . . .	40
6.4	Submit Results . . . . .	40
6.5	Provide Explanations for Test Failures . . . . .	40
	<b>Part II ODI and CIOS LAN Driver Certification for NetWare</b>	<b>43</b>
<b>7</b>	<b>Introduction to the LAN Tests</b>	<b>45</b>
7.1	Multi-Mode and Multi-speed Test Requirements . . . . .	45
7.2	Additional Testing Suggestions . . . . .	45
7.3	Support . . . . .	45
<b>8</b>	<b>Setting Up the Test Environment</b>	<b>47</b>
8.1	Hardware Requirements . . . . .	47
8.2	Software Requirements . . . . .	48
8.2.1	Stimulus Driver (Ethernet only) . . . . .	48
8.3	Setting Up . . . . .	49
8.3.1	Loading Frame Types . . . . .	50
8.3.2	IP Addresses . . . . .	51
8.3.3	Sample Autoexec.ncf . . . . .	51
<b>9</b>	<b>Running the Tests</b>	<b>55</b>
9.1	A Note on Testing PC Card Adapter and Drivers . . . . .	55
9.2	Running the LAN Tests . . . . .	55
9.3	Test Execution . . . . .	57
9.3.1	Automated Tests . . . . .	57
9.3.2	Manual Tests . . . . .	58
9.3.3	Optional Test . . . . .	59
9.4	Test Parameters . . . . .	59
9.5	Configuration Table . . . . .	60
9.6	Statistics Table . . . . .	60
9.7	Exiting and Saving Test Results . . . . .	61
9.8	Test Results . . . . .	61
9.9	Debug Mode . . . . .	62
9.10	Changing Test Suites . . . . .	62
<b>10</b>	<b>Submitting Your Test Results</b>	<b>65</b>
10.1	Additional Information . . . . .	65
10.2	Enter Product & Company Information . . . . .	65
10.3	Driver and Module Information . . . . .	66



10.4	NetWare Server Information . . . . .	66
10.5	Submit Results. . . . .	66
10.6	Provide Explanations for Test Failures . . . . .	66

**Part III Appendixes 69**

**A Certification Requirements 71**

A.1	General . . . . .	71
A.1.1	Req_LINUX_LAN_G001 . . . . .	71
A.1.2	Req_LINUX_LAN_G002 . . . . .	71
A.1.3	Req_LINUX_LAN_G003 . . . . .	71
A.1.4	Req_LINUX_LAN_G004 . . . . .	71
A.2	Documentation. . . . .	71
A.2.1	Req_LINUX_LAN_D001 . . . . .	72
A.2.2	Req_LINUX_LAN_D002 . . . . .	72
A.3	Installation Tests . . . . .	72
A.3.1	Req_LINUX_LAN_I001 . . . . .	72
A.3.2	Req_LINUX_LAN_I002 . . . . .	72
A.3.3	Req_LINUX_LAN_I003 . . . . .	72
A.3.4	Req_LINUX_LAN_I004 . . . . .	72
A.4	Operational Requirements. . . . .	72
A.4.1	Req_LINUX_LAN_O001 . . . . .	73
A.4.2	Req_LINUX_LAN_O002 . . . . .	73
A.4.3	Req_LINUX_LAN_O003 . . . . .	73
A.4.4	Req_LINUX_LAN_O004 . . . . .	73
A.4.5	Req_LINUX_LAN_O005 . . . . .	73
A.4.6	Req_LINUX_LAN_O006 . . . . .	73
A.4.7	Req_LINUX_LAN_O007 . . . . .	73
A.4.8	Req_LINUX_LAN_O009 . . . . .	73
A.4.9	Req_LINUX_LAN_O010 . . . . .	73
A.4.10	Req_LINUX_LAN_O011 . . . . .	73
A.4.11	Req_LINUX_LAN_O012 . . . . .	73
A.4.12	Req_LINUX_LAN_O013 . . . . .	74
A.4.13	Req_LINUX_LAN_O014 . . . . .	74
A.4.14	Req_LINUX_LAN_O015 . . . . .	74
A.4.15	Req_LINUX_LAN_O016 . . . . .	74
A.4.16	Req_LINUX_LAN_O017 . . . . .	74
A.4.17	Req_LINUX_LAN_O018 . . . . .	74
A.4.18	Req_LINUX_LAN_O019 . . . . .	74
A.4.19	Req_LINUX_LAN_O020 . . . . .	74
A.4.20	Req_LINUX_LAN_O021 . . . . .	74
A.4.21	Req_LINUX_LAN_O022 . . . . .	74
A.4.22	Req_LINUX_LAN_O023 . . . . .	74
A.4.23	Req_LINUX_LAN_O024 . . . . .	74
A.4.24	Req_LINUX_LAN_O025 . . . . .	75
A.4.25	Req_LINUX_LAN_O026 . . . . .	75
A.4.26	Req_LINUX_LAN_O027 . . . . .	75
A.4.27	Req_LINUX_LAN_O028 . . . . .	75
A.4.28	Req_LINUX_LAN_O029 . . . . .	75
A.4.29	Req_LINUX_LAN_O030 . . . . .	75
A.4.30	Req_LINUX_LAN_O031 . . . . .	75
A.4.31	Req_LINUX_LAN_O032 . . . . .	75
A.4.32	Req_LINUX_LAN_O033 . . . . .	75
A.4.33	Req_LINUX_LAN_O034 . . . . .	75
A.4.34	Req_LINUX_LAN_O035 . . . . .	75
A.4.35	Req_LINUX_LAN_O036 . . . . .	76

A.4.36	Req_LINUX_LAN_O037	76
A.4.37	Req_LINUX_LAN_O038	76
A.4.38	Req_LINUX_LAN_O039	76
A.4.39	Req_LINUX_LAN_O040	76
A.4.40	Req_LINUX_LAN_O041	76
A.4.41	Req_LINUX_LAN_O042	76
A.4.42	Req_LINUX_LAN_O043	76
A.4.43	Req_LINUX_LAN_O044	76
A.4.44	Req_LINUX_LAN_O045	76
A.4.45	Req_LINUX_LAN_O046	76
A.4.46	Req_LINUX_LAN_O047	77
A.4.47	Req_LINUX_LAN_O048	77
A.4.48	Req_LINUX_LAN_O049	77
A.4.49	Req_LINUX_LAN_O050	77
A.4.50	Req_LINUX_LAN_O051	77
A.4.51	Req_LINUX_LAN_O052	77
A.4.52	Req_LINUX_LAN_O053	78
A.4.53	Req_LINUX_LAN_O054	78
A.4.54	Req_LINUX_LAN_O055	78
A.4.55	Req_LINUX_LAN_O056	78

**B Test Descriptions 79**

B.1	Test Descriptions	79
B.1.1	Basic Transmit and Receive Tests	79
B.1.2	I/O Control (IOCTL) Tests	82
B.1.3	Traffic and Endurance Tests	88
B.1.4	Miscellaneous Tests	90
B.1.5	Manual Tests	92
B.1.6	Optional Test	93

**C Stimulus Driver Generation 95**

C.1	Generation Procedure for a 32-bit Driver	95
-----	--	----

**D Revision History 97**

# About This Guide

The test suites in the LAN Test Tools are required for Linux, 32-bit ODI, ATM (Fat), and CIOS LAN drivers to receive the Novell® YES CERTIFIED™ status and logo. We think you will find these new test tools easier to set up and faster to run. We hope you will also find them useful as development and testing tools.

This guide contains the following sections:

- “LAN Certification for SUSE LINUX” on page 13
- “ODI and CIOS LAN Driver Certification for NetWare” on page 43
- “Revision History” on page 97

## Audience

This guide is intended for LAN development engineers and testers on the NetWare® and Novell SUSE® Linux platforms.

## Feedback

We want to hear your comments and suggestions about this manual and the other documentation included with this product. Please use the User Comments feature at the bottom of each page of the online documentation.

## Documentation Updates

For the most recent version of this guide, see [ODI and CIOS LAN Test Tools \(http://developer.novell.com/wiki/index.php/LAN\\_Test\\_Tools\)](http://developer.novell.com/wiki/index.php/LAN_Test_Tools).

## Additional Information

For information about the LAN Driver certification, see [LAN Driver Development \(http://developer.novell.com/devres/lan\)](http://developer.novell.com/devres/lan).

For the developer support posts for the ODI and CIOS LAN Test Tools, see the related [Developer Support Forum \(http://developer.novell.com/ndk/devforums.htm\)](http://developer.novell.com/ndk/devforums.htm).

- [YES Certification on SUSE LINUX \(http://developer.novell.com/devnet/yes/\)](http://developer.novell.com/devnet/yes/)
- [Novell on Linux \(http://www.novell.com/linux\)](http://www.novell.com/linux)

## Documentation Conventions

In this documentation, a greater-than symbol (>) is used to separate actions within a step and items within a cross-reference path.

A trademark symbol (®, ™, etc.) denotes a Novell trademark. An asterisk (\*) denotes a third-party trademark.



# LAN Certification for SUSE LINUX

This part contains the test information and procedures for certifying your LAN driver for the Novell® SUSE® Linux platform. It contains the following sections:

- Chapter 1, “Introduction to the LAN Test Tools for SUSE LINUX,” on page 15
- Chapter 2, “Setting Up the Test Environment,” on page 17
- Chapter 3, “Using TestConsole,” on page 23
- Chapter 4, “Testing Your Adapter,” on page 35
- Chapter 5, “Testing in a Virtual Environment,” on page 37
- Chapter 6, “Submitting Your Test Results,” on page 39



# Introduction to the LAN Test Tools for SUSE LINUX

# 1

These tools provide a test environment to ensure LAN driver/adaptor compatibility with Linux, NetWare<sup>®</sup>, and other Novell<sup>®</sup> products. The test tools currently focus on Ethernet connectivity. Successful completion of these tests is required to obtain the YES CERTIFIED™ status and logo from Novell.

## 1.1 Additional Testing Suggestions

Although these tests are designed to ensure that drivers conform to the specification and meet minimum quality requirements set by Novell, we highly recommend that additional testing be done in order to improve driver quality and reliability. This should be done at your site prior to submission. Some of the additional testing can be done using all or parts of the LAN Test Tools. The others will require tests of your own design.

The following are some suggestions for additional testing you can do.

- Exercise all hardware configuration combinations on your adapter.
- Design and perform hardware diagnostic tests for your adapter (refer to the IEEE specification for ideas).
- Execute additional performance testing.
- Perform additional stress testing, such as using larger network configurations with heavy, varied traffic, over an extended period of time.
- See [Appendix A, “Certification Requirements,” on page 71](#) for criteria that you should consider in your testing.

---

**IMPORTANT:** Novell does not provide software or support for additional testing. Consult your driver developers for help with additional testing.

---

## 1.2 Support

The following support resources are available to assist you in your certification testing:

- [LAN group web page \(http://developer.novell.com/devres/lan/\)](http://developer.novell.com/devres/lan/)
- [Novell LAN Test Tools \(http://developer.novell.com/wiki/index.php/LAN\\_Test\\_Tools\)](http://developer.novell.com/wiki/index.php/LAN_Test_Tools)
- [Minimum Patch List page \(http://support.novell.com/produpdate/patchlist.html\)](http://support.novell.com/produpdate/patchlist.html)
- [Developer Support Forums \(http://developer.novell.com/ndk/devforums.htm\)](http://developer.novell.com/ndk/devforums.htm)
- e-mail: [devsup@novell.com](mailto:devsup@novell.com)
- Phone: (800) REDWORD or (801) 861-5544
- FAX: (801) 861-5224





# Setting Up the Test Environment

# 2

This chapter outlines the hardware and software requirements for LAN testing. It also provides instructions on setting up the test environment. It covers the following topics:

- [Section 2.1, “Reference Architecture Requirements,” on page 17](#)
- [Section 2.2, “Software Requirements,” on page 17](#)
- [Section 2.3, “Virtual Environments,” on page 17](#)
- [Section 2.4, “Uninstalling the Test Tools,” on page 18](#)
- [Section 2.5, “Installing the Test Tools,” on page 18](#)
- [Section 2.6, “Setting Up,” on page 18](#)
- [Section 2.7, “IP Addresses,” on page 20](#)
- [Section 2.8, “Setting the Device Activation Mode,” on page 20](#)
- [Section 2.9, “Troubleshooting,” on page 21](#)

## 2.1 Reference Architecture Requirements

The test machines should have the following hardware:

- Must be previously certified on SUSE<sup>®</sup> Linux Enterprise Server (SLES) 9
- At least 2 CPUs (hyperthreading counts as 2)
- At least 2 GB RAM (6 GB recommended)
- At least 1 Ghz CPU speed

## 2.2 Software Requirements

The tests must be executed on a server running the latest version (with the latest support packs) of the Novell<sup>®</sup> SUSE Linux OS on which you are certifying.

The test software is supplied by Novell on the Novell Developer Kit (NDK), available on CD-ROM or on the [NDK \(http://developer.novell.com/ndk/index.htm\)](http://developer.novell.com/ndk/index.htm).

To run the LAN tests, you need to download both the LAN Test Tools and TestConsole for SUSE Linux.

## 2.3 Virtual Environments

If you want to certify that your network adapter or driver works properly in a virtual environment, see [Chapter 5, “Testing in a Virtual Environment,” on page 37](#).

## 2.4 Uninstalling the Test Tools

If you are installing over a previous version of the TCLink, TestConsole, and/or the LAN Test Tools, you should uninstall the old version before you begin.

- 1 Log in as root.
- 2 Open a terminal and enter the following:  

```
rpm -e tconsole <Enter> to uninstall TestConsole.
```

or  

```
rpm -e lankit <Enter> to uninstall the LAN Test Tools.
```

or  

```
rpm -e tclink <Enter> to uninstall TCLink.
```

## 2.5 Installing the Test Tools

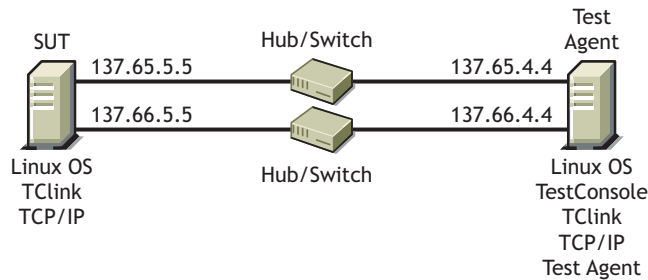
- 1 Ensure that you install and run these tests in an isolated test environment.
- 2 Ensure the GCC Compiler and the Linux Kernel Source have been installed (e.g., kernel-source-2.6.5-7.145.src.rpm). The test kit includes a kernel module that is compiled upon installation. This requires that the Linux source be present.
  - 2a Open a terminal.
  - 2b Type `rpm -qa | grep -i gcc <Enter>`.  
It should return the version of the GCC compiler. If nothing is returned, the compiler is not installed and you will need to use YaST to install it.
  - 2c Type `rpm -qa | grep -i kernel-source <Enter>`.  
It should return the version of the Linux kernel source. If nothing is returned, the kernel source is not installed and you will need to use YaST to install it.
  - 2d Log out and restart the system.
- 3 Log in as root.
- 4 Open a terminal and change to the directory where you downloaded the LAN Test Tools.
- 5 At a terminal, type `tar -xzf lantestkit.tar.gz <Enter>` to untar the file.
- 6 Use the command `rpm -ihv <rpm name>` to install the TCLink RPM and the LANKit rpm on two machines running the latest released version of SUSE® Linux with the latest support packs installed.
- 7 Use the command `rpm -ihv <rpm name>` to install the TestConsole RPM on one of the machines.

## 2.6 Setting Up

As shown in the figure below, testing a LAN adapter and driver pair requires at least two machines—one machine contains the adapter to be tested. This machine is referred to as the System Under Test (SUT). Software to test the adapter and driver runs on this system. The second machine will have running a special piece of software called the Test Agent. This machine is called the test

agent machine. The test agent machine must contain adapters compatible with the adapters being tested (e.g., speed, etc.).

**Figure 2-1** Test Configuration



The testing process is controlled by a program called TestConsole. This program is separate from both the test agent software and the software running on the SUT that performs the tests. TestConsole can be run on either the SUT or the Test Agent machine or can optionally be run on a separate third machine. We recommend that TestConsole be run on the same machine as the test agent.

One final software program should be mentioned here. TClink is a program that runs on both the SUT and the Test Agent machine. TestConsole also starts its own version of TClink.

- 1 Ensure that four Ethernet adapters are installed: Two adapters of the type being certified in the system under test and two in the Test Agent machine. We recommend that all four adapters be of the same type to ensure compatible characteristics (maximum linespeed, etc.). The two adapters under test must be identical.
- 2 Ensure the test machines are connected via Ethernet switches or hubs that support the maximum linespeed supported by the LAN adapter under test.

There should be two separate LAN segments and thus two separate switches or hubs connecting the test machines. One adapter from each machine should connect to one switch or hub and the second adapter from each machine should connect to a second switch or hub.

- 3 Log in as root on each of the machines in the test configuration.
- 4 Open a terminal and type `tclink <Enter>` on each of the machines in the test configuration.

This brings up a TClink agent that provides communications to TestConsole as well as providing other functionality to the LAN tests themselves. Leave this agent running for the duration of the testing. Status information is displayed on the TClink screen. This information may be useful if problems occur during testing.

- 5 Ensure that the IP address displayed in the TClink screen of the system being tested is the IP address of the adapter being tested. If the address is not the same as the adapter being tested, modify the `/etc/hosts` file to ensure that the adapter IP address you are testing has the host name.

---

**NOTE:** If TClink has been re-started on one of the machines, use the refresh button in the TestConsole test station selection dialog to add the IP address for that machine to the dialog.

---

## 2.7 IP Addresses

The LAN test suite requires that each LAN adapter in the system under test and the system running the test agent be assigned separate IP addresses. These addresses must all be in the same IP address class. In addition, the subnet mask for each of these addresses must be the “default” mask for the given address class as shown in the table below.

Class	Address Range	Default Mask
A	0.0.0.0 to 127.255.255.255	255.0.0.0
B	128.0.0.0 to 191.255.255.255	255.255.0.0
C	192.0.0.0 to 223.255.255.255	255.255.255.0

Each LAN adapter to be tested must be assigned a different subnet address. For example, 137.65.5.5. and 137.66.4.4 for a mask of 255.255.0.0.

Only use Internet addresses assigned to your company to avoid problems you may encounter with other companies or individuals through Internet connections, or verify your test systems are not connected to the Internet, such as through a company backbone/Intranet.

---

**IMPORTANT:** The IP address that you assign to the adapter under test is the IP address that you enter in TestConsole when asked for the IP address of the system under test.

---

The IP addresses of the SUT and the Test Agent can be entered manually in the TestConsole test station selection dialog, however, to be able to browse for the desired addresses, the IP address listed by TCLink, when it is first started, should match the address assigned to the adapter under test or the corresponding test agent adapter. If the address is not the same, modify the `/etc/hosts` file to ensure that the host name is associated with the desired IP address and that it is the first address listed in the file.

For example:

```
10.1.1.1 host.mycompany.com  hostname
10.2.1.1 host.mycompany.com
```

## 2.8 Setting the Device Activation Mode

The device activation mode for each network interface must be set to At Boot Time. You set the mode in YAST.

- 1 Start YAST and go to *Network Devices > Network Card > Network Card Configuration Overview*.
- 2 Highlight the network card that you want to configure and click *Edit*.
- 3 Click the *General* tab and select *At Boot Time* in the Device Activation drop-down list. Under SLES 9 the Device Activation drop-down list is found in the *Advanced* drop-down list, then *Detailed Settings*.

## 2.9 Troubleshooting

Run “Verify Linux Setup Test” in the TestConsole Linux Test Suite again if you unload then re-load TCLink.

Verify that only one copy of TCLink is running in each machine.

If you can't get TCLink to communicate, it might be a broadcast problem. Use YaST to delete the LAN adapter under test and then add it again. YAST should automatically detect the adapter and assign it the correct broadcast address. If it doesn't you can configure the file in the YAST utility.



# Using TestConsole

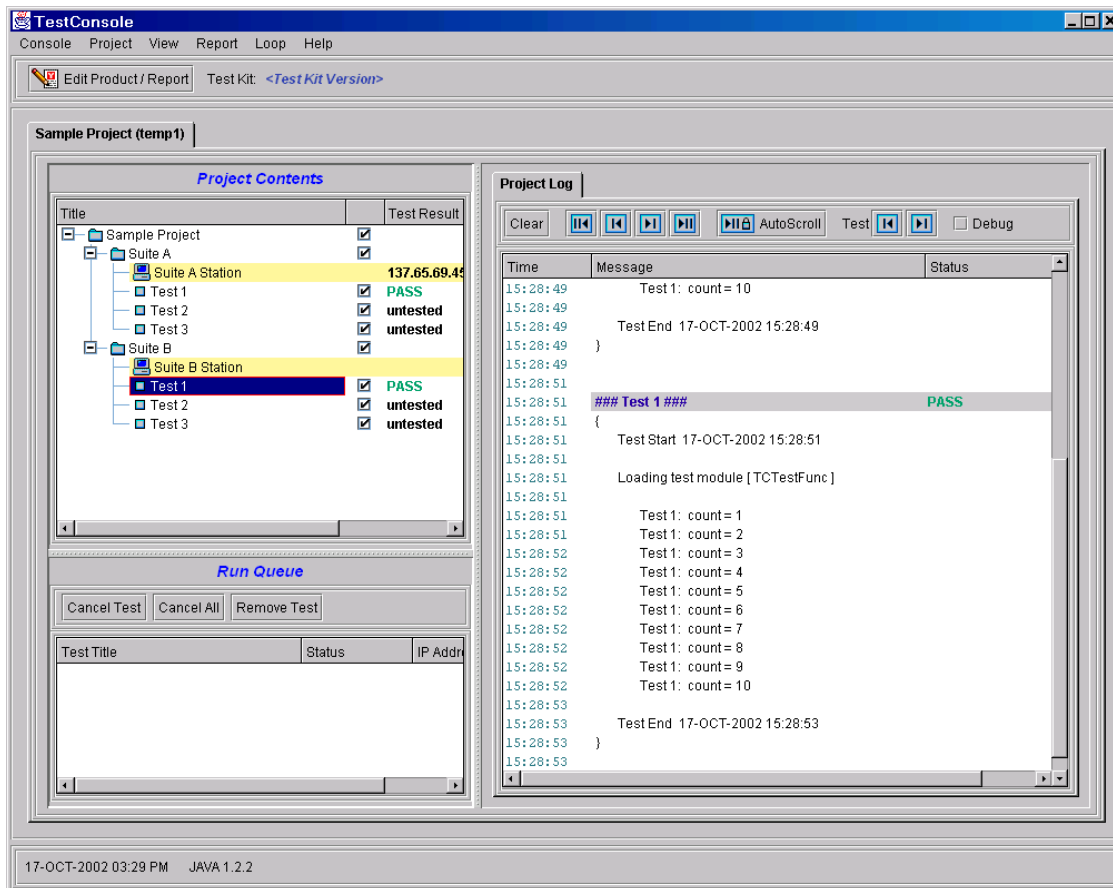
# 3

This section explains how to use TestConsole to run the Novell YES CERTIFIED tests. It covers the following topics:

- Section 3.2, “Managing Test Projects,” on page 26
- Section 3.3, “Testing in Debug Mode,” on page 26
- Section 3.4, “Using Loop Mode,” on page 27
- Section 3.5, “Reporting Test Results,” on page 28

## 3.1 TestConsole Overview

Figure 3-1 Shows the TestConsole Project Window - Tiled View



The TestConsole main window displays the following panes:

- Project Contents
- Run Queue
- Project Log

These panes can be displayed as tiled or tabbed. You can change the display in the View menu.

---

**NOTE:** Use the tabbed view if your monitor has a low resolution.

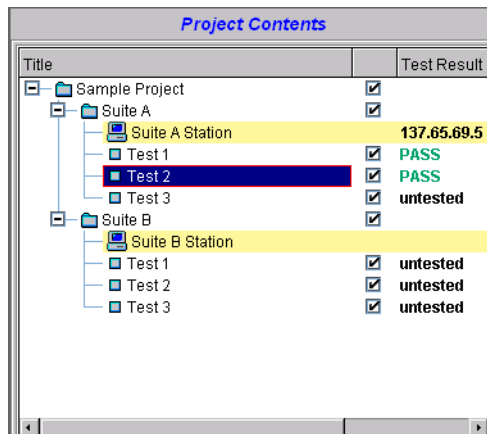
---

Each open test project is displayed on its own tab and can be quickly accessed by clicking on the tab with the test project name.

### 3.1.1 Project Contents

The Project Contents pane uses an expanding tree format to display the tests associated with the project you have chosen. For example, to expand or collapse a group of tests, click on the plus or minus icons next to the test.

*Figure 3-2 Shows the Project Contents panel of the TestConsole window.*



There are three columns in the Project Contents pane. The first column displays the title of the test. The second column indicates whether the test is selected to be run as part of a group. The third column indicates the final test result (untested, pass, fail, etc.).

The Project Contents pane may also display a Station Address icon (computer icon) to enable you to select your test station.

#### Running Tests

To run tests (or test groups) and view properties, right-click on any item in the Project Contents pane and choose an item from the popup menu. You may also double-click the test, test group, or suite that you want to run.

#### Selecting and Deselecting Tests

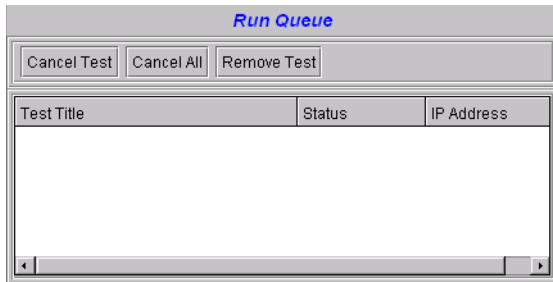
To select or deselect a test, click on the check marks in the second column. A test is selected if a check mark appears in the column. You can then run all tests in a group, suite, or project and it will skip the tests that have been deselected.



### 3.1.2 Run Queue

The Run Queue pane displays the tests that are currently running or queued to run. It displays the test name, its running status, and the IP address of the test station.

**Figure 3-3** Shows the Run Queue panel of the TestConsole window.



The buttons at the top of Run Queue pane enable you to cancel or remove tests in the queue.

---

Cancel	Cancels individual tests that are queued to run.
Cancel All	Cancels all tests that are queued to run.
Remove Test	Allows you to remove a test, even if it is hung.

---

**NOTE:** When a test is running, the word “running” appears in the status column. If the station under test loses its connection the status column will indicate the broken connection.

---

### 3.1.3 Project Log

The Project Log pane displays a log of the tests in your test project. It records the time each test started and finished, the result of the test, etc.

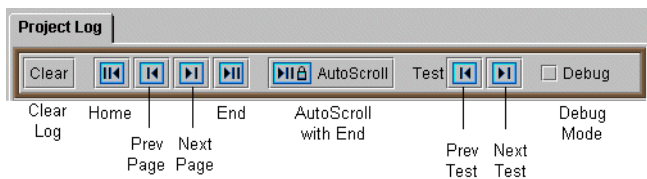
**NOTE:** To display additional information about the test, errors, warnings, and failures in the Project Log pane, run the test in debug mode (see [Section 3.3, “Testing in Debug Mode,”](#) on page 26). Not all tests provide debug information.

---

#### Project Log Navigation

The following graphic displays the control buttons and their functions in the log toolbar.

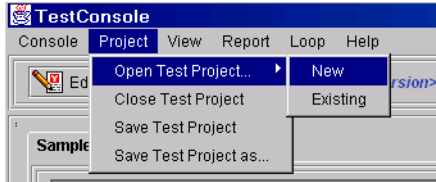
**Figure 3-4** Shows the Project Log toolbar.



## 3.2 Managing Test Projects

The tests that you run, along with their results and other information about your product, are called Test Projects. You can open, save, and close test projects.

*Figure 3-5 Shows the options on the Project menu.*



### 3.2.1 Opening a New Test Project

- 1 Open TestConsole by clicking the TestConsole icon on the desktop.
- 2 From the main menu, click Project > Open Test Project > New.  
The Select Project window appears. This window displays any installed PartnerNet compatibility test kits.
- 3 Select the type of test you want to load.
- 4 Click Select.

### 3.2.2 Opening an Existing Test Project

- 1 From the main menu, click Project > Open Test Project > Existing.  
The Select Project window appears. This window displays any previously saved test projects.
- 2 Select the project you want to run.
- 3 Click Select.

### 3.2.3 Saving a Test Project

- 1 To save a test project, from the main menu, click Project > Save Test Project.

Or

- 1 To save a test project with a different name, click Project > Save Test Project As.

### 3.2.4 Closing a Test Project

- 1 To close a test project, from the main menu, click Project > Close Test Project.

## 3.3 Testing in Debug Mode

Debug Mode displays additional information about errors and failures encountered while running the test. This information is displayed in the Event Log pane of the TestConsole window.

---

**NOTE:** Not all test modules support the logging of debug information.

---

To enable debug mode, click the Debug check box in the Event Log control bar.

---

**NOTE:** Debug mode runs slower than the normal test mode and can quickly fill up the event log. By default, the log is limited to 20,000 entries. When it exceeds 20,000, the entries at the beginning of the log are lost.

---

## 3.4 Using Loop Mode

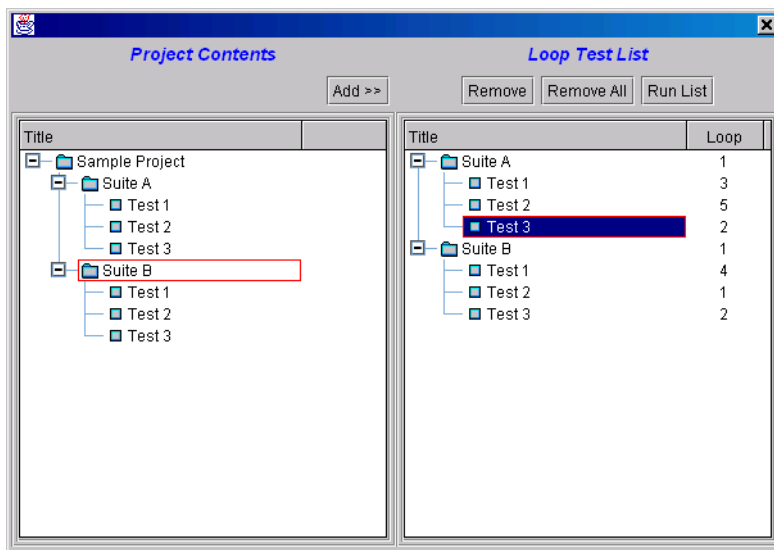
Loop mode allows you to set up a list of tests or test groups in any order and to run the tests multiple times in a loop. You can specify loop counts for individual tests as well as groups of tests.

### 3.4.1 Editing the Loop Test List

- 1 To create or edit the Loop Test List, from the menu bar select `Loop > Edit Loop List`. This displays a dialog box with two panes. The left pane displays the available Project Contents. The right pane displays the Loop Test List.
- 2 In the Project Contents pane, highlight the test or group that you want to add to the Loop Test List and click `Add`. The selected items are added to the loop list. You can continue to add items in any order.
- 3 To delete a test or group, highlight the item in the Loop Test List pane and click `Remove`.
- 4 To rearrange the order of the tests within a group, right-click the item in the Loop Test List pane and select `Move Up` or `Move Down` from the popup menu.
- 5 To edit the loop count for individual tests or test groups, right-click the test title and select `Loop Count`. This displays a dialog that allows you to edit the loop count properties.

The Loop Test List is saved until the project is closed. You can close the Loop dialog at any time.

**Figure 3-6** Shows the Loop dialog.



## 3.4.2 Running the Loop Test List

- 1 If the Loop List dialog window is open, click Run List.
- 2 If the Loop List dialog window is closed, from the menu bar click Loop > Run Loop List.

---

**NOTE:** Currently, loop mode restricts the number of test instances that can be placed on the Run Queue to 100 maximum. For example, if you have 2 tests in a group that run 25 times each with the group itself running 2 times, then you have reached your maximum.

---

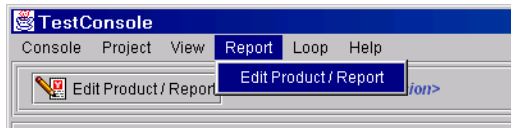
## 3.5 Reporting Test Results

All certification test kits use a common report window to record information about the product being tested. The Product & Report Information window enables you to do the following:

- Edit or view product and report information.
- Import or export information in the product library files.
- Print a sample test report (sample bulletin).
- Verify report data and test results.
- Create a Novell report for certification.
- Explain exceptions and testing errors.
- Clear all report data and test results.

To access the Product & Report Information window, click Report > Edit Product/Report from the main menu.

**Figure 3-7** Selecting Edit Product/Report from the Report menu.

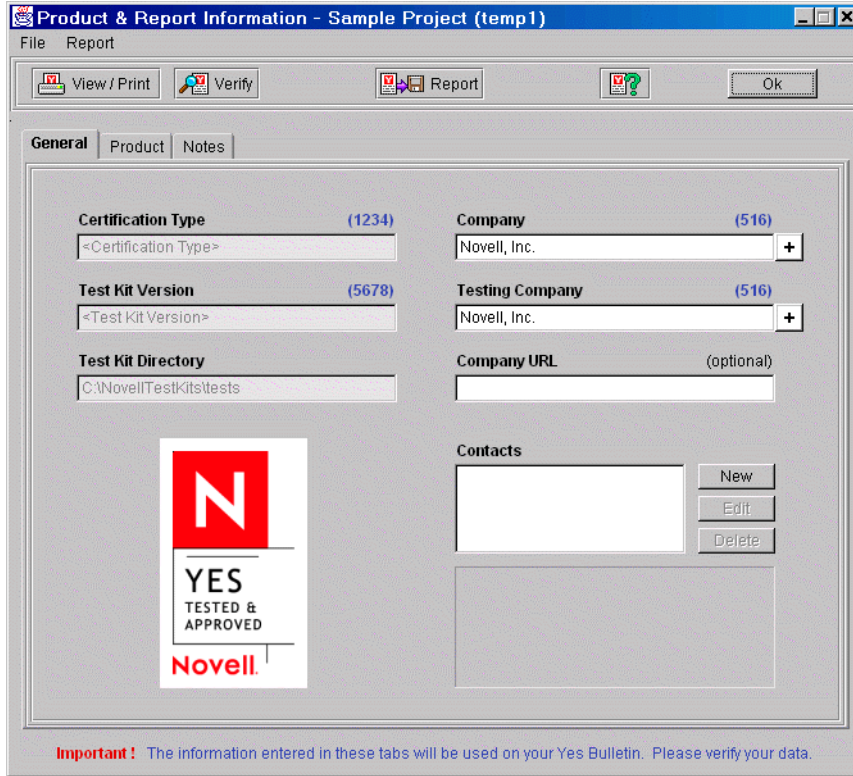


### 3.5.1 Entering Product Information

You enter information about your company and the product you are testing in the tabs of the Product & Report Information window. The tabs that appear in the window are different for each Novell certification test kit since each kit requires different kinds of information.

The data that you enter in the tabs of the Product & Report Information window is displayed on your product's bulletin. Therefore, it is important that the information be clear, concise, accurate, and suitable for publication.

**Figure 3-8** Shows the Report window.



### 3.5.2 Importing Product Information

You can retrieve product information. This feature streamlines data entry, enabling you to use information you have previously entered to test another product. You will, however, need to make adjustments to the data for your new product.

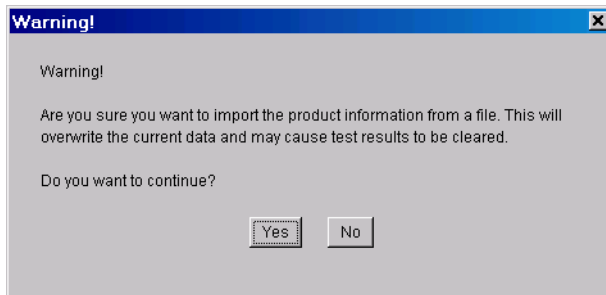
We recommend that you only use this feature when creating a new test project, before you have entered any data in the Product & Report Information window or run any tests, because importing product information automatically overwrites any existing data and may clear existing test results.

To import product information, do the following:

- 1 Click `File > Import Product File` from the Main menu.

When you import product information, TestConsole displays the following dialog to notify you that current data and test results may be cleared.

Figure 3-9 Shows a warning dialog.



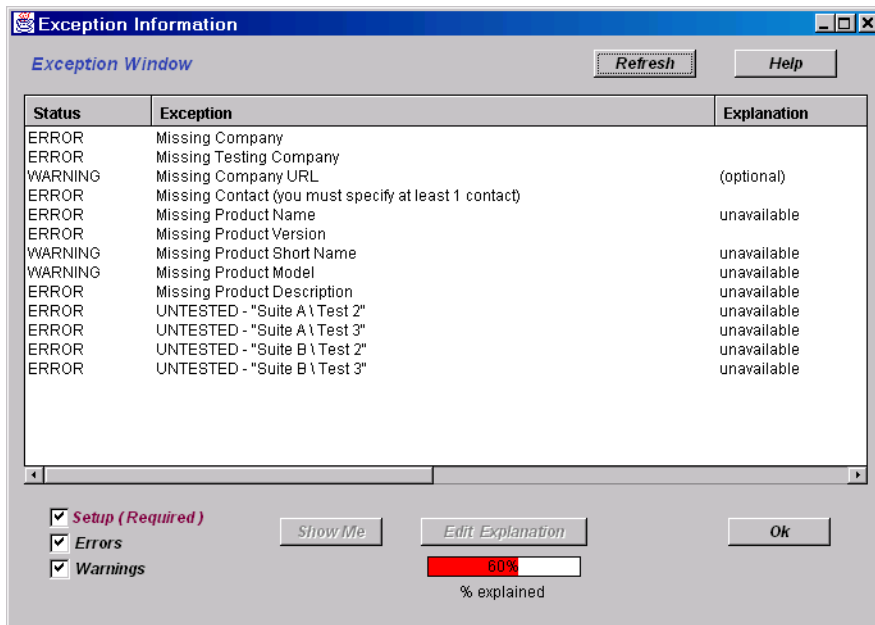
- 2 Click Yes if you want to continue.
- 3 Select the product file (listed with a .tsf extension) and click Open.  
The product file data populates the tabs in the Product & Report Information window.
- 4 Make the necessary modifications to the data for the new product you are testing.

### 3.5.3 Verifying Report Data

The Verify feature ensures that all report information has been entered and that all tests have a valid result. It enables you to validate your test project before generating a report.

To verify your test report, click Report > Verify Report Data from the Report menu. You can also click Verify on the Product & Report Information window toolbar.

Figure 3-10 Shows the Verify window.



Validation results are grouped into the following three categories:

---

Setup	Indicates issues that must be resolved before certification tests can be run.
Errors	Indicates a required item has a failure or is incomplete. Errors must be corrected before a test report can be generated.
Warnings	Indicates an optional item has a failure or is incomplete. Warnings should be corrected, but do not stop you from generating a test report and receiving certification.

---

The buttons on the window perform the following functions:

---

Refresh	Refreshes the list.
Help	Displays the Help dialog
Show Me	Takes you directly to the location of the selected error. You can then enter the correct information.
Edit Explanation	Launches the Explain Exception window.
OK	Closes the Exception Information window.

---

### Explaining Exceptions

There may be times when you are unable to correct an error, or a required portion of the test may not be applicable to the product you are testing. TestConsole requires that you include an explanation for these instances before you can generate a test report. A Novell Developer Support engineer evaluates each exception and its accompanying explanation and determines if a certification bulletin can be issued.

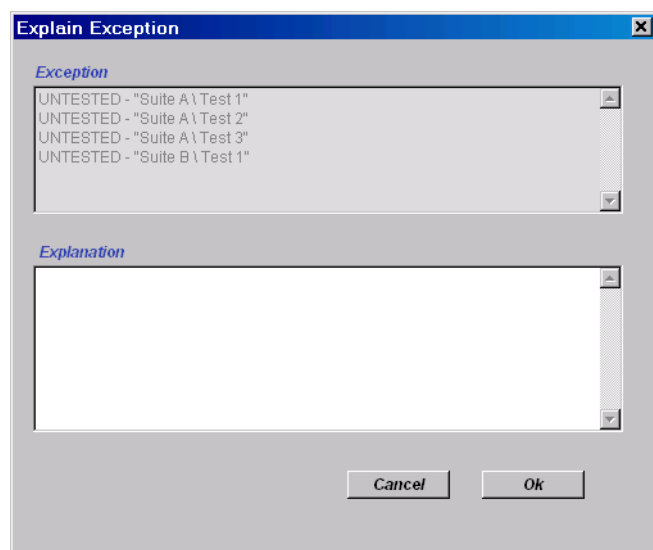
Select one or more errors to explain, then click the Edit Explanation button. This brings up the Explain Exception window for you to explain the selected errors.

---

**TIP:** If one explanation can be used for multiple exceptions, select the exceptions (hold down the <Shift> or <Control> keys while selecting), enter your explanation, then click the OK button.

---

**Figure 3-11** Shows the Exception Information window.



When all of your exceptions or failures have been corrected or explained, you can generate your test report.

### 3.5.4 Generating a Novell Report

After you have completed your testing and entered all required information in the Product & Report Information window, you can generate your test report.

A test report consists of your test results, information about the product and company, and all other files necessary for certification.

To generate a report, do the following:

- 1 Click Report > Generate Report from the main menu, or click Report on the toolbar.
- 2 Enter a filename for the results file.
- 3 Click Save.

TestConsole creates a .zip file of all necessary files and save them in the Results directory of the test kit you are using (for example, c:\novelltestkits\storage\results)

### 3.5.5 Viewing or Printing a Summary

You can view or print a summary of your YES CERTIFIED submission. We recommend that you print the summary and review it thoroughly to ensure that the information is correct and suitable for publication in the bulletin. If you need to make changes, you can do so, and then regenerate the report file.

---

**NOTE:** Requires a browser on the TestConsole station.

---

To print your summary report,



Click File > View/Print Summary from the Product & Report Information window, or click View/Print on the toolbar.

**Figure 3-12** Shows the Summary Report.



### 3.5.6 Clearing a Report

The Clear Report action is used to clear all report information and test results in the project. You should only perform this action if you want to clear all the user-entered information in the Report window.

To clear a report, do the following:

- 1 Click Report > Clear Report from the Product & Report Information window.

The following warning window is displayed to verify this is what you intended to do.

- 2 Click Continue.



# Testing Your Adapter

# 4

This section covers the following topics:

- [Section 4.1, “Running the Tests,” on page 35](#)

## 4.1 Running the Tests

---

**NOTE:** To run the tests in a virtual environment, see [Chapter 5, “Testing in a Virtual Environment,” on page 37](#).

---

- 1** If TCLink has been re-started on one of the machines, use the refresh button in the TestConsole test station selection dialog to add the IP address for that machine to the dialog.
- 2** From the Project menu, press *New*.
- 3** Select *Novell Linux LAN Tests (LinuxLAN.def)* for the project.
- 4** Double-click `Linux Test Suite` or right-click on it and select `Run Suite`. This will run all of the tests required for YES CERTIFIED status. Each of the tests can also be run individually by double-clicking on the test name. The tests are organized in a hierarchical structure.

Please note that some of the tests display performance numbers (average data rate: Mbytes/sec). These numbers are intended to be used for comparison between one run of a specific test and another run of the same test. The numbers are not meant to be used as absolute performance figures. When, in the tests, data is to be transferred, a beginning timestamp is saved. After the data has been sent or received, another timestamp is taken and the difference between it and the initial timestamp gives the time of the transfer. Then the performance is computed by taking the number of bytes transferred and dividing it by the time. The tests run in user space and there could be a significant amount of time overhead above and beyond the actual transfer of data.



# Testing in a Virtual Environment

# 5

If you want to certify that your network adapter or driver works properly in a virtual environment, you need to test with Xen Virtual Machine Host Server (XVMHS) installed and active. You must also create two virtual machines (VMs) running the latest version of either SUSE Linux Enterprise Desktop (SLED) 10 or SUSE Linux Enterprise Server (SLES) 10.

## 5.1 Installing XEN Virtual Machine Host Server

While installing the latest version of SLED 10 or SLES 10, be sure to select *Xen Virtual Machine Host Server* as one of the software patterns to install. If XVMHS was not installed during the initial installation of SLED or SLES, you can install it using Yast as follows:

- 1 Start YaST.
- 2 Select *Software* in the left pane of the YaST window.
- 3 Select *Software Management* in the in the right pane.
- 4 Expand the *Filter* drop-down list and select *Patterns*.
- 5 Go to the *Primary Functions* area and select *XEN Virtual Machine Host Server*.
- 6 Click *Accept* and follow the online instructions.

## 5.2 Creating the Virtual Machines

If your hardware supports full virtualization, you must create one fully virtualized machine and one paravirtualized machine. If your hardware does not support full virtualization, you must create two paravirtualized machines. This is done as follows:

- 1 Start YaST.
- 2 Select *System* in the left pane of the YaST window.
- 3 Select *Virtual Machine Management (XEN)* in the right pane.
- 4 Click *Add*.
- 5 Use the default method for installing the VM OS unless you have a disk image or a physical disk that contains the OS boot files.
- 6 Click *Next*.
- 7 Select the following settings in Virtual Machine Settings:
  - If the test hardware supports full virtualization, set the *Virtualization Mode* to *Full Virtualization* for one of the two virtual machines you need to create, otherwise select *Paravirtualization*.
  - Ensure that the *Operating System Installation Source* is set to the same location as for the original test system installation (install the latest version of SLED or SLES).
- 8 Under *Network*, select a unique IP address for each VM.
- 9 Click *Next* and follow the rest of the instructions until the VM is completely installed.
- 10 Repeat the steps above to create a second VM. Choose *Paravirtualization* for the second VM.

## 5.3 Running the Tests

- 1 Install and run TCLink and the LAN Test Tools for SUSE Linux on both VMs and the XVMHS (See [Section 2.5, “Installing the Test Tools,”](#) on page 18).
- 2 To run the tests, go to the TestConsole on XVMHS and select the *LinuxLAN-xen.def* project. For additional information see [Chapter 4, “Testing Your Adapter,”](#) on page 35.

# Submitting Your Test Results

# 6

This section provides instructions for submitting your test results to Novell after you have successfully completed all testing.

## 6.1 Additional Information

The TestConsole documentation contains information about other features of the Product & Report Information window. It explains how you can validate your data and print a sample bulletin.

To view the TestConsole documentation click `Help > Using TestConsole` from the main TestConsole window. Information about the Product & Report Information window is found under Reporting Test Results.

## 6.2 Enter Product & Company Information

- 1 Click `Report > Edit Product/Report` from the Test Console's Main menu bar, or click `Edit Product/Report` from the Quick Access Bar.
- 2 Click the General Tab.
- 3 Click the scroll arrow in the Company field and select your company from the list. If your company name does not appear in the list, select `New` and enter your company's name. Click `Select` to go back to the General tab.
- 4 Click on the scroll arrow in the Testing Company field and select the company performing the certification tests from the list. Click `Select` to return to the General tab.
- 5 Click in the Product URL field and enter the URL for the product's information page on your company web site.
- 6 In the Contacts area, click `New` and enter marketing and engineering contact information in the appropriate fields.
- 7 Click the Product tab.
- 8 In the Adapter Name and Adapter Version fields, enter the name and version number of the product you are testing.
- 9 Provide the adapter short name, product model (optional), and short description of your product in the appropriate fields.
- 10 Click the Attributes tab.
- 11 Click in Attribute Types box and highlight the first item in the list (Adapter Bus Width).
- 12 Click in the Attribute Values box and select the characteristic that corresponds to the item that is highlighted in the Product Attributes box.

If nothing appears in the Attributes Characteristics box, you can provide additional information in the field provided.

---

**NOTE:** You may not be able to select some fields. These are items that TestConsole will detect automatically and are listed as "Not User Settable."

---

- 13 Repeat the previous steps for each subsequent item in the Product Attributes box.

## 6.3 Driver and Module Information

The driver and module information is automatically detected and entered by TestConsole when the first test is executed.

---

**IMPORTANT:** We suggest you view the driver and module fields after the first test completes to verify the information for your driver is correct. Otherwise you could find out after testing is completed that the wrong driver was loaded, rendering your tests invalid.

---

- 1 While in the Product & Report Information window, select the Modules tab.
- 2 View the information and verify it is correct.

## 6.4 Submit Results

When you complete the test(s), submit the results to Novell. To submit files to Novell

- 1 Open the test project that you want to submit.
- 2 From the Test Console main menu, click `Report > Edit Product/Report`.
- 3 Click `Report > Generate Report`.

If there are any failures in the test(s) you have run, you will be prompted to explain the failures (See [Section 10.6, “Provide Explanations for Test Failures,”](#) on page 66).

- 4 Enter a filename for the results file.
- 5 Click `Save`.

TestConsole will create a .zip file of all necessary files and save them in the Results directory of the test kit you are using (for example, `/opt/novell/NovellTestKits/LAN/results`)

- 6 E-mail the .zip file to your Novell Developer Support Representative.

## 6.5 Provide Explanations for Test Failures

When you gather test results, TestConsole identifies any exceptions or failures that occurred during testing and prompts you to provide an explanation for it.

If there are unresolved exceptions in your test results, when you generate your test report a warning window will appear instructing you to click the `Verify` button. When you click `Verify`, the `Exception Information` window will appear.

Select each item in the `Exception` column, then click `Edit Explanation` and enter an explanation for the exception or failure. After you have entered an explanation, click `OK`.

---

**TIP:** If one explanation can be used for multiple exceptions, select the exceptions (hold down the `<Shift>` or `<Control>` keys while selecting), enter your explanation, then click `OK`.

---

When 100% of your exceptions or failures have been explained, click `OK` and continue generating your test report.

### Additional Information

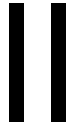
The TestConsole documentation contains information about other features of the Product & Report Information window. It explains how you can validate your data and print a sample bulletin.



To view the TestConsole documentation click `Help > Using TestConsole` from the main TestConsole window. Information about the Product & Report Information window is found under Reporting Test Results.



# ODI and CIOS LAN Driver Certification for NetWare



This part contains the test information and procedures for certifying your ODI/LAN driver for the NetWare platform. It contains the following sections:

- [Chapter 7, “Introduction to the LAN Tests,” on page 45](#)
- [Chapter 8, “Setting Up the Test Environment,” on page 47](#)
- [Chapter 9, “Running the Tests,” on page 55](#)
- [Chapter 10, “Submitting Your Test Results,” on page 65](#)



# Introduction to the LAN Tests

# 7

The LAN Test Tools provide a test environment to ensure LAN driver/adaptor compatibility with NetWare and other Novell products. Successful completion of these tests is required to obtain the YES CERTIFIED status and logo from Novell.

## 7.1 Multi-Mode and Multi-speed Test Requirements

The Line Speed, Connector Type, and Half/Full Duplex tests are part of the NetWare server suite for adapters/drivers capable of multiple speeds and modes (for example, 10/100 and full/half duplex). Therefore, additional passes through the full suite for alternate modes, speeds, and connection types are unnecessary for compatibility testing. We do recommend, however, that you run the complete suite for each mode and speed prior to compatibility testing as a quality measure.

---

**IMPORTANT:** You must complete the testing process (full NetWare server suite) using the worst case scenario for your adapter, such as 1 GB and/or full duplex.

---

## 7.2 Additional Testing Suggestions

Although these tests are designed to ensure that drivers conform to the ODI or CIOS specification and meet minimum quality requirements set by Novell, we highly recommend that additional testing be done in order to improve driver quality and reliability. This should be done at your site prior to submission. Some of the additional testing can be done using all or parts of the LAN Test Tools. The others will require tests of your own design.

The following are some suggestions for additional testing you can do.

- Exercise all hardware configuration combinations on your adapter.
- Design and perform hardware diagnostic tests for your adapter (refer to the IEEE specification for ideas).
- Execute additional performance testing.
- Perform additional stress testing, such as using larger network configurations with heavy, varied traffic, over an extended period of time.

---

**IMPORTANT:** Novell does not provide software or support for additional testing. Consult your driver developers for help with additional testing.

---

## 7.3 Support

The following support resources are available to assist you in your certification testing:

- [LAN group web page \(http://developer.novell.com/devres/lan\)](http://developer.novell.com/devres/lan)
- [Novell LAN Test Tools \(http://developer.novell.com/wiki/index.php/LAN\\_Test\\_Tools\)](http://developer.novell.com/wiki/index.php/LAN_Test_Tools)
- [Minimum Patch List page \(http://support.novell.com/produpdate/patchlist.html\)](http://support.novell.com/produpdate/patchlist.html)

- LAN Developer's Newsgroup (<http://developer.novell.com/ndk/devforums.htm>)
- e-mail: [devsup@novell.com](mailto:devsup@novell.com)
- Phone: (800) REDWORD or (801) 861-5544
- FAX: (801) 861-5224

# Setting Up the Test Environment

# 8

This chapter outlines the hardware and software requirements for LAN testing. It also provides instructions on setting up the test environment.

## 8.1 Hardware Requirements

The network hardware is supplied by the tester and must meet the minimum requirements as outlined in this chapter.

### Client

One client system is required to initiate and control the tests under TestConsole. It must be connected to the server through the server's primary adapter.

The minimum requirements for the client system are as follows:

- Pentium 133 processor
- 64 MB RAM
- Windows 98, ME, 2000, XP, or NT 4 with the current support pack
- Novell Client 32 for Windows
- One LAN adapter

---

**NOTE:** Ensure that there is only one LAN adapter to prevent problems with IP address detection by TestConsole.

---

### Servers

All server hardware, except the adapters under test, must be YES CERTIFIED. To determine if your server hardware is YES CERTIFIED, search for it on the [Novell YES CERTIFIED Product Certification Web site](http://www.novell.com/partners/yes/) (<http://www.novell.com/partners/yes/>).

Two servers (a primary and a secondary) are required for testing a 32-bit ODI and CIOS LAN drivers.

The minimum requirements for each server are as follows:

### Netware 4.x Servers

- Pentium 133 processor
- 64 MB RAM
- 500 MB disk space
- Two LAN adapters

---

**NOTE:** The adapters under test should be installed in the primary server and must be identical—unless you are testing an embedded LAN. The adapters in the secondary server can either be identical to those in the primary sever, or a pair of previously YES CERTIFIED adapters.

---

### NetWare 5.x Servers

- 3 Pentium 133 processors
- 256 MB RAM
- 1 GB disk space
- Two LAN adapters

---

**NOTE:** The adapters under test should be installed in the primary server and must be identical unless you are testing an embedded LAN. The adapters in the secondary server can either be identical to those in the primary sever, or a pair of previously YES CERTIFIED adapters.

---

### NetWare 6.x Servers

- 3 Pentium II processors
- 512 MB RAM
- 2+ GB disk space (200 Mb for DOS and 2 Gb for SYS)
- Two LAN adapters

---

**NOTE:** The adapters under test should be installed in the primary server and must be identical unless you are testing an embedded LAN. The adapters in the secondary server can either be identical to those in the primary sever, or a pair of previously YES CERTIFIED adapters.

---

It may be easier to install four identical adapters in the two test servers. However, if you have failures, you will need to replace the adapters in the secondary server with two YES CERTIFIED adapters to help in troubleshooting.

## 8.2 Software Requirements

The tests must be executed on a server running the latest version (with the latest support packs) of the NetWare OS on which you are certifying.

The test software is supplied by Novell on the Novell Developer Kit (NDK), available on CD-ROM or on the [NDK \(http://developer.novell.com/wiki/index.php/Category:Novell\\_Developer\\_Kit\)](http://developer.novell.com/wiki/index.php/Category:Novell_Developer_Kit) web site.

### 8.2.1 Stimulus Driver (Ethernet only)

To run the optional Promiscuous Mode Bad Packet and the Reject Error Packets tests, a “Stimulus” driver must be loaded for both adapters on the secondary server. This isn't necessary if your driver doesn't support bad packets in promiscuous mode, and you don't choose to run the Reject Error Packets test. A Stimulus driver is a driver that has been modified to be able to generate runt, jabber, long, and very short packets.

---

**NOTE:** A hub is required for the Promiscuous Mode Bad Packet and the Reject Error Packets tests. Using a switch will cause bad packets to be filtered out.

---

If you have an NE2000 adapter available, we suggest you use the SCNE2000.LAN (C language) or SNE2000.LAN Stimulus driver that are supplied with the tests, or you can also create a Stimulus



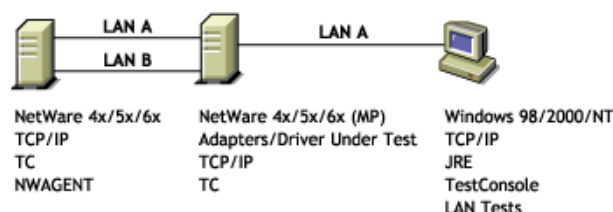
driver based on your driver code. See [Appendix C, “Stimulus Driver Generation,”](#) on page 95 for instructions on creating a Stimulus driver.

If you use the NE2000 adapter/driver you should only load it while running the Promiscuous Mode Bad Packet and the Reject Error Packets tests since the NE2000 only supports 10 Mb transfer rates. If you generate a Stimulus driver for your adapter you should have no problems leaving it loaded on the secondary server.

## 8.3 Setting Up

The following diagram shows hardware configuration that you should use to run the LAN Test Tools.

**Figure 8-1** Test Configuration



It is important to install your test server software in the following order.

- 1 Install NetWare on both test servers.
- 2 Install the latest patches on both servers. These can be found on the [Novell Support Web site \(http://support.novell.com/produpdate/patchlist.html\)](http://support.novell.com/produpdate/patchlist.html). This should place the current support modules (MSM/TSM) in the SYS:SYSTEM directory of each server.
- 3 Copy the driver and LDI file to be tested to the SYS:SYSTEM directory of each server.  
The LAN driver LDI file will be tested along with the driver itself. If the LDI file has problems, you will need to refer to the LDI specification and correct any problems. The LDI specification can be found on the [NDK \(http://developer.novell.com/wiki/index.php/ODI\\_LAN\\_Driver\\_Components\)](http://developer.novell.com/wiki/index.php/ODI_LAN_Driver_Components).
- 4 From the client, map a network drive to the SYS volume of both your primary and secondary test servers. You must also have Admin rights at the SYS volumes.

---

**NOTE:** Anywhere from 2 to 9 NetWare 4.x/5.x/6.x servers can be set up and mapped to for testing purposes, and the Install program will load the necessary files on all selected servers.

---

- 5 Download the [Novell TestConsole \(http://developer.novell.com/wiki/index.php/TestConsole\)](http://developer.novell.com/wiki/index.php/TestConsole) from the NDK and install it on the local drive of your test client.

TestConsole requires a client with an IP connection to 2 servers. The servers can be located remotely from the client on an intranet; however, you will need easy access to the servers to load test software, set up the test configuration, and run the manual tests. The tests launch from the client, but run on the primary server with a second providing traffic.

---

**IMPORTANT:** The [Java Runtime Environment \(http://java.sun.com\)](http://java.sun.com) (JRE) version 1.1.8 or 1.2.2 must also be installed.

---

- 6 Download the [Novell LAN Test Tools \(http://developer.novell.com/wiki/index.php/LAN\\_Test\\_Tools\)](http://developer.novell.com/wiki/index.php/LAN_Test_Tools) from the NDK and install it on your test client.

**7** Edit the startup.ncf and autoexec.ncf files.

**7a** Load NWCONFIG (If this is a NetWare 4 server, load INSTALL).

**7b** Select NCF Files Options.

**7c** Select Edit STARTUP.NCF file and add the following:

```
set maximum physical receive packet size = XXXXX (see note
below)
```

---

**NOTE:** The maximum physical receive packet size parameter reserves memory for incoming packets. Set the value to at least 16 bytes greater than the driver maximum packet size, or 24682, which is the largest possible value. For example, set the value equal to 1530 bytes for 1514 byte Ethernet packets.

---

**7d** Select Edit AUTOEXEC.NCF file.

---

**TIP:** See the “[Sample Autoexec.ncf](#)” on page 51 for an example of what yours should look like.

---

**7e** Verify the Load TCPIP statement is in the AUTOEXEC.NCF, and add it if not. The TCPIP protocol stack must load before the LAN driver.

**7f** Verify that the proper search paths are mapped in the AUTOEXEC.NCF and that the TC.NLM file is being loaded. It should contain the following:

```
search add sys:\testkits\lan
search add sys:\testkits\lib
search add sys:\testkits\tc
load tc
```

**7g** Edit the AUTOEXEC.NCF file to load the driver and all supported frame types for each adapter (see “[Loading Frame Types](#)” on page 50). The load and bind strings must be complete so that no driver parameters are requested at load time. See the instructions and example AUTOEXEC.NCF below.

**7h** Press <F10> to save the file.

**8** Restart the servers.

### 8.3.1 Loading Frame Types

You must load all supported frame types on both the primary and secondary adapters. The supported frame types for each topology are listed below. The IP protocol must be bound to one frame type for each adapter. If you are testing NetWare 4.x, IPX must also be bound to a second frame type, unless you set up an IP Gateway. Refer to “[IP Addresses](#)” on page 51.

---

**NOTE:** If using an IP Gateway, it is not necessary to bind an adapter to IPX. Otherwise bind one frame type to IP and one to IPX for each adapter, and be consistent on both primary and secondary servers. Other frame types need not be bound.

---

Ethernet:

- Ethernet\_802.2
- Ethernet\_802.3
- Ethernet\_SNAP

- Ethernet\_II (bind to IP)

Token-Ring:

- Token-Ring
- Token-Ring\_SNAP (bind to IP)

FDDI:

- FDDI\_802.2
- FDDI\_SNAP (bind to IP)

In addition to the frame types, the following statements will be added by the LANTEST.EXE install program.

```
set immediate purge of deleted files = on
set cpu hog timeout amount = 0
set upgrade low priority threads = on      (NetWare 4.x only)
```

Also for NetWare 4.x:

```
set minimum packet receive buffers = 800
set maximum packet receive buffers = 1000
```

For NetWare 5.x/6.x:

```
set minimum packet receive buffers = 1000
set maximum packet receive buffers = 1500
```

### 8.3.2 IP Addresses

The LAN test suite currently requires that all LAN adapters in the primary and secondary servers have one frame type bound to the IP protocol, as shown previously. It also requires this to be a Class-B IP address which is defined as the range from 128.1.0.01 to 191.254.0.0 (Mask = 255.255.0.0).

See the sample AUTOEXEC.NCF file that follows for examples.

You must verify that your IP subnet mask matches a Class B address, rather than the NetWare default Class C address.

Only use class-B addresses assigned to your company to avoid problems you may encounter with other companies or individuals through Internet connections, or verify your test systems are not connected to the Internet, such as through a company backbone/Intranet. Currently, TestConsole will only work with Class B addresses.

### 8.3.3 Sample Autoexec.ncf

A sample AUTOEXEC.NCF for an Ethernet NE2000 driver on the server is shown below. Both the primary and secondary servers should be set up the same way, except that the secondary may be running a different NetWare OS. This example is for a server without an IP Gateway setup, therefore IPX is bound to one frame type.

```
file server name TEST
ipx internal net 4321
```

```

;(

SECTION 1 - Statements in this section can be configured when
installing NetWare, or by editing AUTOEXEC.NCF after installation)

load tcpip

; Primary Adapter

load NE2000 port=340 int=4 frame=ETHERNET_802.2 name=E1_802.2
bind IPX to E1_802.2 net=1111

load NE2000 port=340 int=4 frame=ETHERNET_802.3 name=E1_802.3
load NE2000 port=340 int=4 frame=ETHERNET_SNAP name=E1_SNAP
load NE2000 port=340 int=4 frame=ETHERNET_II name=E1_II
bind IP to E1_II addr=137.77.1.1 mask=255.255.0.0

; Secondary Adapter

load NE2000 port=360 int=5 frame=ETHERNET_802.2 name=E2_802.2
bind IPX to E2_802.2 net=aaaa

load NE2000 port=360 int=5 frame=ETHERNET_802.3 name=E2_802.3
load NE2000 port=360 int=5 frame=ETHERNET_SNAP name=E2_SNAP
load NE2000 port=360 int=5 frame=ETHERNET_II name=E2_II
bind IP to E2_II addr=137.88.1.2 mask=255.255.0.0

;END OF SECTION 1

;SECTION 2 - Statements in this section should be added later by
TestConsole installation program - if not, add them before testing

set immediate purge of deleted files = on
set cpu hog timeout amount = 0
set upgrade low priority threads = on      (NetWare 4.x only)

search add sys:\testkits\lan
search add sys:\testkits\lib
search add sys:\testkits\tc
load tc

;END OF SECTION 2

;RECOMMENDED ADDITIONS - Edit AUTOEXEC.NCF

set restart server after abend = 0

;(This NetWare 5 statement will keep the server from restarting
automatically after an abend, which is better for troubleshooting)

#startx.ncf
;(this keeps Console Manager, ConsoleOne and RConsoleJ from loading -
saves approx. 32 MB of memory - Optional)

```

---

**IMPORTANT:** Do not load ROUTE.NLM on the server. The Source Routing Test uses its own Source Route Handler. Also the driver must be loaded from the server SYS:SYSTEM directory.

---



# Running the Tests

# 9

This chapter describes how to execute the LAN tests.

## 9.1 A Note on Testing PC Card Adapter and Drivers

You can test PC Card adapters/drivers on NetWare 6.x and 5.x using uniprocessor mode. Currently you will need to ignore warnings on the TestConsole screen indicating that you need to have 3 processors enabled. The Setup Test results will also indicate “Pass With Warning”, which is acceptable for PC Card testing. (See [Section 10.6, “Provide Explanations for Test Failures,”](#) on [page 66](#))

## 9.2 Running the LAN Tests

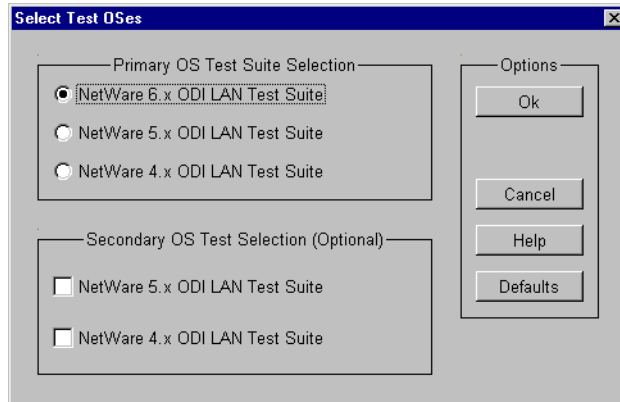
To run the LAN Tests, do the following:

- 1 Restart both servers to initiate the changes made in the STARTUP.NCF and AUTOEXEC.NCF files. In addition to loading the necessary programs on the test-server, the LANTEST installation also sets up the appropriate Search Drives which are necessary to run the tests, specifically:

```
search add sys:testkits\tc
search add sys:testkits\lan
```

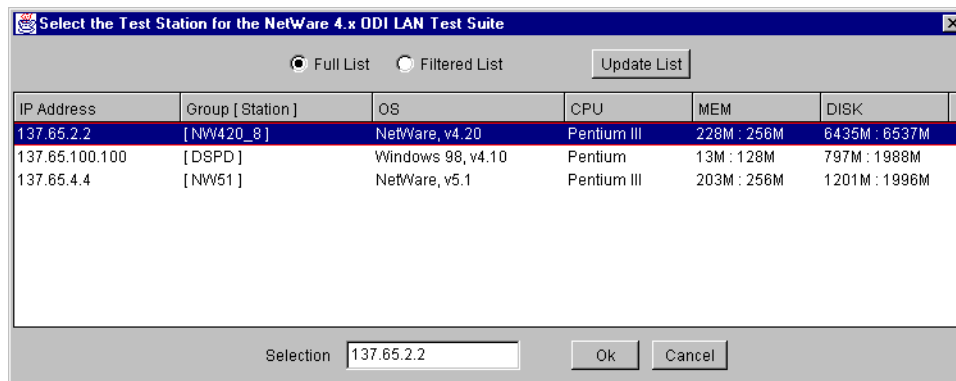
- 2 On the secondary server *you must* load the traffic generating program by typing  
LOAD NWAGENT
- 3 Run the TestConsole program by double-clicking the TestConsole icon placed on the Windows desktop during installation.
- 4 From the Project menu, select Open Test Project.
- 5 Select New if you are just beginning testing, or select Existing if you have already begun testing and saved a project.
- 6 Select LAN Test Tools.

- When the Select Test OS dialog appears, select the version or versions of NetWare you want to test, and click OK.



The OS that you select as your Primary OS will determine what Secondary OS selections are available. For example, if you select NetWare 6.x as your primary, you can select both NetWare 5.x and NetWare 4.x as secondary OS test suites. If you select NetWare 5.x as your primary, you can select NetWare 4.x as your secondary test suite.

- Select the test server by double-clicking on the Station Address icon for either the NetWare 6.x, NetWare 5.x, or NetWare 4.x LAN Test Suite. The Select the Test Station window will appear. The Full List shows all clients/servers on the subnetwork with TLink loaded, while the Filtered List shows only those servers running the correct version of NetWare for the suite selected.



Since the NetWare 4.x, 5.x, and 6.x test suites appear in the Project Contents window, you will be selecting the server for the suite you wish to run at this point. Selecting the Station Address icon for the NetWare 4.x suite in step 7 would allow you to select the test server for the NetWare 4.x suite, and the filtered list would show only NetWare 4.x servers. If you wish to see all client and server IP addresses on the network, select the Full List radio button at the top of the Select the Test Station window.

- Select the IP address of your test server and then click OK.
- Run the Verify LAN Setup Test to verify the test configuration.
- Select the Test Parameters button near the top left of the TestConsole window only if you wish to set values different than the defaults (see [Section 9.4, "Test Parameters,"](#) on page 59).



## 9.3 Test Execution

The NetWare test suite is designed to exercise the complete functionality of 32-bit LAN drivers on the NetWare 4.x/5.x/6.x operating system. The test suite is divided into the following categories:

- Basic Transmit and Receive Tests
- I/O (IOCTL) Control Tests
- Traffic and Endurance Tests
- Miscellaneous Tests
- Manual Tests
- Optional Tests

The tests can be executed in any order the operator prefers. Each test must be executed and generate a valid test result. All the tests are automated except for the Manual Tests listed below. The tests can be executed as the whole test suite at once, as a group, or individually. Individual tests allow you to select which sub-tests to run.

To begin testing select the test group and tests you wish to run. Some of the options for selecting and running tests are:

- Double-click on a suite or test group to run all tests in that suite/group.
- Expand a suite or test group and double-click on an individual test.
- Right-click on any of the above to choose other options, such as Quick Select, or viewing the Test/Group properties.

---

**NOTE:** Only one suite can be run at a time, either the NetWare 6.x, 5.x, or 4.x LAN Test Suite. Before switching to the other, we recommend that you save your project results (see [Section 9.10](#), “Changing Test Suites,” on page 62).

---

### 9.3.1 Automated Tests

Automated tests do not require operator assistance, and consist of the following test groups.

- “Basic Transmit and Receive Tests” on page 79 (20 to 40 minutes)
- “I/O Control (IOCTL) Tests” on page 82 (2 to 3 hours)
- “Traffic and Endurance Tests” on page 88 (2 to 5 hours)
- “Miscellaneous Tests” on page 90 (1 hour)

These groups also consist of individual tests and sub-tests. TestConsole allows you to select tests or groups over 4 levels, from the suite level (i.e., the NetWare 5.x LAN Test Suite) to the sub-test level.

---

**NOTE:** The times listed for Basic Transmit and Receive Tests are for drivers with maximum packet sizes of 4500 or less. If a driver supports promiscuous mode, test duration is extended by approximately 1.5 hours. Drivers that support 16K or larger packet sizes require 8 hours to complete these tests. If a driver supports 16K or larger packets and promiscuous mode, the Basic Transmit and Receive Tests require approximately 16 hours to complete. The I/O Control and Traffic/Endurance tests will also take longer with larger packet sizes.

---

To run the automated tests, do the following:

- 1 Select a suite, test group(s), or individual test(s) from the Project Contents window.

Right-click on a test group to view the various test options, such as running the entire test group or using the Quick Select option.

- 2 To run the tests you selected, double-click on the suite or test group to run all tests selected below that level.

For example, if you want to run just the first 2 tests under the Basic Transmit and Receive Tests group, you would

- 2a Expand the Basic Transmit and Receive Tests group by clicking on the (+).
- 2b Next select the Receive Test and the Look Ahead Test by clicking in the column next to the test name, making sure they are the only 2 with a check mark.
- 2c Double-click the Basic Transmit and Receive Tests group heading to execute the selected tests.

You may select individual tests or sub-tests by clicking on the (+) sign to expand and view the tests or sub-tests, then double-click on the test to execute the test. Sub-tests can only be executed from the “test” level or above. A test will always run its initialization and any selected sub-tests.

### 9.3.2 Manual Tests

Manual tests require operator intervention such as connecting and disconnecting cables or changing hardware options. The manual tests can be executed individually, or as a group in the order described below. These tests may take up to an hour to complete, depending on your setup time.

- “Connector Types Tests” on page 92 (5 minutes)
- “Linespeed Tests” on page 92 (5 minutes)
- “Half/Full Duplex Tests” on page 92 (5 minutes)
- “HIN Aware Tests” on page 93 (5 minutes)
- “Cable Disconnect Tests” on page 93 (5 minutes)

To begin testing:

- 1 Select *Manual Tests* from the Project Contents window.
- 2 Right-click on Manual Tests to view the test options, such as Running the test group (all manual tests).
- 3 Select individual tests by clicking on the (+) sign and opening the Manual Tests folder.

When tests are executed, the results are displayed in TestConsole's Event Log window. Pop-up windows on both the client and test server provide instructions during the tests, such as when to disconnect and reconnect cables and when to reconfigure your adapters. The Run Queue window displays the status of each test and indicates which test is currently running, along with the IP address of the test server.

The Event Log shows the results and other information about the tests as they run, including warnings. The buttons at the top of the Event Log window allow various options in viewing the log.

### 9.3.3 Optional Test

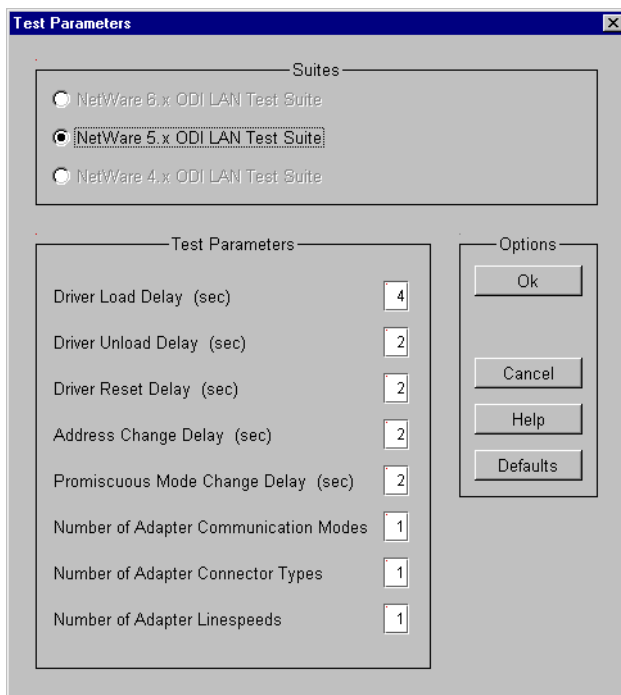
The Optional Test section consists of the “Performance Test” on page 93 (10 minutes).

Double-click or right-click on this test to execute it.

## 9.4 Test Parameters

The LAN test suites provide an option to adjust various parameters used in testing. Examples are driver load/unload/reset delays, adapter connection type and line speed. Most adapters can run with the defaults, but if you need to set them, do the following:

- 1 Run the Verify LAN Setup Test if it hasn't already been run; this initializes the default values.
- 2 Click on the Test Parameters button located near the upper left corner of the TestConsole main window; the Test Parameters window will appear.

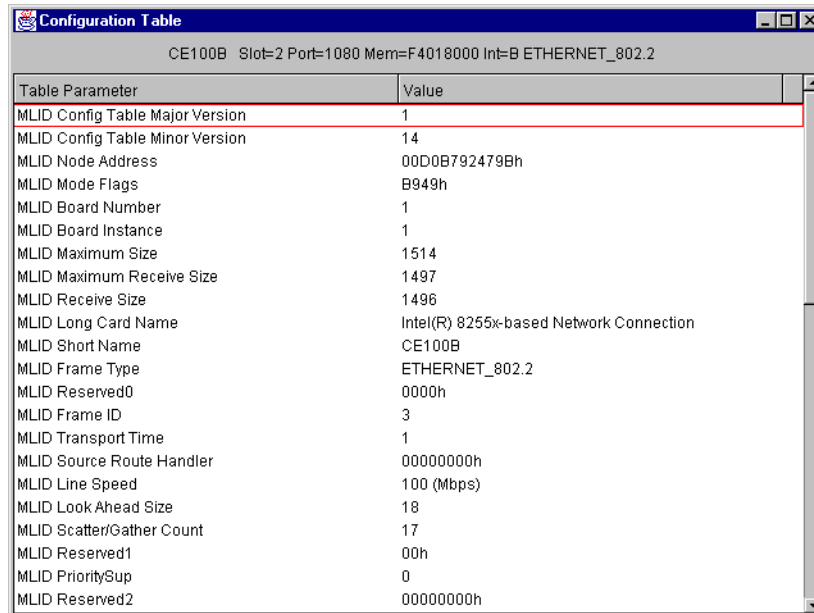


- 3 Select the test suite you are running.
- 4 Click the Help button to view information about the allowable ranges.
- 5 Type in any test parameters you wish to change in the windows to the right of each category.
- 6 Click the Ok button when done. A warning will appear if any parameters are set out of range, and it will tell you what the setting will be. The setting will be as close as allowed to the value you requested.
- 7 The Defaults button will reset all parameters back to the defaults.

## 9.5 Configuration Table

A button near the top of the TestConsole/Lan Test window activates the configuration table viewer. This will allow you to view the information for your adapter/driver.

*Figure 9-1 Shows the Configuration Table*



The screenshot shows a window titled "Configuration Table" with a subtitle "CE100B Slot=2 Port=1080 Mem=F4018000 Int=B ETHERNET\_802.2". The window contains a table with two columns: "Table Parameter" and "Value".

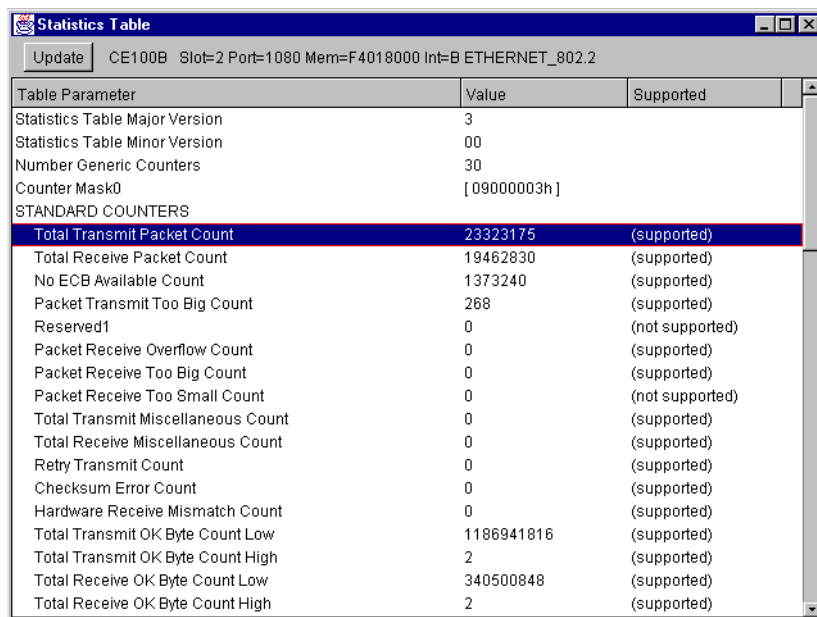
Table Parameter	Value
MLID Config Table Major Version	1
MLID Config Table Minor Version	14
MLID Node Address	00D0B792479Bh
MLID Mode Flags	B949h
MLID Board Number	1
MLID Board Instance	1
MLID Maximum Size	1514
MLID Maximum Receive Size	1497
MLID Receive Size	1496
MLID Long Card Name	Intel(R) 825x-based Network Connection
MLID Short Name	CE100B
MLID Frame Type	ETHERNET_802.2
MLID Reserved0	0000h
MLID Frame ID	3
MLID Transport Time	1
MLID Source Route Handler	00000000h
MLID Line Speed	100 (Mbps)
MLID Look Ahead Size	18
MLID Scatter/Gather Count	17
MLID Reserved1	00h
MLID PrioritySup	0
MLID Reserved2	00000000h

## 9.6 Statistics Table

Another button near the top of the TestConsole/Lan Test window activates the statistics table viewer. This will allow you to view the statistics for your adapter/driver.

To view the most recent statistics, click Update.

**Figure 9-2** Shows the Statistics Table Window



The screenshot shows a window titled "Statistics Table" with a toolbar containing an "Update" button and system information: "CE100B Slot=2 Port=1080 Mem=F4018000 Int=B ETHERNET\_802.2". The main area contains a table with three columns: "Table Parameter", "Value", and "Supported". The table lists various network statistics, with "Total Transmit Packet Count" highlighted in blue.

Table Parameter	Value	Supported
Statistics Table Major Version	3	
Statistics Table Minor Version	00	
Number Generic Counters	30	
Counter Mask0	[ 09000003h ]	
STANDARD COUNTERS		
Total Transmit Packet Count	23323175	(supported)
Total Receive Packet Count	19462830	(supported)
No ECB Available Count	1373240	(supported)
Packet Transmit Too Big Count	268	(supported)
Reserved1	0	(not supported)
Packet Receive Overflow Count	0	(supported)
Packet Receive Too Big Count	0	(supported)
Packet Receive Too Small Count	0	(not supported)
Total Transmit Miscellaneous Count	0	(supported)
Total Receive Miscellaneous Count	0	(supported)
Retry Transmit Count	0	(supported)
Checksum Error Count	0	(supported)
Hardware Receive Mismatch Count	0	(supported)
Total Transmit OK Byte Count Low	1186941816	(supported)
Total Transmit OK Byte Count High	2	(supported)
Total Receive OK Byte Count Low	340500848	(supported)
Total Receive OK Byte Count High	2	(supported)

## 9.7 Exiting and Saving Test Results

When testing is completed or you wish to exit, select an option from the Project menu to save or rename your project. Selecting Close Test Project will also give you an option to save your project.

---

**NOTE:** We recommend saving your test project (results) often to avoid possible loss of your test results.

---

**IMPORTANT:** If the LAN driver, HSM or TSM is altered in any way, the test suite will reset and the *results will be lost*, requiring all tests to be rerun.

---

When all the tests have completed successfully, you will be ready to enter the company and product information, and submit your results. See [“Submitting Your Test Results” on page 65](#) for instructions.

## 9.8 Test Results

TestConsole displays the official results of each test in the Project Contents window. Test results can be viewed in the Events Log as the tests are running.

Some tests are specific to certain topologies or test an optional feature not supported by all drivers. If a test does not apply to the driver under test, the result for that test indicates “Not Applicable.” If a test is applicable to the driver under test, but the feature being tested is optional and not supported by the driver, the result for that test indicates “Not Supported.”

A result of “Pass with Warning” requires that you either fix the problem or submit an explanation with your submission. An example of this would be if your driver uses an outdated API that is allowed during the transition to a new driver specification. Also PC Card adapters will receive an

invalid setup warning because they can't run on a system with the required 3 processors enabled. Except for this instance, you should correct the problem as soon as possible.

Each test should indicate one of the following results for submission:

- Pass
- Not Applicable
- Not Supported

These are the only acceptable results for certification testing. If a test indicates a different result, the problem must be corrected if possible, and the test must be run again. Otherwise an explanation must be included with your submission for consideration (see [Section 10.6, “Provide Explanations for Test Failures,” on page 66](#)).

Other test results are listed below:

- Pass w/Warning - Although the test passed, there was a configuration or other type of problem such as only having one processor enabled on a NetWare 5.x/6.x test setup. If the problem can't be fixed an explanation is required.
- Fail - The test failed due to a problem, likely in the driver or adapter.
- Not Testable - The test could not be run. This is often due to a setup problem such as no Stimulus driver for Promiscuous Mode testing.
- User Abort - Test was aborted by the operator during process.
- Error Abort - The test was aborted due to an error.
- In Progress - One or more segments of the test have not been completed, for example: one of the connector types or line speeds has not been tested.
- Untested - The test has not been executed.
- Testing - The test is currently running.

View the Events Log and turn on debug mode if necessary to help you with any test results you don't understand.

## 9.9 Debug Mode

To run the tests in debug mode, from the TestConsole screen select Console, then Debug Mode, or select the Debug check box on the Project Log tab.

Debug mode will show you additional testing information as it runs to help you debug driver problems. The tests will also run much slower, so only use it when necessary. When not in debug mode only the necessary amount of information is displayed on the log screen.

## 9.10 Changing Test Suites

If you configured your primary and secondary servers with the same adapters/drivers and test software, it is a simple process to change test suites.

- 1 Save the test results of the NetWare OS you have finished testing.
- 2 Verify that the NetWare server that you want to test is running the identical adapters/driver you just tested on the other server.

**3** Load the traffic agent on a server not under test by typing

```
LOAD NWAGENT
```

You may now begin testing by selecting the desired version of the NetWare LAN Test Suite from the Project Contents window in TestConsole.





# Submitting Your Test Results

# 10

This section provides instructions for submitting your test results to Novell after you have successfully completed all testing.

## 10.1 Additional Information

The TestConsole documentation contains information about other features of the Product & Report Information window. It explains how you can validate your data and print a sample bulletin.

To view the TestConsole documentation click `Help > Using TestConsole` from the main TestConsole window. Information about the Product & Report Information window is found under Reporting Test Results.

## 10.2 Enter Product & Company Information

- 1 Click `Report > Edit Product/Report` from the Test Console's Main menu bar, or click `Edit Product/Report` from the Quick Access Bar.
- 2 Click the General Tab.
- 3 Click the scroll arrow in the Company field and select your company from the list. If your company name does not appear in the list, select `New` and enter your company's name. Click `Select` to go back to the General tab.
- 4 Click on the scroll arrow in the Testing Company field and select the company performing the certification tests from the list. Click `Select` to return to the General tab.
- 5 Click in the Product URL field and enter the URL for the product's information page on your company web site.
- 6 In the Contacts area, click `New` and enter marketing and engineering contact information in the appropriate fields.
- 7 Click the Product tab.
- 8 In the Adapter Name and Adapter Version fields, enter the name and version number of the product you are testing.
- 9 Provide the adapter short name, product model (optional), and short description of your product in the appropriate fields.
- 10 Click the Attributes tab.
- 11 Click in Attribute Types box and highlight the first item in the list (Adapter Bus Width).
- 12 Click in the Attribute Values box and select the characteristic that corresponds to the item that is highlighted in the Product Attributes box.

If nothing appears in the Attributes Characteristics box, you can provide additional information in the field provided.

---

**NOTE:** You may not be able to select some fields. These are items that TestConsole will detect automatically and are listed as “Not User Settable.”

---

- 13 Repeat the previous steps for each subsequent item in the Product Attributes box.

## 10.3 Driver and Module Information

The driver and module information is automatically detected and entered by TestConsole when the first test is executed.

---

**IMPORTANT:** We suggest you view the driver and module fields after the first test completes to verify the information for your driver is correct. Otherwise you could find out after testing is completed that the wrong driver was loaded, rendering your tests invalid.

---

If there is a problem, verify that your driver is loading from the SYS:SYSTEM directory. The support modules also load from this directory.

- 1 While in the Product & Report Information window, select the Modules tab.
- 2 View the information and verify it is correct.

## 10.4 NetWare Server Information

- 1 While in the Product & Report Information window, select the NW Server tab.
- 2 Verify the information inserted by TestConsole is correct.
- 3 Enter the required information in the blank fields for the Test Station Make, Model and CPU speed. Also select the Test Station CPU Type from the pull-down menu.
- 4 Repeat steps 1-3 for each OS that was tested.

## 10.5 Submit Results

When you complete the test(s), submit the results to Novell. To submit files to Novell

- 1 Open the test project that you want to submit.
- 2 From the Test Console main menu, click `Report > Edit Product/Report`.
- 3 Click `Report > Generate Report`.

If there are any failures in the test(s) you have run, you will be prompted to explain the failures (See [Section 10.6, "Provide Explanations for Test Failures,"](#) on page 66).

- 4 Enter a filename for the results file.
- 5 Click `Save`.

TestConsole will create a .zip file of all necessary files and save them in the Results directory of the test kit you are using (for example, C:\NovellTestKits\Print\Results)

- 6 E-mail the .zip file to your Novell Developer Support Representative.

## 10.6 Provide Explanations for Test Failures

When you gather test results, TestConsole identifies any exceptions or failures that occurred during testing and prompts you to provide an explanation for it.

If there are unresolved exceptions in your test results, when you generate your test report a warning window will appear instructing you to click the Verify button. When you click Verify, the Exception Information window will appear.

Select each item in the Exception column, then click `Edit Explanation` and enter an explanation for the exception or failure. After you have entered an explanation, click `OK`.

---

**TIP:** If one explanation can be used for multiple exceptions, select the exceptions (hold down the `<Shift>` or `<Control>` keys while selecting), enter your explanation, then click `OK`.

---

When 100% of your exceptions or failures have been explained, click `OK` and continue generating your test report.

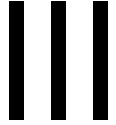
### **Additional Information**

The TestConsole documentation contains information about other features of the Product & Report Information window. It explains how you can validate your data and print a sample bulletin.

To view the TestConsole documentation click `Help > Using TestConsole` from the main TestConsole window. Information about the Product & Report Information window is found under `Reporting Test Results`.



# Appendixes



This section contains the following appendixes:

- [Appendix A, “Certification Requirements,” on page 71](#)
- [Appendix B, “Test Descriptions,” on page 79](#)
- [Appendix C, “Stimulus Driver Generation,” on page 95](#)



# Certification Requirements

# A

Certification requires successful completion of all of the tests in this test kit. All of the required tests are automated.

The certification tests in this kit cover basic functionality. Your own testing should be more thorough and comprehensive. The following criteria should be a guide for your testing.

This section covers the following topics:

- [Section A.1, “General,” on page 71](#)
- [Section A.2, “Documentation,” on page 71](#)
- [Section A.3, “Installation Tests,” on page 72](#)
- [Section A.4, “Operational Requirements,” on page 72](#)

## A.1 General

This section contains general requirements.

### A.1.1 Req\_LINUX\_LAN\_G001

The LAN adapter developer must supply the name and direct phone number of a technical support contact to assist Novell in the technical support of the adapter.

### A.1.2 Req\_LINUX\_LAN\_G002

The LAN adapter developer must resolve any major defects reported to Novell Technical Support within 90 days.

### A.1.3 Req\_LINUX\_LAN\_G003

The LAN adapter developer must resolve all major defects reported to Novell Technical Support before new versions of the adapter will be eligible for a bulletin.

### A.1.4 Req\_LINUX\_LAN\_G004

All configuration notes on previous bulletins must be resolved prior to granting a bulletin for the new versions of the product.

## A.2 Documentation

This section contains documentation requirements.

### **A.2.1 Req\_LINUX\_LAN\_D001**

The LAN adapter installation and setup/configuration procedures must be clearly outlined in the adapter's documentation.

### **A.2.2 Req\_LINUX\_LAN\_D002**

Common error messages generated by the driver are detailed in the product's documentation, and include why the error occurred and possible actions to resolve the error.

## **A.3 Installation Tests**

This section contains installation requirements.

### **A.3.1 Req\_LINUX\_LAN\_I001**

The LAN adapter will be tested on a Novell Approved Certification Testing Platform.

### **A.3.2 Req\_LINUX\_LAN\_I002**

The LAN driver does not have requirements or dependencies that require a recompiled replacement for any part of the SuSE Linux system, including the kernel, glibc, or any other packaged with the distribution.

Additionally, the LAN driver does not disrupt any other software previously installed on the approved testing platform, including SuSE provided components as well as products from other vendors.

### **A.3.3 Req\_LINUX\_LAN\_I003**

The LAN driver is installable, maintainable (updates) and removable without traces (except for configuration files) using the command line SuSE packaging backend rpm(8).

Software application packages have to be packaged using the build(8) script from build.rpm.

[See: [http://portal.suse.com/sbc/en/2004/02/yast\\_instsrc.html](http://portal.suse.com/sbc/en/2004/02/yast_instsrc.html)]

[Also see: yast2-packagemanager-devel.rpm]

### **A.3.4 Req\_LINUX\_LAN\_I004**

The LAN driver works properly when installed, then un-installed, then reinstalled.

## **A.4 Operational Requirements**

This section contains operational requirements.



#### **A.4.1 Req\_LINUX\_LAN\_O001**

Operate as documented without “hanging”, “crashing”, or producing protection faults (e.g. Writing beyond its own allocated memory).

#### **A.4.2 Req\_LINUX\_LAN\_O002**

Detect, and gracefully handle, any failed dynamic resource allocations such as memory, DMA channels, I/O ports, etc. This relates to Req\_LINUX\_LAN\_O001.

#### **A.4.3 Req\_LINUX\_LAN\_O003**

Transmit and Receive unicast packets of any size within the minimum to maximum packet size range for all supported frame types.

#### **A.4.4 Req\_LINUX\_LAN\_O004**

Transmit and Receive broadcast packets of any size within the minimum to maximum packet size range for all supported frame types.

#### **A.4.5 Req\_LINUX\_LAN\_O005**

Properly handle transmission of packets with multiple data fragments.

#### **A.4.6 Req\_LINUX\_LAN\_O006**

Properly handle packet fragments that have a zero byte length.

#### **A.4.7 Req\_LINUX\_LAN\_O007**

Properly handle packets with the maximum allowed number of fragments.

#### **A.4.8 Req\_LINUX\_LAN\_O009**

Properly handle packets with more than the maximum allowed number of fragments.

#### **A.4.9 Req\_LINUX\_LAN\_O010**

Properly handle multiple fragments that straddle two physical address blocks.

#### **A.4.10 Req\_LINUX\_LAN\_O011**

Properly handle Frame 802.2 Type I and Type II packets.

#### **A.4.11 Req\_LINUX\_LAN\_O012**

Properly maintain and report configuration information.

#### **A.4.12 Req\_LINUX\_LAN\_O013**

Properly maintain and report statistics such as error information.

#### **A.4.13 Req\_LINUX\_LAN\_O014**

Transmit and receive multicast packets of any size within the minimum to maximum packet size range for all supported frame types.

#### **A.4.14 Req\_LINUX\_LAN\_O015**

Properly handle the maximum number of multicast addresses supported.

#### **A.4.15 Req\_LINUX\_LAN\_O016**

Properly manage multicast addresses - addition and deletion of addresses, etc.

#### **A.4.16 Req\_LINUX\_LAN\_O017**

Operate properly in promiscuous mode if supported.

#### **A.4.17 Req\_LINUX\_LAN\_O018**

Properly handle heavy traffic load with various packet sizes for a period of at least 24 hours.

#### **A.4.18 Req\_LINUX\_LAN\_O019**

Properly handle transmit and receive overflow conditions.

#### **A.4.19 Req\_LINUX\_LAN\_O020**

Properly handle Quality of Service functionality (802.1p) if supported.

#### **A.4.20 Req\_LINUX\_LAN\_O021**

Operate properly on both uniprocessor and multiprocessor machines.

#### **A.4.21 Req\_LINUX\_LAN\_O022**

Properly handle jumbo packets if supported.

#### **A.4.22 Req\_LINUX\_LAN\_O023**

Properly handle dynamic loading and unloading of the driver.

#### **A.4.23 Req\_LINUX\_LAN\_O024**

Properly handle override of MAC address, if supported.

#### **A.4.24 Req\_LINUX\_LAN\_O025**

Properly handle packets with errors.

#### **A.4.25 Req\_LINUX\_LAN\_O026**

Properly handle hardware checksumming and other offload operations, if supported.

#### **A.4.26 Req\_LINUX\_LAN\_O027**

Properly handle cable disconnect and re-connect.

#### **A.4.27 Req\_LINUX\_LAN\_O028**

Properly operate at each line speed supported by the adapter. Also, properly auto-sense the line speed.

#### **A.4.28 Req\_LINUX\_LAN\_O029**

Properly operate in half duplex and full duplex modes as supported by the adapter.

#### **A.4.29 Req\_LINUX\_LAN\_O030**

Properly handle supported IOCTL operations. This includes those required to support Ethtool and Ifconfig.

#### **A.4.30 Req\_LINUX\_LAN\_O031**

Properly handle IOCTL requests that are not supported.

#### **A.4.31 Req\_LINUX\_LAN\_O032**

Properly handle Ipv6 (if supported) as well as Ipv4 addressing and operation.

#### **A.4.32 Req\_LINUX\_LAN\_O033**

Operate properly with at least two cards of each type to be certified.

#### **A.4.33 Req\_LINUX\_LAN\_O034**

All dynamically allocated resources are properly cleaned up and/or released. The driver must have no memory leaks.

#### **A.4.34 Req\_LINUX\_LAN\_O035**

Any exported driver functions properly handle incorrect parameters.

#### **A.4.35 Req\_LINUX\_LAN\_O036**

If the driver has load parameters, all allowable parameters are properly handled.

#### **A.4.36 Req\_LINUX\_LAN\_O037**

If the driver has load parameters, illegal or non-supported parameters are properly handled.

#### **A.4.37 Req\_LINUX\_LAN\_O038**

Driver code has no major problems as reported by Lint or similar tool.

#### **A.4.38 Req\_LINUX\_LAN\_O039**

Properly handles Wake on LAN functionality if supported.

#### **A.4.39 Req\_LINUX\_LAN\_O040**

Properly handles zero copy feature if supported.

#### **A.4.40 Req\_LINUX\_LAN\_O041**

Properly handles VLAN functionality if supported.

#### **A.4.41 Req\_LINUX\_LAN\_O042**

Exercise your driver's functions using a re-usable test plan. The test plan must cause 100% (or as close to 100% as possible) of the driver's internal functions to be exercised.

#### **A.4.42 Req\_LINUX\_LAN\_O043**

Properly handles multiple “simultaneous” accesses (from multiple processes, etc.).

#### **A.4.43 Req\_LINUX\_LAN\_O044**

Properly handles RDMA feature if supported.

#### **A.4.44 Req\_LINUX\_LAN\_O045**

The driver accesses shared resources (such as memory, etc.) by implementing appropriate synchronization (locking) mechanisms sufficient to prevent data corruption.

#### **A.4.45 Req\_LINUX\_LAN\_O046**

The driver does not modify native kernel machine code, or the machine code of any other loaded Loadable Kernel Module.

#### **A.4.46 Req\_LINUX\_LAN\_O047**

Driver IOCTL handlers return an appropriate error code for unknown IOCTL requests (for example, -EINVAL or -ENOTTY). Do not add driver IOCTLs that duplicate existing ones.

#### **A.4.47 Req\_LINUX\_LAN\_O048**

The driver does not implement privileged CPU instructions directly. Instead, the driver uses the standard Linux APIs for these instructions. API definitions of common privileged instructions are found in `asm/processor.h`.

#### **A.4.48 Req\_LINUX\_LAN\_O049**

The driver does not contain any dead code, or data structures (i.e., variables) in the binary executables. Also, the driver does not invent redundant functions for functionality that already exists in the kernel.

---

**NOTE:** If dead code or data structures exist in the source code, and is needed for future code/data expansion, it should be properly marked as such (using `#ifdefs` for example) to eliminate those code/data segments from the resulting binary

---

#### **A.4.49 Req\_LINUX\_LAN\_O050**

The driver reports unusual/unexpected errors using the standard `printk()` kernel function only when necessary. Also, driver `printk()` messages are informative, avoid flooding the console with error messages when error conditions are detected (flood control), and associate a `KERN_*` message (as defined in `/linux/include/linux/kernel.h`).

---

**NOTE:** `KERN_*` messages include `KERN_EMERG`, `KERN_ALERT`, `KERN_CRIT`, `KERN_ERR`, `KERN_WARNING`, `KERN_NOTICE`, `KERN_INFO`, and `KERN_DEBUG`. Code which could generate many messages should limit the rate at which calls to `printk()` are made (e.g. see `printk_ratelimit()`).

---

#### **A.4.50 Req\_LINUX\_LAN\_O051**

Driver does not call `panic()`. Device drivers should depend on the kernel core sub-systems to call `panic()` if needed, and must not call `panic()` directly.

#### **A.4.51 Req\_LINUX\_LAN\_O052**

Driver properly implements `sysfs` and the New Driver Model.

See also:

[linux-kernel-source/Documentation/driver-model/](#)  
[linux-kernel-source/Documentation/filesystems/sysfs.txt](#)  
[linux-kernel-source/Documentation/kobject.txt](#)

#### **A.4.52 Req\_LINUX\_LAN\_O053**

When initializing, the driver issues (kernel ring buffer) messages indicating which device(s) were found that the driver is prepared interface with.

#### **A.4.53 Req\_LINUX\_LAN\_O054**

The driver should not disable interrupts unless required. When required, the driver should limit disabling interrupts to as short a time as possible, and should not exceed 10ms.

#### **A.4.54 Req\_LINUX\_LAN\_O055**

The driver should use the standard Linux APIs provided in linux/semaphore.h and linux/spinlock.h. Do not implement proprietary (inline assembly) semaphores or spinlocks.

#### **A.4.55 Req\_LINUX\_LAN\_O056**

The driver uses the Linux PCI API to access PCI memory. The driver does not access PCI memory directly. See the following functions:

```
pci_read_config_byte  
pci_read_config_word  
pci_read_config_dword  
pci_write_config_byte  
pci_write_config_word  
pci_write_config_dword
```

# Test Descriptions

# B

This documentation describes the individual tests provided by the Novell LAN Test Tools. Refer to these descriptions when troubleshooting problems and more detail is necessary.

## B.1 Test Descriptions

This documentation describes the individual tests provided by the Novell LAN Test Tools. Refer to these descriptions when troubleshooting problems and more detail is necessary.

### B.1.1 Basic Transmit and Receive Tests

There are eight basic transmit and receive tests. It takes 20 to 40 minutes to complete these tests.

#### Receive Tests

The Receive Tests verify reception of DIRECT packets, min-to-max packet sizes, and all frame types. The tests verify the following:

- ECB status
- ECB BoardNumber
- ECB PacketLength
- ECB Data
- ECB Driver Workspace information
- ECB DriverWorkspace information
- ECB PreviousLink information
- ECB StackID
- ECB ProtocolID
- ECB ImmediateAddress

The receive tests also verify the reception of broadcast packets, min-to-max packet sizes, and all frame types. The tests verify the following:

- ECB DriverWorkspace information
- ECB PacketLength
- ECB Data

#### Look Ahead Tests

The Look Ahead Tests verify LookAhead fields for DIRECT packets, min-to-max packet size, all frame types.

Verify LkAhd\_StartCopyOffset (test selected packet size, selected offsets).

Verify different LkAhd\_DataBytesWanted values are handled correctly.

Verify LookAhead fields for BROADCAST packets (min-to-max packet size, all frames).

- Test IOCTLSetLookAheadSize with value = current value.

Verify IOCTL returns ODISTAT\_SUCCESSFUL and maintains value in configuration table.

- Test IOCTLSetLookAheadSize with value > current value.

Verify IOCTL returns ODISTAT\_SUCCESSFUL and updates value in configuration table.

- Test IOCTLSetLookAheadSize with value < current value.

Verify IOCTL returns ODISTAT\_SUCCESSFUL but does not update value in configuration table.

- Test IOCTLSetLookAheadSize with value = maximum look ahead value.

Verify IOCTL returns ODISTAT\_SUCCESSFUL and updates values in configuration table.

- Test IOCTLSetLookAheadSize with value > maximum look ahead value.

Verify IOCTL returns ODISTAT\_BAD\_PARAMETER and does not update value in configuration table.

Verify LookAhead requirements with promiscuous mode enabled, if supported.

### **Transmit Tests**

Verify transmit from min-to-max, all boards, medium data rate.

- Verify data length and validate data.

Verify transmit ECB fields have not been altered, min-to-max, all frames.

- Verify ESR has not been modified.
- Verify StackID has not been modified.
- Verify ProtocolID has not been modified.
- Verify BoardNumber has not been modified.
- Verify ImmediateAddress has not been modified.
- Verify ProtocolWorkspace has not been modified.
- Verify PacketLength has not been modified.
- Verify FragmentCount has not been modified.
- Verify Fragment has not been modified.

Verify errors for illegal length packets.

Verify the driver rejects its own broadcasts.

Verify data integrity using transmit ECBs with fragment counts of 1 to 16 (selected sizes, all frames).

Verify data integrity using transmit ECBs with 0 length fragments (selected sizes, all frames).



## Raw Send Tests

Verify transmit raw send packets from min-to-max, all boards, medium data rate.

- Verify data length and validate data.
- Verify data size evenized for evenized frame types.

Verify transmit ECB fields have not been altered.

- Verify ESR has not been modified.
- Verify StackID has not been modified.
- Verify ProtocolID has not been modified.
- Verify BoardNumber has not been modified.
- Verify ImmediateAddress has not been modified.
- Verify ProtocolWorkspace has not been modified.
- Verify DataLength has not been modified.
- Verify FragmentCount has not been modified.
- Verify FragmentDescriptor has not been modified.

## Source Routing Tests

Verify source route TX handler is called on transmits.

Verify source route RX handler is called on receives.

- Verify route information is correct.

Verify Address fields are correct and Rii bit is set.

Verify ECB DriverWorkspace has Source Route reception type bit set.

- Verify that ECB Data content is correct.

## Frame 802.2 Type I/II Tests

### Type I Requirements

Verify ECB DriverWorkSpace[1] = 01h for 802.2 Type I packets.

Verify ECB ProtocolID is correct.

Verify packet data length and data.

### Type II Requirements

Verify ECB DriverWorkSpace[1] = 02h for 802.2 Type II packets.

Verify ECB ProtocolID is correct.

Verify packet data length and data.

### **MAC Source Address Tests**

Verify the driver/adapter does not overwrite the MAC header source address in a raw send frame.  
We post only Pass or Not Supported for this test.

### **Virtual Memory Transmit Tests**

Verify the driver handles multiple fragments.

Verify the driver handles a single logical data buffer that straddles two physical address blocks.

## **B.1.2 I/O Control (IOCTL) Tests**

There are ten I/O Control tests. It takes 2 to 3 hours to complete these tests.

### **Configuration Table Tests**

Verify Configuration Signature.

Verify CFG Major and Minor Versions are 1.12 for Spec version 3.20.

Verify CFG Major and Minor Versions are 1.13 for Spec version 3.30.

Verify CFG Major and Minor Versions are 1.14 for Spec version 3.31.

Verify CFG Major and Minor Versions are 1.20 for Spec version 1.10.

Verify CFG Major and Minor Versions are 1.21 for Spec version 1.11.

Verify Node Address.

Verify Mode Flags.

Verify Board Number.

Verify Board Instance.

Verify Maximum Frame size.

Verify Best Data Size is correct for topology.

Verify Worst Data Size is correct for topology.

Verify Card (Long) Name.

Verify Short Name.

Verify Frame Type String and FrameID.

Verify Reserved 0 field is zero.

Verify Transport Time is not zero.

Verify Source Route Handler field.

Verify Line Speed.

Verify Look Ahead Size.

Configuration Table version 1.12 (Asm HSM 3.20)

- Verify Reserved 1 field is zero.

Configuration Table version 1.13 (Asm HSM 3.30)

- Verify Reserved 1 field is zero.
- Verify Scatter gather count is not zero if modeflag MM\_FRAGS\_PHYS\_BIT is set.

Configuration Table version 1.14, 1.20, and 1.21 (Asm HSM 3.31, C HSM 1.10 and 1.11)

- Verify Reserved 1 field is zero.
- Verify Scatter gather count is not zero if modeflag MM\_FRAGS\_PHYS\_BIT is set.
- Verify PrioritySup field is the same for all logical boards.
- Verify Reserved2 field is zero.

Verify Driver Major and Minor Version.

Verify Flags field.

Verify Send Retries.

Verify Driver Link.

Verify Sharing Flags.

Verify Slot field.

Verify IO Port 0.

Verify IO Range 0.

Verify IO Port 1.

Verify IO Range 1.

Verify Memory Address 0.

Verify Memory Size 0.

Verify Memory Address 1.

Verify Memory Size 1.

Verify Interrupt 0.

Verify Interrupt 1.

Verify DMA 0.

Verify DMA 1.

Verify Resource Tag.

Verify Configuration.

Verify Command String.

Verify Logical Name.

Verify Linear Memory 0.

Verify Linear Memory 1.

Verify Channel Number.

Configuration Table version 1.12 (Asm HSM 3.20)

- Verify IO Reserved is NULL.

Configuration Table version 1.13, 1.14, 1.20, and 1.21 (Asm HSM 3.30, 3.31, C HSM 1.10 and 1.11)

- Verify Dbus Tag field is the same on all logical boards.
- Verify DIOCfgMajor Version and DIOCfgMinor Version fields are valid.

### **Statistics Table Tests**

Verify IOCTLGetStatisticsTable (IOCTL 1).

Verify Statistics Table version is 3.0.

Verify number of Generic counters.

Verify Generic counter mask is valid.

Verify unsupported Generic counters are zero.

Verify Custom counters strings are valid and string block is terminated by two NULLs.

Verify number of Custom counters is valid and that there are no zero-length Custom strings.

### **Multicast/Functional Address Tests**

Verify that a single address can be added and deleted on each logical board and verify reception.

Verify return code: BAD\_PARAMETER or ITEM\_NOT\_PRESENT with an invalid address. Test both, add, and delete.

Verify that multiple identical addresses can be added and deleted. Verify reception.

Verify that multiple unique addresses can be added and deleted. Verify reception. After adding multiple unique addresses, verify driver does not receive on a unique address that was not added.

Verify that a unique address can be deleted from the middle of an address list. Verify reception.

Verify reset and shutdown requests do not affect multicast functionality. Verify reception.

### **Promiscuous Mode Good Packet Tests**

Verify basic promiscuous mode functionality. Verify traffic with good packet traffic tests.

Verify proper functionality when performing multiple consecutive promiscuous mode enables and disables.

Verify traffic with good packet traffic tests.

Verify multicast address changes made after promiscuous mode is enabled are resident after promiscuous mode is disabled. Verify traffic with good packet traffic tests.

Verify states and functionality are preserved after a partial shutdown and reset are performed while promiscuous mode is enabled. Verify traffic with good packet traffic tests.

Verify promiscuous remote multicast functionality. Post a warning if the adapter must go into full promiscuous mode in order to support this function. This is for assembly spec version 3.31 or greater and C spec version 1.11 or greater.

If MLID does not support promiscuous mode, verify MLID returns ODISTAT\_BAD\_COMMAND when trying to enable and disable promiscuous mode.

### **Promiscuous Mode Bad Packet Tests**

Verify traffic for valid length packets with no CRC.

Verify traffic for Runt packets with CRC.

Verify traffic for Runt packets with no CRC.

Verify traffic for Long packets with CRC.

Verify traffic for Long packets with no CRC.

Verify traffic for Jabber packets with CRC.

Verify traffic for Jabber packets with no CRC.

Verify traffic for Very Short packets with CRC.

Verify traffic for Very Short packets with no CRC.

Verify traffic for packets with MAC Header length field error (length is less than actual packets data size).

Verify traffic for packets with MAC Header length field error (length is greater than actual packet size).

Verify traffic for packets with MAC Header length field error (length is equal to zero in MAC header only).

### **MLID Management Tests**

Verify MLID returns ODISTAT\_BAD\_PARAMETER for illegal Management PIDs.

Verify MLID returns ODISTAT\_NO\_SUCH\_HANDLER for legal but unsupported Management PIDs.

### **Reserved and Undefined IOCTL Tests**

#### **ASM HSM Spec version 3.20 and 3.30**

Verify MLID returns ODISTAT\_BAD\_COMMAND for all reserved IOCTLs: 4, 7, 8, 12, and 13.

Verify MLID returns ODISTAT\_BAD\_COMMAND for all invalid IOCTLs: >=15.

### **ASM HSM Spec Version 3.31**

Verify MLID returns ODISTAT\_BAD\_COMMAND for all reserved IOCTLs: 4, 7, 8, 12, 13, and 15.

Verify MLID returns ODISTAT\_BAD\_COMMAND for all invalid IOCTLs: >=19.

#### **Transmit and Receive Monitor Tests**

Verify registration of Tx Monitor returns ODISTAT\_SUCCESSFUL.

- Verify monitor routine is called for each logical board and for each transmitted packet.
- Verify disabling Tx Monitor returns ODISTAT\_SUCCESSFUL.
- Verify monitor routine is not called after disabling Tx Monitor.

Verify registration of Tx Monitor twice returns ODISTAT\_OUT\_OF\_RESOURCES.

Verify registration of Tx Monitor with a NULL routine returns ODISTAT\_SUCCESSFUL and no Tx Monitoring occurs.

Verify registration of Tx Monitor still returns ODISTAT\_SUCCESSFUL when MLID is shut down.

Verify packets are not received when Monitor is enabled and MLID is shutdown.

- Verify packets are received when Monitor is enabled and MLID is reset.

Verify all TCB values are correct for each transmitted packet and frame type.

Verify disabling Tx Monitor with invalid routine pointer returns ODISTAT\_FAIL ODISTAT\_BAD\_PARAMETER and Tx Monitor is unaffected.

- Verify disabling Tx Monitor with invalid board number returns ODISTAT\_BAD\_PARAMETER and Tx Monitor is unaffected.

Enable promiscuous mode to verify monitor routine functions properly when in promiscuous mode.

- Verify registration of Tx Monitor returns ODISTAT\_SUCCESSFUL.
- Verify monitor routine is called for each logical board and for each packet transmitted.
- Verify disabling Tx Monitor returns ODISTAT\_SUCCESSFUL.
- Verify monitor routine is not called after disabling Tx Monitor.
- Disable promiscuous mode.

Verify monitor routine is called when performing a transmitted broadcast packet.

- Verify monitor routine is called when performing a transmitted multicast packet.

Verify registration of Rx Monitor returns ODISTAT\_SUCCESSFUL.

- Verify monitor routine is called for each logical board and for each transmitted packet.
- Verify disabling Tx Monitor returns ODISTAT\_SUCCESSFUL.
- Verify monitor routine is not called after disabling Tx Monitor.

Verify registration of Rx Monitor twice returns ODISTAT\_OUT\_OF\_RESOURCES.

Verify registration of Rx Monitor with a NULL routine returns ODISTAT\_SUCCESSFUL and no Rx Monitoring occurs.

Verify registration of Rx Monitor still returns ODISTAT\_SUCCESSFUL when MLID is shut down.

Verify packets are not received when Monitor is enabled and MLID is shut down.

- Verify packets are received when Monitor is enabled and MLID is reset.

Verify disabling Rx Monitor with invalid routine pointer returns ODISTAT\_FAIL ODISTAT\_BAD\_PARAMETER and Rx Monitor is unaffected.

- Verify disabling Rx Monitor with invalid board number returns ODISTAT\_BAD\_PARAMETER and Rx Monitor is unaffected.

Enable promiscuous mode to verify monitor routine functions properly when in promiscuous mode.

- Verify registration of Rx Monitor returns ODISTAT\_SUCCESSFUL.
- Verify monitor routine is called for each logical board and for each transmitted packet.
- Verify disabling Rx Monitor returns ODISTAT\_SUCCESSFUL.
- Verify monitor routine is not called after disabling Rx Monitor.
- Disable promiscuous mode.

Verify monitor routine is called when performing a transmitted broadcast packet.

Verify monitor routine is called when performing a transmitted multicast packet.

## Reset Tests

Verify basic driver reset functionality. Verify traffic after reset.

Verify consecutive driver reset calls. Verify traffic after final reset.

Verify MLID increments the AdapterResetCounter for the current board instance and not for any other instance.

Verify driver reset with an invalid board number returns ODISTAT\_BAD\_PARAMETER.

Verify basic driver reset instance functionality. Verify traffic after reset instance.

Verify when MLID is not assembly spec version 3.31 or greater or C spec version 1.11 or greater that it returns completion code: ODISTAT\_BAD\_COMMAND when the Reset Instance IOCTL function is called.

## Shutdown Tests

Verify basic partial shut down/reset functionality. Verify Configuration Table shutdown bit functionality by checking that shutdown bit is set after shutdown and cleared after reset.

Verify consecutive partial shutdowns. Verify traffic after a single reset.

Verify IOCTL (5) returns ODISTAT\_BAD\_PARAMETER status for an invalid board.

- Verify MLID will perform a Complete Shutdown and that deregistration has occurred by ensuring IOCTLGetConfiguration does not return ODISTAT\_SUCCESS.

- Verify the same number of boards are loaded at the end of the test as at the beginning.

Verify MLID will perform Complete Shutdown after Partial Shutdown with no reset between them.

Verify MLID will perform Complete Shutdown for hot plug functionality. This is checked by doing a Complete Shutdown and then reloading without having to completely unload after the shutdown.

- Verify applicable boards are de-registered.
- Verify MLID will reload, rebind, and then function properly.

Verify basic Partial Shutdown Instance/Reset Instance functionality of each logical board. Verify functionality of each board with traffic. Verify Configuration Table shutdown bit functionality by checking that bit is set after Partial Shutdown Instance and is cleared after Reset Instance. Check for bad command return code if spec does not support these IOCTLS.

Verify consecutive Partial Shutdown Instances and Reset Instances. Verify traffic after a single Reset Instance and then after multiple Reset Instances.

Verify Partial Shutdown Instance and Reset Instance of logical boards in sequential order. Boards will be shutdown in sequential order until all have been shutdown. Boards will then be reset in sequential order until all have been reset. Verify functionality with traffic after each board is partially shutdown or reset.

- Verify MLID will perform a Complete Shutdown Instance and that de-registration has occurred by ensuring that IOCTLGetConfiguration does not return ODISTAT\_SUCCESS. Verify same number of boards are loaded at the end of the test as at the beginning. Check for bad command return code if spec does not support IOCTL.

Verify MLID will perform Complete Shutdown Instance after Partial Shutdown Instance with no reset between them.

Verify MLID will perform Complete Shutdown Instance of sequential logical boards with no reload/rebind of boards until they have all been shutdown. Verify functionality of each board with traffic.

### **B.1.3 Traffic and Endurance Tests**

There are seven traffic and endurance tests. It takes 2 to 5 hours to complete these tests.

#### **Frame Integrity Tests**

Verify data integrity with incrementing (min-to-max) and decrementing (max-to-min) data sizes at medium load.

Verify data integrity with incrementing (min-to-max) and decrementing (max-to-min) data sizes at heavy load.

Verify data integrity with various sizes (100 packets each size) with the data pattern changing with each size at medium load.

Verify data integrity with various sizes (100 packets each size) with the data pattern changing with each size at heavy load.

Verify data integrity with random packet sizes and pattern change with each size for 5 minutes medium load.



Verify data integrity with random packet sizes and pattern change with each size for 5 minutes heavy load.

### **Internal Router Tests**

Verify data integrity with incrementing (min-to-max) and decrementing (max-to-min) data sizes at medium load.

Verify data integrity with incrementing (min-to-max) and decrementing (max-to-min) data sizes at heavy load.

Verify data integrity with various sizes (100 packets each size) with the data pattern changing with each size at medium load.

Verify data integrity with various sizes (100 packets each size) with the data pattern changing with each size at heavy load.

Verify data integrity with random packet sizes and pattern change with each size for 5 minutes medium load.

Verify data integrity with random packet sizes and pattern change with each size for 5 minutes heavy load.

The data patterns used in these tests are as follows:

- 8-Bit Incremental: Increment each byte (00, 01, 02, 03, ...)
- 8-Bit Decremental: Decrement each byte (FF, FE, FD, FC, ...)
- 8, 16, & 32 Bit Inverted: Invert each byte, word, or dword (AA, 55, ...)
- 8, 16, & 32 Bit Rotate Left: Rotate bit through each byte, word, or dword (01, 02, 04, 08, 10, 20, 40, ...)
- All bits set or cleared: All bytes FF or all bytes 00

### **Transmit Overflow Tests**

Verify MLID calls Transmit Complete in the correct sequence for all ECBs sent.

Verify MLID actually transmits ECBs in the correct sequence by monitoring Echo packets received back from the TGEN client.

Verify MLID calls Transmit Complete in the correct sequence for all RAW ECBs sent.

Verify MLID actually transmit RAW ECBs in the correct sequence by monitoring Echo packets received back from the TGEN client.

### **Receive Overflow Tests**

Verify the MLID properly handles multiple NO\_ECB\_AVAILABLE return status when calling any of the MSM AllocateRCB functions. Verify MLID has recovered from no ECB condition by sending normal traffic.

Verify MLID can handle a heavy receive load and recover from any receive overflow error conditions. Verify MLID has recovered from overflow condition by sending normal traffic.

### **Priority Packet Tests**

Verify MLID calls Transmit

Complete in the correct sequence for all ECBs sent of a particular priority level.

Repeat test case with background receive traffic.

Verify MLID calls Transmit Complete in the correct sequence for all RAW ECBs sent of a particular priority level.

Repeat test case with background receive traffic.

### **LAN Stress Test**

Multiprocessor stress test:

—Verify data integrity (15 minutes)

Uniprocessor stress test:

—Verify data integrity (10 minutes)

### **Jumbo Packet Test**

The Jumbo Packet Test duplicates the Frame Integrity Test, but does so by utilizing jumbo packets. The test is only applicable for drivers that support jumbo packets.

## **B.1.4 Miscellaneous Tests**

There are nine miscellaneous tests. It takes about 1 hour to complete these tests.

### **Information Integrity Tests**

Verify driver static information (description strings, spec version, driver version, and date).

### **Outdated API Tests**

Verify driver import list is valid for ASM HSM specification.

Verify driver import list is valid for C HSM specification.

### **Load/Unload Tests**

Verify DriverParameterBlock information. Allowances must be made for each of the different specification versions.

Verify repetitive load and unload calls. Verify same number of frames loaded after unload/reload sequence and validate with traffic.

Verify driver does not load twice with the same options.

Verify driver returns all resources when unloaded.

### **Node Address Override Tests**

Verify a valid node override address is accepted and the client responds on the new address.

- Verify the client does not respond to the original (default) node address.

Verify a multicast (or functional) node override address is rejected.

- Verify the address is modified to be a valid node or the driver maintains the default node address.

Verify a broadcast node override address is rejected.

- Verify the driver keeps the default node address or the address is modified to be a valid address.

### **Canonical/Noncanonical Tests**

Verify that the driver can operate correctly when configured to a canonical or noncanonical mode.

### **Reject Error Packet Tests**

Verify traffic for valid length packets with no CRC.

Verify traffic for Runt packets with CRC.

Verify traffic for Runt packets with no CRC.

Verify traffic for Long packets with CRC.

Verify traffic for Long packets with no CRC.

Verify traffic for Jabber packets with CRC.

Verify traffic for Jabber packets with no CRC.

Verify traffic for Very Short packets with CRC.

Verify traffic for Very Short packets with no CRC.

Verify traffic for packets with incorrect value in MAC header frame length field. Packet size is computed from MAC frame length is less than actual packet size.

Verify traffic for packets with incorrect value in MAC header frame length field. Packet size is computed from MAC frame length is greater than actual packet size.

Verify traffic for packets with incorrect values in MAC header frame length field. MAC frame length is zero.

### **HSM Checksum Tests**

Verify management IOCTL for HSM Checksumming for MCECB\_CON\_GET\_CAPABILITIES control function.

Verify management IOCTL for HSM Checksumming for MCECB\_CON\_ENABLE\_ACTIVE\_CAP control function.

Verify management IOCTL for HSM Checksumming for MCECB\_CON\_DISABLE\_ACTIVE\_CAP control function.

Verify management IOCTL for HSM Checksumming for MCECB\_CON\_DISABLE\_REMOVE\_CAP control function.

Verify HSM checksum functionality for IPv4.

- Verify HSM checksum functionality for TCP.
- Verify HSM checksum functionality for UDP.
- Verify HSM checksum functionality for RSVP.
- Verify HSM checksum functionality for ICMP.

### **MLID Time Slice Tests**

Verify the MLID's shutdown (driver unload) and initialization (driver load) routines do not run over 10 seconds without relinquishing control.

Verify the MLID's reset (IOCTL 6) routines do not run over 10 seconds without relinquishing control.

### **LDI/NBI File Tests**

Verify the LDI file syntax is correct and complete.

Verify the driver/adaptor can be auto-detected (if applicable) based on the PAR: and PROD: field values.

## **B.1.5 Manual Tests**

There are five manual tests. It takes about 30 minutes to complete these tests.

### **Connector Types Tests**

For each connector on the adapter, verify traffic data integrity with random packet sizes and random data patterns at a medium-high load. Traffic will be tested using the Frame Integrity test case 6. Record all connectors tested in the Setup Options.

### **Linespeed Tests**

For each linespeed supported by the adapter, verify traffic data integrity with random packet sizes and random data patterns at a medium-high load. Traffic will be tested using the Frame Integrity test case 6. Record all linespeeds tested in the Setup Options.

### **Half/Full Duplex Tests**

For each mode of transmission supported by the adapter, verify traffic data integrity with random packet sizes and random data patterns at a medium-high load. Traffic will be tested using the Frame Integrity test case 6. Record all transmission modes tested in the Setup Options.

### **HIN Aware Tests**

Verify that the bus tag and the slot values of the driver configuration table on all loaded boards are correct.

## **Cable Disconnect Tests**

Verify the driver can handle cable disconnects and reconnects at different locations. Verify the driver handles normal traffic before disconnecting the cable. Verify the driver recognizes a disconnected cable and/or times out on the transmit returning the ECB to the upper layers within 20 seconds (driver calls TransmitComplete/ServiceEvents). Verify the driver can handle normal traffic again when the cable is reconnected.

## **B.1.6 Optional Test**

There is one optional Performance Test. It takes about 10 minutes to complete this test.

### **Performance Test**

The Performance Test is designed as a development tool to assist LAN developers determine how driver modifications affect LAN throughput. The test measures the driver's ability to receive packets from 16 bytes to max packet size in 2x steps.

To use this test, simply run the test using the default/unmodified driver. Make the desired driver modifications and copy the revised driver to the SYS:SYSTEM directory replacing the existing driver. Rerun the test and compare the results by scrolling to the previous test results. To ensure the best possible results use servers with high speed CPUs and a quality hub/switch. This test was not designed to compare different LAN adapters on different systems due to hardware dependencies. If you wish to publish comparison results, use a test suite designed for that use that monitors the test environment as well as the performance.



# Stimulus Driver Generation

# C

This appendix describes the procedure for generating a stimulus driver to use in place of a normal 32-bit driver when testing promiscuous mode with bad packets. Since the *Bad Packet Tests* are only applicable to Ethernet, you only need to create a stimulus driver if the driver under test is Ethernet, the hardware supports bad packets, and you cannot use an NE2000 adapter. If you have an NE2000 available you may use the SCNE2000.LAN or SNE2000.LAN for testing. These drivers will be provided with the tests.

A stimulus driver is a driver that has been modified to be able to generate runt, jabber, long and very short packets; as well as to support the CRC and Length tests.

For your driver to work as a stimulus driver it must support raw sends and work with TCBs rather than ECBs. If your adapter is ECB-aware or doesn't support raw sends you will need to test using the NE2000 or else modify another driver for an adapter that has been YES CERTIFIED.

If for any reason the above options will not work for you, contact Developer Support for help.

## C.1 Generation Procedure for a 32-bit Driver

You will need the STIMULUS.INC file and SNE2000 source code to generate an assembly language 32-bit stimulus driver (.LAN). For a C-language driver you will need the STIMULUS.H file and the SCNE2000 source code. These files are installed with the LAN Test Tools in the C:\NovellTestKits\LAN\Stimulus.src directory.

To generate a 32-bit stimulus driver:

1. Copy the entire *Driver Management Routine* from the SCNE2000 or SNE2000, and place it at the beginning of the *Driver Management Routine* in your driver source code. If your driver doesn't have a *Driver Management Routine* you will need to copy it in as well as fill in the pointer *DriverManagementPtr*.
2. Copy the "IF STIMULUS" section of code from the beginning of the appropriate NE2000 *Driver Send Routine* into the beginning of your *Driver Send Routine*.
3. The *Driver Send Routine* uses the "TransmitConfiguration" variable. This variable must be defined in your driver and contain the address of the port that controls CRC operations on your adapter. You may change the variable name to conform to your driver.
4. Scan the appropriate NE2000 code for the word "STIMULUS" and make sure your driver has comparable code for each section you find.
5. You will need to modify the STIMULUS.H or STIMULUS.INC file to run tests that are supported in the driver under test and omit tests that are not. This is done at the end of either file where it says: "Modify These Values for Your NIC." For example, if your driver does not support the *Runt Test*, this line should indicate:

```
RUNT_SUPPORTED          equ Not_Supported
```

Insert "Supported" or "Not\_Supported" for each test case listed.

6. Finish generating your driver as you normally would, making sure the STIMULUS.H or STIMULUS.INC file is included.

---

**NOTE:** The SNE2000 has been changed to make room for two 2048-byte buffers and a special structure for handling bad packets has been added. Make sure these are included or modified in your driver. Also the SNE2000 has disabled back-to-back sends to make it easier to generate and maintain the stimulus driver. We recommend you do the same in your driver.

---



# Revision History

# D

The following table lists all changes made to the LAN Test Tools documentation.

Date	Changes
June 2006	Added <a href="#">Chapter 5, "Testing in a Virtual Environment,"</a> on page 37.
March 1, 2006	Updated format. Added <a href="#">Section 2.8, "Setting the Device Activation Mode,"</a> on page 20.
October 20, 2005	Removed a broken link. Updated Linux testing section. Updated NetWare 6.x requirements in the NetWare testing section.
October 5, 2005	Updated style and format.
June 15, 2005	Updated the following: <ul style="list-style-type: none"><li>• <a href="#">Chapter 1, "Introduction to the LAN Test Tools for SUSE LINUX,"</a> on page 15</li><li>• <a href="#">Section 1.1, "Additional Testing Suggestions,"</a> on page 15</li><li>• <a href="#">Section 2.7, "IP Addresses,"</a> on page 20</li><li>• <a href="#">Section 2.5, "Installing the Test Tools,"</a> on page 18</li><li>• <a href="#">Chapter 4, "Testing Your Adapter,"</a> on page 35</li><li>• <a href="#">Appendix A, "Certification Requirements,"</a> on page 71</li></ul>
March 2, 2005	Updated <a href="#">Section 2.7, "IP Addresses,"</a> on page 20 and <a href="#">Section 2.6, "Setting Up,"</a> on page 18.
November 18, 2004	Updated <a href="#">Chapter 1, "Introduction to the LAN Test Tools for SUSE LINUX,"</a> on page 15 and <a href="#">Section 2.1, "Reference Architecture Requirements,"</a> on page 17.
September 29, 2004	Changed references from DeveloperNet to PartnerNet and Yes, Tested and Approved to YES, CERTIFIED.
September 2, 2004	Updated the following sections: <ul style="list-style-type: none"><li>• <a href="#">Chapter 1, "Introduction to the LAN Test Tools for SUSE LINUX,"</a> on page 15</li><li>• <a href="#">Chapter 2, "Setting Up the Test Environment,"</a> on page 17</li><li>• <a href="#">Chapter 4, "Testing Your Adapter,"</a> on page 35</li><li>• <a href="#">Appendix A, "Certification Requirements,"</a> on page 71</li></ul>
August 9, 2004	Divided the document into parts for NetWare and Linux certification. Added Linux certification tests.
February 18, 2004	Minor Edits.
June 18, 2003	Updated links to DeveloperNet and Product Support.
March 2003	Updated documentation with minor editing revisions.
September 2002	Added references for CIOS.

---

Date	Changes
May 2002	Updated instructions for validating test results and explaining test failures.
February 2002	Updated procedures for submitting test results.
October 2001	Updated documentation for NetWare 6 testing.
Chapter 5, "Testing in a Virtual Environment," on page 37 June 2001	Titles added to all figures to ensure 508 compliance.

---