

**Getting Started
with
Core Internet-Connect ^(TM)
Version 2**

Core Ssystems
PO Box 31022
Walnut Creek, CA 94598
(510) 943-5765

Core Internet-Connect Release Notes

Table of Contents

License Agreement
1. Introduction
2. Internet-Connect Features
3. List of Files
4. Installation procedure
5. How to start a PPP connection
6. How to start a SLIP connection
7. Internet-Connect Scripts.
8. Internet-Connect in a LAN (Ethernet) configuration
8.1 Packet Driver
8.2 NDIS Driver (Lan Manager)
8.3 ODI Driver (Novell Netware)
9. How to setup a gateway configuration.
10. WSACONF
11. CORE.INI contents
12. WSACORE
13. WSADIAL
14. WSASPY
15. WSAPING
16. WSAFINGER
17. How to install your own IP packet filter.
18. Online help
19. Report Problems.
20. Trouble shooting tips.
21. Network configuration databases
22. Shareware and Public Domain software

Notes to users of Internet-Connect evaluation copy

If you are using an evaluation copy of Internet-Connect (downloaded from Internet), please note that the evaluation copy has a 30-day limit license number and will not run when the license expires. In addition, the evaluation copy allows applications to open up to 16 sockets only. If you are running many applications simultaneously, or if you are running applications which make heavy use of sockets, you may encounter out-of-sockets (10055) error message. Internet-Connect regular copy does not have this limitation.

IMPORTANT

PLEASE READ THE FOLLOWING LICENSE AGREEMENT CAREFULLY BEFORE INSTALLING AND/OR USING THE SOFTWARE. If you do not agree to this license, discontinue use of the software.

1. GRANT OF LICENSE: Core Systems grants to you the right to use one copy of the enclosed Core System software program ("The SOFTWARE") on a single terminal connected to a single computer (i.e. with a single CPU). You may not use the SOFTWARE on more than one computer or computer terminal at the same time.
2. COPYRIGHT: The SOFTWARE is owned by Core Systems and is protected by United States copyright laws and international treaty provisions. Therefore, you must treat the SOFTWARE like any other copyrighted material except that you may either (a) make one copy of the SOFTWARE solely for backup or archival purpose, or (b) transfer the SOFTWARE to a single hard disk provided you keep the original solely for backup or archival purposes. You may not copy the written materials accompanying the software.
3. OTHER RESTRICTIONS: This License permits you to use the SOFTWARE for your internal use provided that (a) the SOFTWARE is not modified in any way and (b) you maintain the copyright notice on all copies of the SOFTWARE. You may not rent, lease or otherwise transfer the SOFTWARE. You may not reverse engineer, decompile, or disassemble the SOFTWARE.

CORE SYSTEMS DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. NEITHER CORE SYSTEMS NOR ITS SUPPLIERS SHALL BE LIABLE FOR ANY DAMAGES WHATSOEVER, INCLUDING DIRECT, INDIRECT, CONSEQUENTIAL, INCIDENTAL, LOST PROFITS OR INFORMATION, BUSINESS INTERRUPTION, OR OTHER PECUNIARY LOSS, EVEN IF CORE SYSTEMS HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

The SOFTWARE is provided with RESTRICTED RIGHTS. Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 or subparagraphs (c)(1) and (2) of the Commercial Computer Software--Restricted Rights at 48 CFR 52.227-19, as applicable.

Manufacturer is Core Systems Inc., PO Box 31022, Walnut Creek, CA 94598. (510) 943-5765

This Agreement is governed by the laws of the State of California.

1. Introduction

Welcome to **Core Internet-Connect**, a powerful, yet easy-to-use networking package designed to enable Windows user to get access to the Internet.

Core Internet-Connect comprises of a fast and sophisticated TCP/IP stack, Winsock API, a PPP/SLIP driver with built-in dialer, automatic dialing, dynamic address assignment, scripting feature, network configuration utility and popular Internet applications.

Core Internet-Connect supports both LAN (Ethernet) connections and Serial (SLIP/CSLIP/PPP) connections. This means whether you are running Windows at work, where your machines are typically connected via a local area network (LAN), or at home where connection to the Internet is made via a modem, **Internet-Connect** applies well in either setting.

Core Internet-Connect comes with a Windows-based setup program which makes installation and setup a breeze.

Core Internet-Connect is designed with speed and ease-of-use in mind. The stack is implemented using 32-bit VxD and DLL. It is the fastest TCP/IP stack for Windows in the market today. It also requires zero DOS memory. For ease-of-use, the product supports **BOOTP** and **Dynamic Host Configuration Protocol (DHCP)** which means network configuration details can be administered from a central site rather than being entered by end users. For users requiring access to Internet via modem, **Core Internet-Connect** provides support for all protocols currently used for serial line accesses to the Internet, which means dial-up, SLIP, CSLIP and PPP. The product provides the most comprehensive PPP features in the market today. These include:

- Demand dial-up, which means connection to the net can be made simply by clicking an application, rather than going through multi-step process: dial-up service provider, login, starting application;
- Automatic modem hangup, idle timeout detection, redial.
- Dynamic address assignment.
- Callback (for secure connection).
- Server configuration.
- Scripting.
- Filtering.
- User-Id/Password Authentication Protocol (PAP).

Core Internet-Connect provides features to serve both as a development tool as well as a networking software platform. Whether you are a professional developer seeking a Windows^(TM) network development tool; or a user requiring a fast, robust networking package to run Window Socket applications, or a Value-Added-Reseller seeking a software utility to add networking capability to existing applications, this product is for you.

This release of **Core Internet-Connect** contains the following features:

^(TM) Product names are Registered Trademarks of their respective manufacturers

- **Winsock 1.1 API DLL**, library, header, and online reference help files.
- **TCP/IP protocol stack**. The stack is implemented completely as a Windows[™] DLL hence requires no below-640K conventional DOS memory. The stack is designed for robustness and speed. It provides a stable and high performance networking platform on which production applications can be deployed. Internal benchmark comparisons with other commercial TCP/IP stacks shows Core Systems TCP/IP consistently provide superior data transfer speed and throughput. Other features include support for **Domain Name Server (DNS)**, **128 sockets**, **raw socket**, **loopback**.
- **VXD driver** provides interface between protected mode DLLs and real mode LAN drivers. The driver's 32-bit mode operation enhances speed and responsiveness of the software.
- LAN (**Ethernet**) and **SLIP/CSLIP/PPP** connections (concurrently active).
- **BOOTP** and **DHCP** support.
- **WSASPY** utility captures and traces all Winsock API calls. This utility runs totally outside of Winsock and application under trace. It requires no modifications to either application or Winsock code, and can be used with Core's or other vendor's Winsock.
- Windows-based **PING** and **FINGER** utilities.
- **File Transfer (FTP)** application.
- Core **Internet-Connect** is compatible with the following network operating systems: **Novell Netware**, **Microsoft LAN Manager**, **Banyan VINES**, **Artisoft LANtastic** **DEC Pathworks**, and **IBM LANServer**. It supports the following LAN drivers: **Packet Driver**, **ODI Driver** and **NDIS Driver**.

This following shareware or public-domain software provide features which complement **Internet-Connect**. They are available on many Internet archive sites. If you have no Internet access, or have problem downloading these software, please contact us.

- **Pktdrvr**: Crynwr packet driver collection. This collection provides drivers for all commercial LAN adapters.
- **Qvtnet**: this shareware package provides Winsock-based **telnet**, **ftp**, **ftp server**, **rcp server**, **mail**, **news**, and **lpr** utilities.
- **Mosaic**: an excellent graphical interface tool available from NCSA, designed for user to browse, search and navigate for information on the Internet.
- **Netscape**: similar to Mosaic, from Netscape Corporation.
- **Eudora**: electronic mail package.
- **Networking shims**: (**odipkt.com**, **dis_pkt9.dos**, **ndispkt.sys**) conversion software which enable **Internet-Connect** to interface to different types of LAN drivers: **ODI**, **NDIS drivers**.

A wide variety of applications written to Winsock API currently exist. Many are available as shareware or public domain software. It is anticipated that over the next couple of years most distributed (client/server) applications running on PCs will be using Winsock as their main transport mechanism.

^(™) Product names are Registered Trademarks of their respective manufacturers

2. Summary of Features

Version 2.0 (LAN & SLIP/CSLIP/PPP support)

Version 1.1 (LAN support only)

Version 2.0 Feature Summary

- *Complete Winsock conformant: the product fully supports Winsock 1.1 Specification. All base and optional functions are supported:*
 - *Synchronous and Asynchronous function calls.*
 - *TCP, UDP and Raw socket types.*
 - *Up to 128 concurrently open sockets.*
 - *4KB UDP datagram size.*
- *TCP/IP stack , 100% implemented as Windows DLL, hence takes up zero below-640K DOS memory.*
- *32-bit VxD driver, provides super fast data transfer and interrupt responses.*
- *Full featured PPP support, including: dynamic address assignment, redial, demand dial-up, VJ compression, client and server mode, userid/password authentication (UPAP), scripting, filtering, callback.*
- *SLIP/CSLIP support.*
- *Multiple interfaces (Ethernet and SLIP/PPP active concurrently) with IP forwarding and routing.*
- *Custom-installed packet filter library.*
- *BOOTP and DHCP support.*
- *DHCP Client.*
- *Dialer with scripting and directory maintenance.*
- *Terminal emulation.*
- *Local loopback for testing purpose.*
- *Built-in DNS (Domain Name Server) support.*
- *Built-in networking databases (such as services, protocols files) to ease TCP/IP setup complexities.*
- *Windows-based configuration utility: enables first-time users to setup and configure a working system in minutes.*
- *Winsock monitor utility: provides:*
 - *Packet data trace.*
 - *SLIP/PPP trace.*
 - *TCP/UDP trace.*
 - *Buffer usage statistics.*
- *Comprehensive protocol, interface and session statistics.*
- *Winsock Spy utility: provides:*
 - *Winsock call trace: captures and logs Winsock calls and results.*
 - *Selective enabling and disabling tracing of particular Winsock function calls.*
 - *Hooks for future addition to trace other calls, such as Windows API.*
 - *Unobstrusive trace, requiring no modification to application code.*
 - *Works with all Winsock-compliant TCP/IP stacks.*

- *Windows Ping application.*
- *Windows Finger application.*
- *Windows FTP applications.*
- *Online Help, Winsock API and Release notes.*
- *Professional Windows-based installation program.*
- *Internet-Connect can run with the following network drivers:*
 - *NDIS.*
 - *ODI.*
 - *Packet Drivers.*
- *Internet-Connect is compatible with the following LAN networks:*
 - *Novell Netware.*
 - *Novell Personal Netware.*
 - *Microsoft Lan Manager, Windows for Workgroup (3.1 and above).*

Version 1.1 Feature Summary

Winsock Support

Winsock API v1.1
Winsock DLL, LIB, Header File
Winsock Online API Reference
128 sockets
Raw sockets

TCP/IP

100% DLL
TCP,UDP,IP protocols
ARP
Domain Name Server (DNS)

Network Interface Support

ODI, NDIS, Packet Driver

Network Operating System Compatibility

Novell Netware
Microsoft LAN Manager
Microsoft Windows for Workgroup
Banyan VINES
Artisoft LANtastic
DEC Pathworks
IBM LANServer

Debugging Tool

Winsock Call Trace

Applications

Ping, Finger

Hardware Requirements

IBM PC/XT/AT, PS/2, or compatible systems with 2MB or more memory
2 MB available disk space
Any Ethernet adapter with one of the following drivers:
NDIS (Microsoft/3Com)
ODI (Novell)
ODI/NDIS (Novell)
Packet Driver (Public Domain)

Software Requirements

PC-DOS or MS-DOS 3.30, 4.01, 5.0, 6.0, 6.1, 6.2
Microsoft Windows 3.x

3. List of Files

The following files are packaged in this product:

- readme.lst

Setup and initialization files:

- setup.exe
- core.ini

This documentation:

- relnotes.wri

Winsock API libraries and supported files:

- winsock.dll
- winsock.lib
- winsock.h
- winsock.hlp
- winsock.ico

Core TCP/IP protocol stack:

- coretcp.dll

Protected Mode/Real Mode interface driver:

- vxdwsa.386

Network configuration databases

- hosts
- services
- protocol
- resolver.cfg

Winsock background monitor program:

- wsacore.exe

Winsock API spy utility:

- wsaspy.exe
- wsatrap.dll
- dllpatch.dll

Winsock internet utility:

- wping.exe
- wfinger.exe
- cftp.exe

Problem report form

- problem.wri
- problem.txt

4. Installing Internet-Connect.

Internet-Connect includes a Windows-based installation program. To install, start Windows, then run the **Internet-Connect** setup program (setup.exe). After the installation is completed, you must add COREWSDS directory to your PATH in autoexec.bat and reboot the machine for the change to take effect.

After installing Internet-Connect, please be sure to enter your license number. Core Internet-Connect will not run without a valid license number, or will only run for a limited time with an evaluation license number. To enter license number, start Internet-Connect Network Configuration Utility, click More... button, then enter license number as shown on your Internet-Connect installation floppy.

5. How to start a PPP Connection

If Dynamic Address Assignment (i.e. IP address of your PC or the remote host is not known before connection is made) is used:

A. If IP address of remote host is known but address of local machine is not known:

- Start the **Network Configuration Utility (wsaconf.exe)**.
- On **Core Winsock Configuration/Local Host** dialog, leave all fields blank.
- On **Core Winsock Configuration/Name Server** dialog: check **DNS** button, fill in **Domain Name** field, and **Primary Name Server** IP address field. If you do not have these information, contact Technical Support of your Online Service Provider, or Administrator of your network.
- Click **Ethernet...** button, if you are not running both Ethernet and Serial connections at the same time, make sure the **Enabled** button on the **Ethernet Interface Configuration** dialog is unchecked.
- Click **SLIP/PPP...** button, from the **Serial Interface Configuration** dialog, fill in **Remote IP** field, leaving **Local IP** field blank.
- On **Protocol** group, select **PPP** button, and check **Primary Interface**, **Enabled**, **ICMP**, **VJ Header** boxes. These boxes have following meanings:

Primary Interface: in multiple-interface configuration (i.e. both Ethernet and Serial Interfaces are used), the interface that has this option checked is used by the stack as the default interface to reach the remote host if it fails to find a route to the remote host via other means.

Enabled: uncheck this option if the interface is not used, e.g. **Core Internet-Connect** supports up to four serial connections in parallel (COM1-COM4). For most configurations, only one of these interfaces should have this box checked, the other three interfaces should have this box unchecked.

ICMP: if this box is checked, ICMP packets (e.g. Ping packets) are allowed to be sent and received from this interface. Otherwise, they will be filtered out.

VJ Header Compression: check this box to enable Van Jacobsen compression of protocol header field.

Server: check this box if the local machine is to be used as PPP server, i.e. to receive incoming connection requests.

- Leave **Dialer** field blank, unless you wish to use your own dialer program, in which case fill in complete path name of your dialer here.
- Check that paths of **Port Monitor** (wsaslip.exe) and login **Script File** are correct.
- Click **Edit** to edit script file.
- Click **PPP Options...**, check **Dynamic Address Assignment** box, and verify other options. If unsure, leave other option boxes unchecked.
- For many PPP connections, **Password Authentication Protocol (PAP)** will be used. This means there is an exchange of user-id/password (at the PPP protocol level) required by the remote host at connection time for a successful PPP connection. Please note that this set of user-id and password is extra and different from the set of user-id and password which you normally type in to start a telnet or ftp session. If

your PPP service requires this information, enter **User ID** and **Password** fields of the **PAP Parameter** Group Box. Do not enter your telnet or Internet Account ID and Password here.

- Click **Port Config...**, verify that the settings are correct. Note **Online Status Check** group box. These are options to be used by Internet-Connect to find out if a connection is alive or not. Which option to use normally depend on type of modem: if you are using an internal modem, null modem or unsure, check **None**; if you are using a regular external modems, check **DCD**.

- Click **OK** on all dialogs.

- Start **Wping** program. Try to ping remote host. If the script is correct, modem should start dial and connection to remote host made.

- If you have problems making the connection, start **Wsaslip** and turn on the **Debug** option. This will allow you to view the execution and result of your script, as well as diagnostic messages generated during the connection.

Tips: If **wsaslip** fails to dial, click on its **Settings** menu item. Check to see if its **Port** settings are correct, i.e. match your actual configuration. Correct them to match your configuration, close **wsaslip** and restart your application.

B. If IP addresses of both remote host and local machine are not known:

- From **Network Configuration/Serial Interface** dialog, leave both **Remote IP** and **Local IP** fields blank.

- Follow same instructions as in option A above.

If Dynamic Address Assignment is NOT used (i.e. IP addresses of your machine and the remote host are preassigned and known)

- From **Network Configuration/Serial Interface** dialog, fill in **Remote IP** and **Local IP** fields.
- From **PPP Options...** dialog, uncheck **Dynamic Address Assignment** option.
- Follow same instructions as in option A above.

6. How to start a SLIP Connection

- Review instructions shown in option A (PPP Connection) above.
- Start the **Network Configuration Utility (wsaconf.exe)**.
- From **Network Configuration/Serial Interface** dialog, check **SLIP** protocol.
- Fill in **Local IP** and **Remote IP** addresses (This can be left blank if script is used to obtain IP addresses at the time connection is made).
- Check **VJ Header Compression** box if **CSLIP** is used.
- Click **Port Config...** and verify settings.
- Verify that paths shown in **Port Monitor** and **Script File** fields are correct.
- Click **Edit** and modify script file.
- Start **Wping** program to test connection.

7. Internet-Connect script syntax

```
#
#
#      Internet-Connect (TM) SLIP/PPP Startup Script Sample
#
# Each script line has the following general syntax
#      command arg1 arg2 ...
# Null line or line started with # is ignored (comment line)
# Number of arguments following each command depends on the command.
# Valid commands are:
#      done                -end script normally
#      fail                -terminate script abnormally
#      goto label_name     -jump to script line with label_name
#      label label_name    -label a script line
#      raisedtr            -raise DTR signal
#      dropdtr             -drop DTR signal (e.g. to hangup manually)
#      setcr [CarriageReturn_character] -change default CR character (\r)
#      pause pause_in_secs -short wait
#      nap pause_in_msecs  -shorter wait
#      launch application_to_be_launched [label_to_go_to_if_fail]
#      timeout default_timeout_in_secs
#      expect "expect_string" [timeout_in_secs [label_to_go_to_if_timeout]]
#          Note that expect_string must be in quotes if contain spaces
#      expectaddr [timeout_in_secs [label_to_go_to_if_timeout]] - expect IP address
#      send "send_string"   -send a string to host (CR appended automatically)
#      sendpass "password" -send a password (will not write to log)
#      send $variable      -prompt for variable string then send it to host
#      expect $variable [timeout_in_secs [label_to_go_if_timeout]]
# where $variable:
#      $p (password, send only)
#      $l (local IP address, expect only)
#      $r (remote IP address, expect only)
#      $c (any command, e.g. start SLIP, send only)
#      $u (userid, send only)
#
#
#Set default timeout
timeout 45
#Modem init string (modify to suit your modem)
send AT&F&D0
expect OK
#Wait a bit here as some modems fail to recognize the dial string
#if it was sent out right away
pause 1
#Number to dial
send ATDT9-123-1234567
#wait for CONNECT string, timeout in 60 secs
expect "CONNECT" 60 failexit
# Send some CR's to solicit a login prompt from host
send ""
send ""
expect ogin: 60 failexit
#Use the following command if you dont want to type in
#user-id everytime a connection is made
#send my_userid
(™) Product names are Registered Trademarks of their respective manufacturers
```



```
#Use following format to let script prompt you for user-id
#at beginning of each connection
send $u
expect assword: 30 failexit
#To let script send in the password
#sendpass my_password
#To let script prompt you for password
sendpass $p
#Wait for Greetings message
expect opyright 30 failexit
#send ""
#send ""
#Exit here if PPP connection
# Send a command to start up SLIP
#send "SLIP"
#Or let script prompt for command
#send $c
#Wait a bit here to give host time to setup SLIP interface
#pause 5
# get host (remote) IP address
#expect $r
# get your (local) IP address
#expect $l
#successful ending of script
done
#abnormal termination
label failexit
fail
```

8. Internet-Connect in a LAN (Ethernet) configuration

Core **Internet-Connect** uses TCP/IP as its transport protocol. It supports a wide variety of LAN drivers and is compatible with other network operating systems including Novell Netware and Microsoft LAN Manager. To setup Internet-Connect in a LAN configuration you need to be somewhat familiar with TCP/IP and its administration. This section provides information to help you configure the networking environment for Internet-Connect.

We are providing two files, lmstup.txt and nwsetup.txt, which contains sample entries in relevant configuration files (autoexec.bat, net.cfg and protocol.ini) as examples of how to setup LAN driver for Core Winsock to work in Windows for Workgroup (TM) and Netware (TM) environments.

The network setup procedure involves two following steps:

Step 1: Setup TCP/IP configuration.

- Determine the IP address and name of your machine or skip this step if **BOOTP/DHCP** is used.
- Start up the Network Configuration utility (wsaconf.exe). Fill in appropriate information.

Step 2: Setup Winsock to run with other networking operating systems. First determine what networking operating system (Lan Manger, Novell, TCP/IP etc.), and LAN driver (ODI, packet driver, NDIS, etc.) you are using. Then follow the examples below to startup Internet-Connect.

8.1 Packet Driver

Note: The following files will need to be modified for this configuration. Be sure to save the existing versions of the files before proceeding with the set up:

autoexec.bat

Core Systems Winsock and TCP/IP stack is designed to run with packet drivers (via VXDWSA.386). If you are currently using packet driver, you should not have to modify your current networking setup or configuration files to run Internet-Connect. After completing **step 1** above, start up Windows then click on **wping** to try out Winsock.

If you are setting up a new network and wish to use packet driver as your LAN driver, you need to load and start the packet driver TSR before starting Windows. Assuming you are using WD8013 Ethernet Adapter, the following sample entries in autoexec.bat shows how this is done.

```
wd8003.com 0x60 7 0x2a0 0xd000
win
```

(The wd8003 line is to load the SMC 8003 Ethernet Adapter driver. This driver is in the Crynwr packet driver package, available from oak.oakland.edu, and many other Internet archive sites).

8.2 NDIS driver (LAN Manager, Windows for Workgroup)

Note: The following files will need to be modified for this configuration. Be sure to save the existing versions of the files before proceeding with the set up:

autoexec.bat
config.sys
protocol.ini

If you are running LAN Manager^(TM) or Windows For Workgroup^(TM), you are probably using a NDIS driver. The following configuration files show a sample of setup required to run Core Winsock with NDIS driver and LAN Manager.

autoexec.bat

```
netbind
win
```

config.sys

```
device=C:\WFW\protman.dos /i:C:\WFW
device=C:\WFW\workgrp.sys
device=C:\WFW\smcmac.dos
device=dis_pkt9.dos
```

(Note the last line, it loads the Packet Driver-to-NDIS converter, which enables software using packet driver interface, such as Core Winsock, to run with NDIS driver. This is a public domain software, available from many Internet archive sites).

NDIS protocol.ini

```
[MS$W13EP]
DriverName=SMCMAC$

[pktdrv]
drivername=pktdrv$
bindings=MS$W13EP
intvec=0x61
```

8.3 ODI driver (Novell Netware)

Note: The following files will need to be modified for this configuration. Be sure to save the existing versions of the files before proceeding with the set up:

autoexec.bat
net.cfg

If you are using Netware or Netware Lite, you are probably using an ODI driver. The following configuration files provide a sample of setup required to run Core Winsock with ODI driver and Netware.

^(TM) Product names are Registered Trademarks of their respective manufacturers

autoexec.bat

```

C:\NETWARE\lsf
C:\NETWARE\wdplus
C:\NETWARE\ipxodi A
C:\NETWARE\netx (/ps=SERVERNAME)
odipkt 1 96

```

(Note the last line, it loads the Packet Driver-to-ODI driver converter, which enables software using packet driver interface, such as Core Winsock, to run with ODI driver. This is a public domain software, available from many Internet archive sites).

net.cfg

```

Link support
    buffers 6 1600

# Setup the WDPLUS card
Link driver WDPLUS
    INT 7
    MEM D0000
    PORT 2A0
    envelope type ETHERNET_802.3
    envelope type ETHERNET_II
#    envelope type ETHERNET_802.2
#    envelope type ETHERNET_SNAP

Protocol IPX
    Bind wdplus

```

9. How to setup a gateway configuration

Internet-Connect provides multiple interface configuration support as well as IP forwarding/routing capability. To setup your machine as a gateway, from Network Configuration, enable both Ethernet and Serial interfaces. Click on More... button. Check IP forwarding box. To enable other machines on your network to make use of the gateway capability of your machine, make sure that the gateway addresses for these machines point to your machine IP address (e.g. if they are Unix machines, enter a route add ... command)

10. Internet-Connect Network Configuration Utility (wsaconf.exe)

This utility maintains the network configuration of your machine. When in doubt about your configuration, start this utility. All settings are stored in core.ini. Peruse this file if you are curious, however do use this utility to change any settings in core.ini file.

11. Core.ini

Core.ini is the initialization file for all components of Internet-Connect: winsock.dll, coretcp.dll and associated utilities: wsaspy, wping, wfinger, wsacore. This file must be in Windows directory or in one of directories defined in **PATH**.

(Note: This file is used by Internet-Connect to determine your network configuration. You should not have to modify this file directly. The Network Configuration Utility (wsaconf.exe) should be used to view and modify this file).

Core.ini follows the format of Windows **.ini** files, that is, sections and entries have following form:

```
[section name]
keyword=value
```

Core.ini currently contains the following sections and keywords:

```
[winsock]
; name of this host
hostname=charlie_brown

; if use DNS for name resolution set the following to yes (default is no)
;usedomainnameserver=yes

; if use NIS for name resolution set the following to yes (default is no)
;usenisnameserver=yes

; domain this host belongs to
domainname=peanuts

; if use DNS or NIS, enter ip addresses of name servers
nameservers=193.46.1.2,193.46.1.5
```

```
; to disable winsock background monitor set the following to no
; (default is yes)
;usewinsockmonitor=no
```

```
; to substitute your own winsock monitor (instead of the one supplied),
; set its name here
customwinsockmonitor=c:\corewsds\wsacore.exe
```

```
; full pathname of hosts, services and protocol files
; note that if name server is used, hosts file is not required. In
; addition if internal services and protocol databases are used (see below),
; pathnames for corresponding external services and protocol files are
; not needed
hostsdb=c:\corewsds\hosts
;servicesdb=c:\corewsds\services
;protocolsdb=c:\corewsds\protocol
```

```
; CORE winsock contains built-in services and protocol files,
; hence obviating the need for external services and protocol files. If
; you do not wish to use the built-in files, set the following keywords
; to no (default is yes)
;useinternalservdb=no
;useinternalprotodb=no
; if use bootp or DHCP server set it to yes (default is no)
;usebootp=yes
useinternalservdb=yes
useinternalprotodb=yes
usenisnameserver=no
usedomainnameserver=no
usehostsfileanddns=yes
usebootp=no
```

```
[tcpip]
licensenum=7W-9767-04B09F1
```

```
; ip address of this host
; if bootp or DHCP server is not used
ipaddress=193.46.1.3
```

```
; network mask
netmask=255.0.0.0
gateways=193.46.1.2,193.46.1.5
ipforwarding=yes
actasgateway=yes
subnetsarelocal=yes
; specify your packet filter library here (useful for gateway only)
;ipfilterlibrary=c:\corewsds\ipfilter.dll
```

```
[wsacore]
;debug=yes
visible=yes
;usepolling=no
; number of secs of inactivity before exiting (0 means never timeout)
inactivetimeout=1
;inactivetimeout=0
;testing=yes
```

(TM) Product names are Registered Trademarks of their respective manufacturers

```
;log=no
datatrace=no
tcptrace=no
udptrace=no
icmptrace=no
```

```
[wping]
hostname=193.46.1.5
verbose=yes
debug=no
dontroute=no
iterations=1
packetsize=300
showmessagewindow=yes
twowindows=no
logfilename=C:\COREWSDS\WPING.log
backgroundcolor=65280
textcolor=16711680
```

```
[wfinger]
hostname=lucy
username=fred
backgroundcolor=255
textcolor=16777215
logfilename=C:\COREWSDS\WFINGER.log
```

```
[ethernet_0]
name=ethernet0-pktdrv
; IP address of interface (if different from local address above)
enabled=yes
localipaddr=193.46.1.3
netmask=255.0.0.0
mtu=1500
```

```
[serial_0]
;name=slip-interface
;name=ppp-interface
; local IP address, set to 0 if dynamically assigned by host
localipaddr=0.0.0.0
; remote IP address, set to 0 if dynamically assigned by host
remoteipaddr=0.0.0.0
enabled=yes
mtu=256
name=ppp-interface
; do not change this
portmonitor=c:\corewsds\wsaslip.exe
; VJ compression, dynamic means support whatever mode used by host
;compressed=dynamic
compressed=yes
sendicmp=yes
port=com1
protocol=ppp
; Set to yes to accept incoming connection (i.e. run as server)
server=yes
```

```
netmask=255.0.0.0
inactivetimeout=300
```

(TM) Product names are Registered Trademarks of their respective manufacturers

; other PPP-related parameters

```

asyncmap=0xffffffff
; userid and password to send to remote if remote uses UPAP
; if not set here, will prompt at connection time
myuserid=my_userid_here
mypassword=my_password_here

; PPP negotiated options
; if run as server, require remote to send in user-id and password
pppopt_upap=no
; do VJC negotiation as defined by old RFC1172 (option 2)
pppopt_rfc1172vjc=no
; do IP address negotiation as defined by old RFC1172 (option 1)
pppopt_rfc1172ipaddr=no
pppopt_cidcompression=no
pppopt_ipaddr=yes
pppopt_vjcompression=no
pppopt_upap=no
pppopt_mru=no
pppopt_accompression=no
pppopt_pcompression=no
pppopt_asyncmap=no
pppopt_magicnumber=no
pppopt_startmode=passive

```

```

[ppp_server_users]
; list of userids allowed to connect to this machine
; userid=password,localipaddr,remoteipaddr
my_user_id=TCwXjXAFGK6,193.47.1.3,193.47.1.5

```

```

[ppp_server_messages]
loginsuccessmsg=Welcome to Internet-Connect (TM) PPP Server
loginfailmsg>Login to Internet-Connect (TM) PPP Server failed

```

```

[wsaslip]
visible=yes
testing=no
; turn these off to maximize performance
debug=no
trace=no
inactivetimeout=1
defaultport=com1
;dialer=c:\corewsds\wsadial.exe
log=no
; login script
scriptfile=c:\corewsds\wsaslip.scr
; hangup script
;hangupscriptfile=c:\corewsds\hangup.scr

```

```

[wsaslip_com1]
enabled=no
;connect=modem
connect=direct
flowcontrol=rtscts
parity=None

```

(TM) Product names are Registered Trademarks of their respective manufacturers

```

stopbits=1
xonxoff=no
baudrate=9600
autoreconnect=yes
databits=8

```

```

[wsaslip_com2]
enabled=yes
;connect=modem
connect=direct
flowcontrol=rtscts
parity=none
stopbits=1
xonxoff=no
baudrate=9600
autoreconnect=yes
databits=8

```

```

[wsaslip_com3]
enabled=yes
connect=modem
flowcontrol=rtscts
parity=none
stopbits=1
xonxoff=no
baudrate=2400
databits=8

```

```

[wsadial]
InitString=AT&F&D0
DialPrefix=ATDT
DialSuffix=
HangUp=ATH0
Connect1=CONNECT 1200
Connect2=CONNECT 2400
Connect3=CONNECT 9600
Connect4=CONNECT 14400
Fail1=NO CARRIER
Fail2=BUSY
Fail3=VOICE
Fail4=NO ANSWER
flowcontrol=rtscts
CarriageReturn=CR
ApplicationName=c:\corewsds\wsaslip.exe
ApplicationDelay=5
DialOnStartup=NO
Trace=NO
Debug=NO
ExitOnConnect=NO
DialEntry=0
Log=NO

```

```

[wsatty]
NewLine=NO
LocalEcho=NO
AutoWrap=YES
Port=3

```

(TM) Product names are Registered Trademarks of their respective manufacturers

BaudRate=-236
 DataBits=8
 FlowControl=2
 Parity=0
 StopBits=0
 XonXoff=NO
 ForegroundColor=16711680
 UseCNReceive=YES
 DisplayErrorMessages=YES

[ethernet_1]
 name=ethernet1-pktdrv
 localipaddr=0.0.0.0
 netmask=255.0.0.0
 mtu=1500
 enabled=no

[serial_1]
 name=slip1-interface
 localipaddr=0.0.0.0
 remoteipaddr=0.0.0.0
 netmask=255.0.0.0
 mtu=1500
 enabled=no
 inactivetimeout=300
 compressed=yes
 server=no
 sendicmp=yes
 protocol=slip
 portmonitor=c:\corewsds\wsaslip.exe

[serial_2]
 name=slip2-interface
 localipaddr=0.0.0.0
 remoteipaddr=0.0.0.0
 netmask=255.0.0.0
 mtu=1500
 enabled=no
 inactivetimeout=300
 compressed=yes
 server=no
 sendicmp=yes
 protocol=slip
 portmonitor=c:\corewsds\wsaslip.exe

[serial_3]
 name=
 localipaddr=0.0.0.3
 remoteipaddr=0.0.0.0
 netmask=255.0.0.0
 mtu=1500
 enabled=no
 inactivetimeout=300
 compressed=yes
 server=no
 sendicmp=yes
 protocol=ppp

(TM) Product names are Registered Trademarks of their respective manufacturers

```
portmonitor=c:\corewsds\wsaslip.exe
```

```
[wsaslip_com4]
enabled=yes
connect=direct
flowcontrol=rtscts
parity=no
stopbits=1
xonxoff=no
baudrate=9600
databits=8
```

```
[DialDirectory]
Entry0=violet,1-908-4772788,COM1,9600,N,8,1,c:\corewsds\wsadial.scr,Modem
Entry1=linus-slip,,COM2,9600,N,8,1,,Direct
```

12. Internet-Connect Monitor (wsacore.exe)

Wsacore is part of Core Winsock stack. It serves as background monitor for the functioning of winsock.dll, coretcp.dll, and carry-outs functions which cannot be done easily by a DLL. **Wsacore** also captures and logs messages generated by winsock.dll and coretcp.dll. These messages are written to the file **wsacore.log** which resides in the same directory as **wsacore**. If debug is enabled (**debug=yes** in **core.ini**, **wsacore** section), debug messages will be written to the file **wsacore.trc** in same directory.

Wsacore starts up automatically by Internet-Connect when the first winsock application starts, and ends when the last application exits. It provides a comprehensive collection of statistics and traces about the internal operation of the network (TCP/IP packet statistics, packet trace, protocol analysis, currently active sessions and applications, interface status).

Wsacore is normally invisible to Internet-Connect user. To make it visible, click on **More...** button on **Network Configuration** (wsaconf.exe) dialog, then check **Make Wsacore Visible** box.

13. Wsadial

Internet-Connect provides two dialers. One is a built-in dialer and is part of the SLIP/PPP Driver (**wsaslip**). The other dialer is **wsadial**, a standalone, generic dialer which can be used with or without Internet-Connect stack.

The built-in dialer is implemented within Internet-Connect SLIP/PPP driver (**wsaslip**). It has scripting capability, autodial, redial, hangup and many other features. For most configurations, the built-in dialer is sufficient to serve the purpose. To instruct Internet-Connect to use the built-in dialer, leave the **Dialer** field of the **Network Configuration/Serial Interface** dialog blank.

The second dialer is wsadial.exe. This is a standalone dialer. It contains scripting capability, autodial, redial, hangup, application startup, dialing directory, and many other features. This dialer is designed as a generic dialer. You may use it with Internet-Connect or other applications. As an example of the latter, click on wsadial to dial and login to a remote host, then start up an application (or specify wsadial to do so) such

(TM) Product names are Registered Trademarks of their respective manufacturers

as a terminal emulation program (wsatty.exe, for example) to exchange data with the remote host.

Wsadial can be used as an integrated part of Internet-Connect. To do so, set the **Dialer** field of the **Network Configuration/Serial Interface** dialog to **wsadial.exe** full path name. Internet-Connect will automatically startup **wsadial** when connection is to be made to a remote host. Note that when **wsadial** is used in this manner, Internet-Connect will automatically make sure that both **wsadial** and the stack are sharing the same configuration data, such as port number, baud rate. What this means is you only have to maintain configuration information in one place, using the Network Configuration Utility (**wsaconf**).

You can also specify a script for **wsaslip** to execute after it has successfully dialed in to the remote host. To do so, start up **wsadial**, then click **Settings**, and fill in path name of the script file in **Login Script** field. For a sample **wsadial** script, see **wsadial.scr** file. Note that this script file is separate from the **SLIP/PPP Driver (wsaslip)** script file, which is set in **Script File field of Network Configuration/Serial Interface (wsaconf)** dialog. If you have set both script fields, both scripts will be executed, one by **wsadial** (after dialing), and the other by **wsaslip**.

Internet-Connect can make use of a third-party standalone dialer. If you have a favorite dialer program, and do not wish to use Internet-Connect built-in dialer, or **wsadial**, just set the path name of your dialer in the **Dialer** field of the **Network Configuration/Serial Interface** dialog. Please make sure that configuration data of your dialer (port settings, baud rate, etc.) match the Internet-Connect configuration.

Tips: If your modem keeps dropping the connection shortly after **wsadial** has successfully made the connection, try adding command **&D0** to the modem initialization string (e.g. **AT&F&D0V1**). Without this command, some modems drop the connection when **wsadial** relinquishes the port (without dropping DTR signal) to another task.

14. Wsapy

Wsapy is a utility designed to capture and log a particular or all winsock function calls. This is a standalone utility. It requires no code change to application under trace, and works with any **winsock.dll** conforming to version 1.1 of the Specification. **Winsock API trace** can be stored in circular buffer in memory (for minimum performance impact) to be written out to file (**wsaspy.trc**) on user's command at later time, or dynamically displayed on screen and continually written out to log file (**wsaspy.log**) (this has some performance impacts). Wsapy log files (**wsaspy.log & wsaspy.trc**) are in the same directory as **wsaspy.exe**.

15. Wping

Wping is Windows version of the popular TCP/IP ping program. It is designed to probe for connectivity of a remote host on the network.

16. Wfinger

A Windows implementation of the popular UNIX finger program..

17. How to install your own IP packet filter

When running as a gateway (i.e multiple interfaces), Internet-Connect provides a hook where you can install your own program to monitor and filter packets passed between interfaces. This feature can be used to setup up your own firewall gateway, or to monitor incoming traffic for accounting or performance measurement purposes. Once installed, the filter program has complete access to the data packets (including IP headers) as they are moved between interfaces. The filter program may copy the packet, modify its contents and instructing the stack to forward or drop the packet.

The filter program must be compiled, linked and installed as a DLL. To install a filter, set its complete pathname in the **IP Filter Library** field of the **Core Winsock Configuration/More** dialog.

A sample filter program is provided (ipfilter.c, ipfilter.h, ipfilter.def files) as a guide for writing your own filter program.

18. Online help.

Winsock API Reference is available online. To invoke Help, click on **winsock.hlp** icon in **Core Winsock** Program Group, or the Help menu of any Internet-Connect utility.

19. Report Problems.

A problem report template (**problem.wri**) is included with this package. Please use this form to report any problems encountered. You may also use this form to send in comments or suggestions on our product. We welcome any feedbacks you can provide to help us improve Internet-Connect. Please send the completed form to Core Systems:

**Core Systems
PO Box 31022
Walnut Creek, CA 94598.
(510) 943-5765**

or email to:

lvuong@cais.com or 71552.3666@compuserve.com.

20. Trouble shooting tips.

Internet-Connect runs fine with Windows 3.1, however fails with WFW 3.11

First, If you are running Windows/Windows For Workgroup 3.11 please add the following line to autoexec.bat:

xx:\corewsds\winpkt.com yy zz

where:

xx: drive where Internet-Connect is installed.

yy: software interrupt value to be used by winpkt (e.g. 0x60)

zz: software interrupt value being used by existing packet driver or equivalent shim (e.g. 0x61)

*Next, as **Windows for Workgroup 3.11** requires a slightly different networking setup than that of WFW 3.1, run network setup, select unlisted or updated driver in the driver dialog, then enter the drive containing Internet-Connect installation floppy and follow network setup dialog instructions.*

System hang.

Probable cause:

Mismatched winsock.dll and TCP/IP protocol stack.

*Incorrect **winsock.dll** is loaded (more than one winsock.dll in your **PATH**)*

core.ini** not initialized properly or not in **PATH

Winsock.dll cannot find vxdwsa.386.

Probable cause:

***vxdwsa.386** failed to load*

device=c:\corewsds\vxdwsa.386** entry missing from **system.ini

***vxdwsa.386** is not in correct path*

Winsock.dll cannot find packet driver.

Probable cause:

***packet driver** TSR or appropriate **shim** failed to load*

***packet driver** TSR or **shim** is loaded at incorrect software interrupt, or*

*You are running **Windows for Workgroup 3.11** which requires a slightly different networking setup than that of WFW 3.1. Run network setup, select unlisted or updated driver in the driver dialog, then enter the drive containing Internet-Connect installation floppy.*

Your packet driver or equivalent shim is loaded in high memory. At start-up time, Internet-Connect scans memory to locate a packet driver. This method does not work if the driver is loaded in high memory. In which case, add the following lines to your system.ini file:

[vxdwsa]

pdint=nn

where nn is the software interrupt value (in decimal) used by your packet driver. This

value is typically in the range of 96 to 111.

Cannot startup any network application.

Probable cause:

***Name server** is not running*

***Hosts** file is missing*

***IP address** or **local host name** is not set*

Can start application but cannot establish communication.

Probable cause:

Host is not responding.

*Incorrect packet driver name or binding in **protocol.ini***

*Incorrect **Ethernet frame type** in **net.cfg***

21. Network configuration databases

\corewsds\hosts

\corewsds\services

\corewsds\protocol

\corewsds\resolver.cfg

22. Shareware and Public Domain software

The following packages are shareware or public domain software. They provide features which complement **Internet-Connect**. They are not developed or marketed by Core Systems. Please refer to notes included with each package for information and rules governing their use.

Crynwr public domain packet driver collection (Current version is Version 11).

QPC Software's QVTNet shareware package: Winsock-based Internet utilities (telnet, ftp, rcp, lpr).

Various public domain shims: odipkt.com, dis_pkt9.dos.

NCSA Mosaic for Windows.