Installing and Configuring Novell's Token-Ring Source Routing Drivers

David Rosenfeld
Senior Systems Engineer
Systems Engineering Division

In a Token-Ring network with source-routing bridges, NetWare nodes often need to communicate with nodes on another ring. This AppNote explains how to install and configure the appropriate Novell source routing driver (ROUTE.VP0, ROUTE.NLM, ROUTE.COM, or ROUTE.SYS) to enable communication across source-routing bridges. These drivers provide source routing capability for all types of NetWare workstations, routers, and servers, and can be used with the IPX, AppleTalk, and TCP/IP network protocols.

Contents

NetWare and Source Routing

If you have a NetWare network running on Token-Ring hardware, and you are doing source routing, you need to use the source routing drivers supplied by Novell. These drivers are supplied with NetWare v2.x and v3.x. They provide source routing capability for DOS (and Windows 3.0), OS/2, Macintosh, and UNIX workstations using the IPX, AppleTalk, and TCP/IP network protocols.

This AppNote explains when to use Novell's source routing drivers, what the drivers do, and how to install them and configure them on NetWare servers, external routers, and workstations. The information presented in this AppNote is geared specifically to the current releases of the Novell drivers, and may not apply to previous versions.

When to Use Source Routing Drivers

Source routing is needed when *all* of the following conditions are met:

- The network is using Token-Ring.
- There is more than one ring.
- The multiple rings are connected by source-routing (in most cases, IBM) bridges.
- NetWare servers or external routers are not used to connect these rings.
- A node (workstation or server) on one ring needs to communicate with a node on a different ring.

Figure 1 illustrates an example source routing scenario. For the DOS, OS/2, Macintosh, and UNIX workstations to access files from either NetWare server shown, source routing is required.

What the Novell Source Routing Drivers Do

Why do you need a separate source routing driver? The answer lies in the fact that source routing information in a Token-Ring packet is optional. It is only required when a Token-Ring frame must traverse a source-routing bridge.

Figure 1: Source routing is needed in this network consisting of various workstations, a source-routing bridge, and two NetWare servers.

Figure 2 shows the structure of a Token-Ring frame. If a Token-Ring frame needs to include source routing information, the Routing Information Indicator in the MAC header's Source Node Address field is set to 1. This indicates to the workstation, server, or router that there will be source routing information located in the Routing Information field.

By default, the current Novell LAN drivers do *not* fill in the source routing information fields. When source routing information must be provided, the ROUTE.*xxx* drivers from Novell supply the data which is inserted into the Routing Information field.

Figure 2: A Token-Ring frame includes a Routing Information Indicator in the Destination Address field and other routing information for source routing.

To support source routing, Novell's ROUTE.*xxx* software maintains a table of discovered routes to other nodes. (The actual route discovery is accomplished by sending *data* frames as either all-routes or single-route broadcasts. For more information about these broadcasts and how they are used for route discovery in a Token-Ring network, see "Understanding Token-Ring Source Routing" in the May 1991 *NetWare Application Notes*.)

It is also ROUTE's responsibility to take the source routing data from its tables and insert it into the Token-Ring frame. Since ROUTE automatically supplies the source routing information for all frames transmitted by NetWare, the source routing drivers work transparently to NetWare and any network applications.

Note that a file server with ROUTE loaded will still support IPX, AppleTalk, or TCP/IP packets from workstations where ROUTE.COM or ROUTE.SYS is not installed. The source routing information is used only to cross source-routing bridges. However, avoid mixing source routing and non-source routing nodes. If you enable source routing on one node in a ring, enable it on all of the ring's nodes.

Files Used to Enable Source Routing

Four versions of ROUTE.*xxx* are used in conjunction with Token-Ring drivers to support source routing:

- ROUTE.VP0 (for NetWare v2.x servers)
- ROUTE.NLM (for NetWare v3.x servers)
- ROUTE.COM (for DOS workstations)
- ROUTE.SYS (for OS/2 workstations)

ROUTE.VP0 comes with NetWare v2.2; ROUTE.NLM comes with NetWare v3.11; ROUTE.COM is on the WSGEN (previously SHGEN) disk; and ROUTE.SYS is included with the NetWare Requester for OS/2. (The TOKENEE.SYS and TOKEN.SYS drivers shipped with previous version of the OS/2 Requester also support source routing. However, you should upgrade to the latest versions.)

The latest drivers may be acquired from Novell dealers or from Novell's NetWire forum on CompuServe. As of this writing, these are the latest versions of the Token-Ring and source routing drivers.

ROUTE.COM (DOS)	version 1.12
ROUTE.SYS (OS/2)	version 1.x
ROUTE.VP0 (NetWare v2.2 Server)	version 1.02
ROUTE.NLM (NetWare v3.11 Server)	version 3.12
TOKEN.LAN (NetWare v3.11 Server)	version 3.15
ATOKEN.OBJ (NetWare v2.2 Server)	version 2.62
STOKEN.OBJ (DOS - WSGEN)	version 2.62
SLANSUP.OBJ (DOS - WSGEN)	version 2.63
TOKEN.COM (DOS - ODI)	version *
LANSUP.COM (DOS - ODI)	version *

*The latest ODI drivers are available in the ODIUP2.ZIP file on NetWire. Likewise, the latest DOS drivers are available in a file called TOKENB.ZIP.

Where ROUTE Is Used by NetWare Servers and Clients

Source routing is needed for several types of communication from workstations and servers. For example, workstations require source routing in the following situations:

● Issuing Get Nearest Server requests (loading NETX.COM)
● Issuing Get Local Target requests (during attach or login)
● Finding a new route to a node (when a bridge has gone down)
● Inserts route, if known, into outgoing packets (sending data packets between nodes)

The file server requires source routing capabilities for:

● Maintaining IPX router tables with the Routing Information Protocol (RIP)—ring segment numbers are not kept in the router table
● Advertising the availability of services, such as file services, through the Service Advertising Protocol (SAP)
● Maintaining routes to connected workstations

Configuration Considerations

There are some configuration considerations that must be addressed when you want to use both Novell routers and source-routing bridges on the same network.

First, do not confuse the Token-Ring ring number (assigned during the configuration of a source-routing bridge) with the network address (assigned when configuring NetWare servers). A ring number is a three-digit hexadecimal number assigned in the bridge configuration program (IBM calls these as ring *segment* numbers). Each ring number on a Token-Ring network must be unique. However, all NetWare file servers and routers connected through source routing Token-Ring bridges must use the *same* network address. (The internal IPX number of each NetWare v3.x server must be unique, however.)

Source-routing bridges are transparent to NetWare. In other words, NetWare views rings that are linked through source-routing bridges as one logical ring. The appropriate IPX network addressing procedure is illustrated in Figure 1.

Second, note that IPX packets (used by Novell's Network-layer protocol) and source-routed Token-Ring frames are *not* two different "types" of packets. A packet transmitted onto a Token-Ring network contains both Token-Ring (including source routing) *and* IPX information. The IPX protocol information is actually enveloped within the Token-Ring frame. The source routing information is contained entirely within the MAC header.

Third, source-routing bridges connect Token-Ring rings. To do this,

the source-routing bridges use information from the Token-Ring frame. Most source-routing bridges cannot connect Token-Ring with other network topologies, such as Ethernet or Arcnet. (IBM's 8209 bridge connects Token-Ring and Ethernet networks, but it does no source routing on the Ethernet side. IBM also provides a Token-Ring to PC Network II bridge.)

Novell servers act as routers, which can connect different topologies. To accomplish this, NetWare uses information from the IPX header. For those familiar with the OSI seven-layer model, source-routing bridges work at the Data Link layer, while IPX routing operates at the Network layer, one layer above. The important point is that even though IBM calls it source *routing*, it is actually a *bridging* protocol.

Keep this distinction in mind when configuring a network that has both source-routing bridges and Novell routers. Do not connect two rings with both a source-routing bridge and a Novell router in parallel.

If you use source-routing bridges in parallel (as a way to create alternate paths and make the network more resistant to failures), be sure the bridges are in automatic mode. This will prevent broadcast storms, and provide a backup routing path in case one bridge fails. (For more information about parallel bridges, refer to the manual provided with the source-routing bridge. IBM's *Token-Ring Architecture Reference* is another good resource.)

The following sections give specific instructions for loading source Novell's routing drivers on the various types of network nodes: NetWare v2.2 and v3.11 file servers, and DOS, OS/2, Macintosh, and UNIX clients.

Source Routing on a NetWare v2.2 Server or Router

NetWare v2.2 is Novell's workgroup oriented network operating system which runs on an 80286-based PC or better. Applications running on the file server are called Value-Added Processes (VAPs). The external router is a standalone PC which routes IPX traffic. An external router provides no file services, but can execute VAPs. Thus, source routing is loaded as a VAP.

Follow the procedure below to implement source routing on a NetWare v2.2 server or router:

1.   To configure the source routing VAP (ROUTE.VP0) before it is loaded, use SRCONFIG. The defaults for ROUTE.VP0 will work under most configurations, so SRCONFIG is rarely needed. Parameters can also be set at the server or router console after ROUTE.VP0 is loaded. (See Step 5 below.)

2.   Copy the ROUTE.VP0 file into the SYS:SYSTEM directory for a NetWare v2.2 server, or the boot directory for an external router.

3.   Down the NetWare v2.2 server or router.

4.   Restart the server or router; the VAP will load automatically. If the

server does not have enough RAM, the VAP will either not load at all or load but not function properly. Look for the console message "Source routing has been enabled on LAN A" (or LAN B, C, or D). This message is your only indication that the VAP has loaded successfully.

5. If any parameters need changing, you can use the following command format at the console command line:

```
ROUTE [board=x, clear, def, gbr, mbr, remove=hhhhhhhhhhhh,
rsp=rr, time=sss]
```

The "ROUTE Command-Line Options" section contains detailed descriptions of these parameters. Entering "ROUTE ?" at the console prompt will display a help panel explaining the parameters.

Source Routing on a NetWare v3.11 Server

NetWare v3.11 is Novell's enterprise-oriented network operating system which runs on an 80386-based PC or better. Applications running on a NetWare v3.11 server are called NetWare Loadable Modules (NLMs). Source routing is implemented on a NetWare v3.11 server as an NLM called ROUTE.NLM.

For IPX communication, a LAN adapter driver that supports source routing, such as TOKEN.LAN, must be loaded before ROUTE.NLM. (For simplicity, we'll use the IBM Token-Ring adapter as an example.) If there is more than one Token-Ring adapter using source routing in the server, ROUTE.NLM must be loaded and numbered for each adapter. You can load ROUTE.NLM with the parameters described under "ROUTE Command-Line Options."

The next two sections show examples of loading ROUTE.NLM for NetWare v3.11 servers that support other network protocols; in particular, AppleTalk and TCP/IP. If you have a server with one Token-Ring adapter supporting *both* AppleTalk and TCP/IP, be aware that ROUTE is only loaded twice—once for each frame type.

AppleTalk and Source Routing

AppleTalk and source routing are supported under NetWare. Figure 3 displays a sample AUTOEXEC.NCF file for configuring a NetWare v3.11 server supporting both IPX and AppleTalk.

Figure 3: Sample AUTOEXEC.NCF file for a NetWare v3.11 server using source routing with both IPX and AppleTalk.

```
File Server Name EXAMPLE_SERVER_1
IPX internal net DABAD00
Mount all
REM
REM   Load IBM 16/4 Token-Ring Adapter with
REM   IPX and IBM Source Routing
REM
load TOKEN int=2 port=320 frame=TOKEN-RING name=TRING
load ROUTE board=1
bind IPX to TRING net=1
REM
```

```
REM    Load IBM 16/4 Token-Ring Adapter with AppleTalk
REM
load APPLETLK -z net=10001
load TOKEN frame=TOKEN-RING_SNAP name=TOKEN-SNAP
load ROUTE board=2
bind APPLETLK to TOKEN-SNAP -z net=1-3
load ATTOKLLC
bind ATTOKLLC to TRING
```

This example also demonstrates a special case where an Apple Internet Router is also being used on the network. Specifically, when the AppleTalk router in NetWare is on a multi-ring Token-Ring network on which Apple Internet Routers and source-routing Token-Ring bridges are present, it requires a special NLM named ATTOKLLC.NLM to communicate with Apple Internet Routers.

Why do you need ATTOKLLC.NLM? The Apple Internet Router sends an 802.2 Test/Poll command to a node when it needs to discover the Token-Ring route (across bridges) to that node. It waits to receive an 802.2 Test/Final (reply) frame, which will contain the Token-Ring route information, before sending AppleTalk frames to the node. The AppleTalk stack/router NLM does not respond to the Test/Poll command. Therefore, the ATTOKLLC NLM, which listens for and responds to these Test/Poll frames, must be installed on the NetWare v3.11 server.

ATTOKLLC.NLM is not installed automatically during the NetWare for Macintosh v3.0 installation. To install this NLM, you must manually copy ATTOKLLC.NLM from the NW-MAC diskette to the directory containing the NetWare for Macintosh product (normally the SYS:SYSTEM directory). You should then load the NLM and bind it to the Token-Ring board that handles the "TOKEN-RING" frame type, as described below.

After copying the NLM to the SYSTEM directory, follow these steps, which are illustrated in Figure 3:

1.   Load the frame type "TOKEN-RING" for the Token-Ring LAN driver being used for ATTOKLLC.NLM. (Note that this is the default frame type used when the driver is loaded. AppleTalk requires the "TOKEN-RING_SNAP" frame type, as in Figure 3, line 15. If you are running IPX and APPLETLK over the same Token-Ring, the "TOKEN-RING" frame type is already loaded, as in Figure 3, line 8.)

2.   Load the Token-Ring source routing NLM (ROUTE.NLM) for the Token-Ring adapter, specifying the TOKEN-RING frame type. Note that this is *not* the same logical board to which APPLETLK is bound. (Again, if you are running IPX and APPLETLK over the same Token-Ring adapter and you have loaded ROUTE for that adapter, this step has already been completed. See Figure 3, line 9.)

3.   Load ATTOKLLC.NLM. There are no options for this NLM (see Figure 3, Line 18).

4.   Bind ATTOKLLC to the Token-Ring board handling TOKEN-RING frames. Note that this is *not* the same logical board to which APPLETLK is bound. (See Figure 3, Line 19.)

TCP/IP and Source Routing

NetWare also supports source routing for the TCP/IP protocol. Figure 4 shows the configuration for supporting IPX and TCP/IP on a NetWare v3.11 server.

Figure 4: A sample AUTOEXEC.NCF file for a NetWare v3.11 server supporting source routing with IPX and TCP/IP.

```
File Server Name EXAMPLE_SERVER_2
IPX internal net DABAD002
Mount all
REM
REM     Load IBM 16/4 Token-Ring Adapter with
REM     IPX and Source Routing
REM
load TOKEN int=2 port=320 frame=TOKEN-RING name=TRING
load ROUTE board=1
bind IPX to TRING net=2
REM
REM     Load IBM 16/4 Token-Ring Adapter with TCP/IP Support
REM
load TCPIP
load TOKEN frame=TOKEN-RING_SNAP name=TOKEN-SNAP
load ROUTE board=2
bind IP to TOKEN-SNAP addr=129.57.128.125
```

Note that the TCP/IP protocol is using a different frame type (TOKEN-RING_SNAP) than IPX. Also, ROUTE must be loaded twice, with different board numbers, even though there is only one Token-Ring adapter in the NetWare v3.11 server. You can include various options when loading ROUTE; these are explained under "ROUTE Command-Line Options" below.

Source Routing at a DOS/Windows Workstation

Whenever traffic from a DOS PC workstation needs to cross a source routing bridge to reach a NetWare server, ROUTE.COM must be loaded. ROUTE.COM, like the rest of the NetWare shell, will support all DOS versions, plus Microsoft Windows. Copy the ROUTE.COM file from the *DOS/ODI Workstation Services* diskette to the directory from which the NetWare drivers and shell are executed.

Here is the loading order for the DOS shell components with drivers created with the SHGEN or WSGEN process:

> IPX
> ROUTE
> NETX

Note that the IPX.COM created from WSGEN will support only IPX packets (no TCP/IP). No special entries are required in the SHELL.CFG (shell configuration) file.

If you are using ODI drivers, load the drivers in the following sequence:

```
LSL
TOKEN*
ROUTE
IPXODI
NETX
```

To include support for TCP/IP (if you are using Novell's LAN Workplace products, for example), load the following software in this order:

```
LSL
LANSUP*
ROUTE
ROUTE BOARD=2
IPXODI
NETX
TCPIP
```

*Note that, in this second example, the LANSUP driver is used in place of the TOKEN driver used in the first example. Actually, you can use either LANSUP.COM or TOKEN.COM, depending on what other software is running in the workstation. LANSUP is used in conjunction with the IBM LAN Support Program, which must be loaded in the CONFIG.SYS file. If a user needs to access an application which requires IBM's LAN Support Program, use the LANSUP driver; otherwise use TOKEN. ROUTE.COM will work equally well with either Token-Ring driver.

Also note that for TCP/IP, ROUTE is loaded a second time with a different BOARD number, even though there is only one Token-Ring adapter.

Figure 5 displays the NET.CFG (network configuration) file that supports source routing for both IPX and TCP/IP packets. NET.CFG is a text file containing parameters which will be used instead of, or in addition to, the defaults of the ODI drivers. (For more information on creating a NET.CFG file, refer to the NetWare manuals.)

Figure 5: Sample NET.CFG file for a DOS/Windows ODI workstation using source routing.

```
Preferred Server = EXAMPLE_SERVER_2

Link Support
    Buffers 8 2100
    MemPool 4096

Link Driver LANSUP
    Frame           TOKEN-RING
    Frame           TOKEN-RING_SNAP
    Protocol IPX E0 TOKEN-RING

Protocol IPX
    BIND            LANSUP

Protocol TCPIP
    BIND            LANSUP
    ip_address      129.57.128.255
    ip_router       129.57.128.254
    ip_netmask      255.255.254.00
    tcp_sockets     8
    udp_sockets     8
    raw_sockets     1
    nb_sessions     4
    nb_commands     8
    nb_adapter      0
    nb_domain
```

Note that indentation is a critical part of the NET.CFG file syntax. Be sure to type the lines exactly as shown, with the main headings flush left and the parameters indented at least one space.

You can use various command line options for ROUTE.COM at the DOS client. "ROUTE Command-Line Options" contains detailed descriptions of these parameters. Entering "ROUTE ?" at the console prompt will display a help panel explaining the parameters.

Source Routing at the OS/2 Workstation

The NetWare OS/2 Requester supports source routing. The components of the NetWare Requester are installed in the CONFIG.SYS file, located in the root directory. Source routing capability is added to an OS/2 node by installing ROUTE.SYS in the CONFIG.SYS immediately after loading TOKEN.SYS:

```
DEVICE=C:\NETWARE\TOKEN.SYS
DEVICE=C:\NETWARE\ROUTE.SYS
```

If any defaults need to be altered, two lines must also be added to the NET.CFG file:

```
protocol route
    source route [parameters]
```

The possible parameters are fully described under "ROUTE Command-Line Options."

Source Routing for Macintosh and UNIX

Clients

On a Macintosh or UNIX client workstation, you don't need to install any Novell drivers for source routing. Instead, Novell supports these workstations' native client environments, which handle source routing with their own, built-in drivers.

ROUTE Command-Line Options

Numerous parameters can be entered with ROUTE.*xxx*. All parameters are optional (except for a few cases, explained above), they are not case-sensitive, and they may be entered in any order. Below is a detailed description of these parameters.

Parameters Used at Workstations, Servers, and Routers

BOARD = *x*. For NetWare v2.2, *x* is the LAN number (A, B, C or D) which corresponds to the Token-Ring adapter for which source routing parameters are targeted. If not specified, ROUTE will find the first LAN number that has source routing enabled.

For NetWare v3.11 servers and workstations using the ODI drivers, *x* is a decimal number (1 to 64) that specifies the board number that is the target for the ROUTE command. It must be a previously installed adapter which supports source routing. If not specified, a default of "1" is used. For workstations using the IPX driver created by WSGEN (formerly SHGEN), the "BOARD" parameter is ignored.

The "BOARD" parameter is particularly important when using more than one frame type (for example, TOKEN-RING and TOKEN-RING_SNAP) at a NetWare v3.11 server or on a DOS or OS/2 workstation.

● For NetWare v3.11, ROUTE.NLM must be loaded for each frame type and for each adapter at the node.
● At a DOS or OS/2 workstation, ROUTE.COM or ROUTE.SYS must be loaded twice (or more) using *different* board numbers, even though there may be only one Token-Ring adapter.

This is demonstrated in the workstation example in Figure 3.

CLEAR. Use this parameter to manually clear the source routing table specified by "BOARD=*x*." This will force ROUTE to dynamically rediscover the routes of all nodes by sending a DEFault frame as NetWare addresses each specific node in the network. Use this parameter when a source-routing bridge has gone down and there is an alternate route available.

DEF, GBR, and MBR. These parameters are used to set the broadcast modes for sending and responding. By default, *none* of these parameters specified, which means that all types of source-routed frames are sent as single-route broadcasts. This typically provides the most efficient routing.

● DEF (DEFault frame) tells ROUTE to send any *data* frames with

unknown destination addresses as all-routes broadcasts. If DEF is not specified, then all data packets will be sent as single-route broadcasts.

● GBR (General BRoadcast) tells ROUTE that general *broadcast* frames should be sent as an all-routes broadcast. If GBR is not specified, then all broadcast packets will be sent as single-route broadcasts.

● MBR (Multicast BRoadcast) tells ROUTE that *multicast* frames should be sent as an all-routes broadcast. If MBR is not specified, then all multicast packets will be sent as single-route broadcasts.

It is beyond the scope of this AppNote to provide a detailed discussion of single-route versus all-routes broadcasting. An excellent reference is "Understanding Token-Ring Source Routing," in the May 1991 *NetWare Application Notes*.

REMOVE = *hhhhhhhhhhhh*. This parameter commands ROUTE to remove the specified node address (*hhhhhhhhhhhh* = 12 hexadecimal digits) and its source routing information from the source routing table. This will force ROUTE to rediscover the route the next time a frame is sent to that node.

Parameters Used Only at a Server or Router

RSP = *rr.* This parameter dictates how a server should respond to a request that has been broadcasted. Valid substitutions for *rr* can be:

AR = All routes
SR = Single Routes
NR = Specific Route

The default is NR. For load balancing in a network with parallel bridges, AR is recommended (note that this is different than the default).

TIME = *sss*. In this parameter, *sss* is a decimal number of seconds that specifies when ROUTE.NLM or ROUTE.VP0 should update its source routing table. If a route that has been entered into the table has not been used for this amount of time, the route will be updated when a different route is available. (If the new route involves a larger number of hops than the existing route, however, the new route is discarded.) The default is 3 seconds, and the valid range is from 1 to 255.

This parameter is useful if the network is configured with multiple paths to the same node. By setting this timer appropriately, ROUTE can perform dynamic route discovery if a source-routing bridge goes down and an alternate path is available.

Parameters Used Only at a Workstation

NODES = *ddd*. In this parameter, *ddd* is a decimal number stipulating the number of table entries in ROUTE.COM. The default is 16, and the valid range is from 8 to 255. If *ddd* is less than 8, a value of 8 is used.

TRO = *ddd*. TRO stands for "This Ring Only." This parameter checks locally for the destination server before sending a discovery

packet. TRO determines the number of IPX time-outs before doing a discovery. This avoids putting unnecessary traffic on the LAN.

The default setting for TRO is 0, and the valid range is from 1 to 255. If a client is accessing at least one local server, include the TRO parameter with a setting of 1 or 2. If all servers are being accessed across at least one source-routing bridge, leave it at the default of 0.

TRO specifies the number of IPX broadcasts (for instance, Get Nearest Server requests) that will *not* be source routed. If no local response comes back, then the IPX retry count is incremented. If there is still no local response, the retry count increments again. Eventually, the frame is source routed. It is very important that the TRO setting be *less* than the IPX retry count.

Summary

This AppNote has explained in some detail how to use source routing on a NetWare network. The key point to remember is that if a NetWare workstation or server needs to communicate across a source-routing bridge, the appropriate ROUTE.*xxx* driver must be loaded at each end.

If you would like more information on Token-Ring source routing, refer to the publications listed below.

Robert Perry and Paul Turner, "Understanding Token-Ring Source Routing," *NetWare Application Notes*, Novell, Inc., May 1991, Novell Part Number 164-000030-005

Robert Perry and Paul Turner, "Source Routing and the Spanning-Tree Protocol," *NetWare Application Notes*, Novell, Inc., August 1991, Novell Part Number 164-000030-008

NetWare Installation Supplements, Novell, Inc., March 1991, Novell Part Number 100-000708-002

NetWare Requester for OS/2 Manual, Novell Inc., April 1991, Novell Part Number 100-000921-002

LAN Workplace for DOS Administrator's Guide, Novell Inc., December 1990, Novell Part Number 100-000882-001

NetWare for Macintosh Installation and Maintenance, Novell, Inc., March 1991, Novell Part Number 100-000914-001