Basic Configuration

451-0084C

Contents

Figures

Overview

The Access Server software supplied by Xyplex Networks operates on Xyplex-supplied communication hardware modules, which are part of the MAXserver family and Network 9000 family of Ethernet-based communication products. This combination of software and hardware is called an access server.

Access servers support connections between serial-interface devices and other devices connected to the Ethernet network. The serial-interface devices include: terminals, serial printers, personal computers running terminal emulation or networking software, modems, serial ports on other access servers, and host computer serial ports. Figure 1 represents an access server configuration.

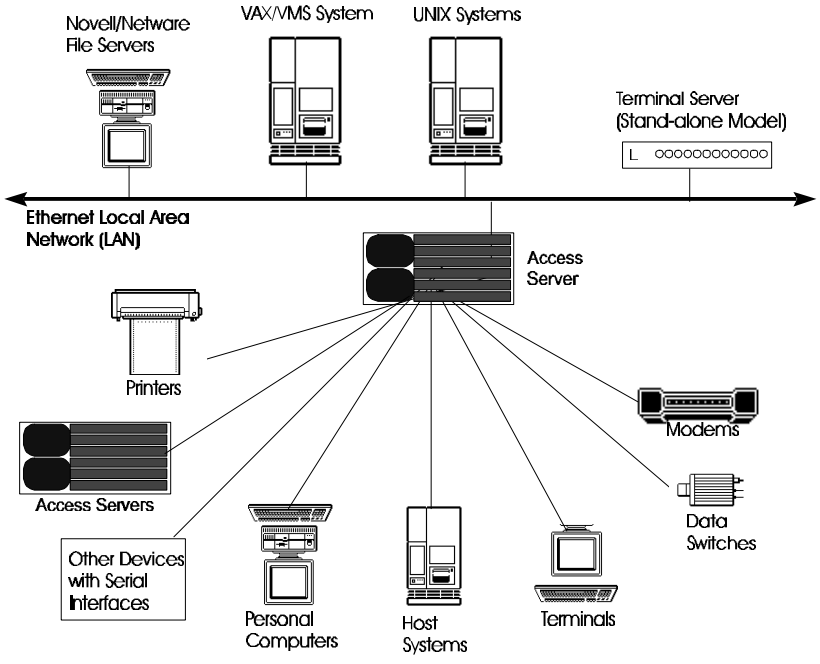


Figure 1. Network Configuration with Access Servers

Figure 1 shows how several different types of devices can be connected to the serial ports of the access server. Users at any of these devices have access to any resources on the network, such as host computers, workstations, etc, and resources available at other access server ports. Because access server software supports multiple communication protocols, these connections can be made regardless of the operating system running at the desired resource.

Compare this to operation without an access server, where users would need to go to a terminal that is directly connected to a serial port on the host or node they want to use. With an access server, users can perform work on any computer that is connected to the network that the access server is on. Providing access from serial ports to host computer resources is referred to as "terminal serving."

Similarly, consider the example of a user who wants to print a job. Without an access server, the user would need to print the job to a printer that is directly connected to a serial port on the host where the data is located. With an access server, many users can have access to a shared printer resource, because the resource is located on the network. Providing shared printing resources is also referred to as "printer serving."

For terminal serving and printer serving, the devices which provide services are connected to a network and the users of those services usually work at a location that is geographically local to the device offering the service. Serial ports at access servers can also be connected to modems, switches and other devices to provide access to services that are available at remote locations or for users who are at remote locations. This is referred to as "access serving."

Access serving configurations include anything from simple dial-in and dial-out modems for low speed interactive traffic (terminal emulation, text editing, file transfers, electronic mail), to more sophisticated applications.

IP/PPP Protocols

The Point-to-Point Protocol (PPP) allows a personal computer (PC), another access server, or router that also supports PPP to gain access to a network, such as Internet networks (IP) or Novell NetWare networks (IPX), through a serial port. PPP devices can connect to the access server directly over a serial line or through a modem.

PPP provides a standard method for transmitting multi-protocol datagrams over point-to-point links. Because PPP is a datagram transmission service, it is not a guaranteed delivery service. (To compensate, flow control methods and the requirement by higher-level protocols that messages be acknowledged before additional messages are sent means that most packets are delivered without error.)

PPP provides an excellent foundation for other applications. PPP defines a Link Control Protocol (LCP) for establishing, configuring, and testing the data-link connections. PPP also provides a family of Network Control Protocols (NCP) for establishing and configuring network layer protocols. The IP Control Protocols (IPCP) and IPX Control Protocol (IPXCP) are NCPs supported by the Xyplex PPP implementation.

IP/PPP (IPCP) Features

The Xyplex implementation of IPCP supports two general network configurations: the single-node configuration and the network configuration. In the single-node configuration, a PC running PPP is attached to an access server port over a serial line. In the network configuration, two access servers route IP traffic between two LANs with different subnet addresses in the Internet.

Figure 2 and 3 show examples of the two network topologies.

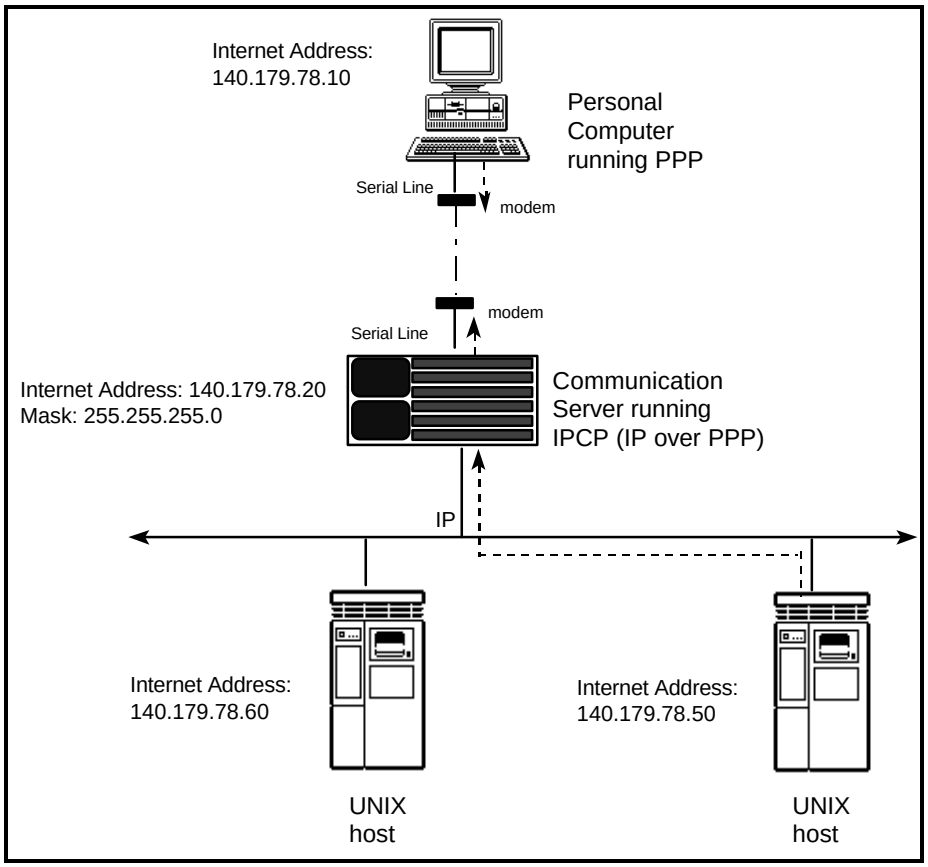


Figure 2. IPCP Single-Node Configuration

The PC in Figure 2 can have an address in the same Internet subnetwork, or subnet, as the access server, or on a remote subnet with a different Internet address. The PC can connect to the access server port directly or through a modem, as shown in this figure. When the PC and the access server are configured appropriately, the PC can gain access to Internet devices on the LAN through the access server port.

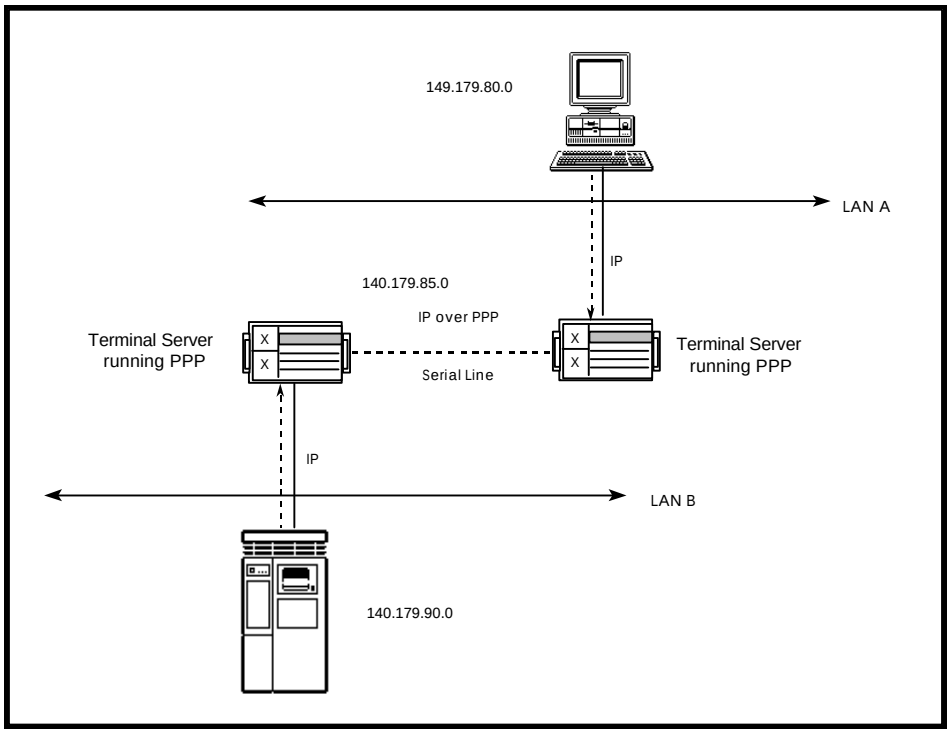


Figure 3. An IPCP Network Configuration

The two LANs in Figure 3 represent different subnets on the Internet. Two access servers running IPCP, connected over a serial line, act as a gateway or router between the two networks. Devices on LAN A can gain access to devices on LAN B, and devices on LAN B can gain access to devices on LAN A.

PPP Features

The following features are also part of the PPP IPCP implementation:

- Support for high speed modems.
- Depending on the access server model and cabling you use, PPP links can be configured to operate at speeds as high as 115.2 Kbps.
- Support for Van Jacobson compression.

  PPP links can transmit and receive packets that have been compressed using the Van Jacobson compression algorithm (refer to RFC 1144). Compression allows PPP links to operate with higher throughput (actual performance depends on your application).
- Compatible with Xyplex security mechanisms.

IPCP can be used in conjunction with all Xyplex access security methods.

IPX /IPXCP Protocols

In Novell NetWare networks, communication is handled using a protocol known as Internetwork Packet Exchange (IPX). IPX is a connectionless, datagram protocol, which means that each packet contains all the information necessary to deliver it to the final destination. The PPP specification defines many Network Control Protocols (NCP) for establishing various network layer protocols. IPX Control Protocol (IPXCP) is one such network control protocol, and specifies a means for handling IPX traffic running over a PPP link.

With Multiprotocol software, an access server provides transparent access to IPX services, devices or networks. Specific applications include:

Connecting a remote (dial-in) Workstation or PC to the network. This is a typical "remote office" or "user-to-LAN" application. Figure 4 depicts this IPXCP configuration.

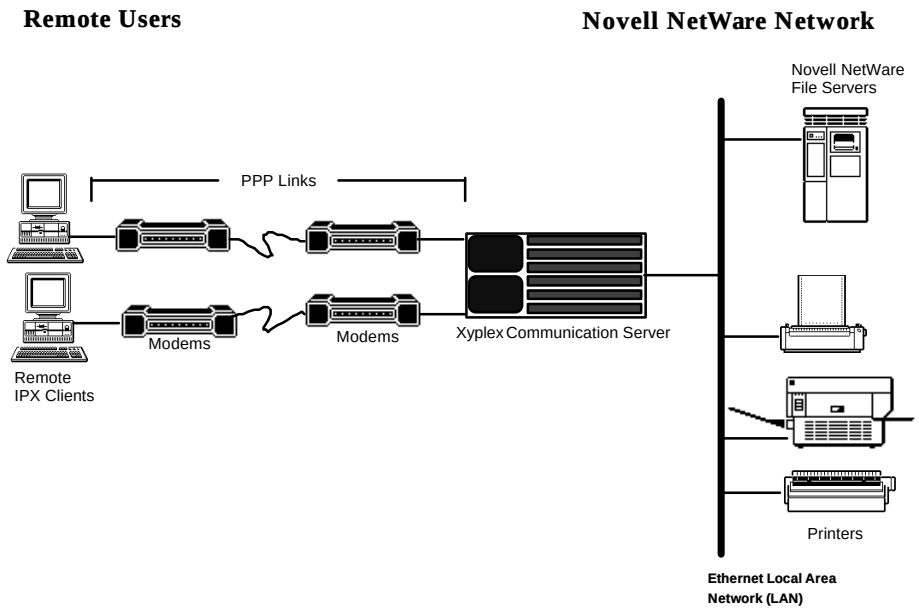


Figure 4. Basic IPXCP Configuration Using a Communication Server

In this application, the users at the remote IPX clients have access to the Novell services offered on the Novell Netware network (unless the network manager chooses to limit that access). The user dials in when he or she needs access to the services, and disconnects when the services are no longer needed.

Connecting a remote network to the local network through a serial port. In this application, the port functions as a router connecting two networks. Figure 5 depicts this IPXCP configuration.

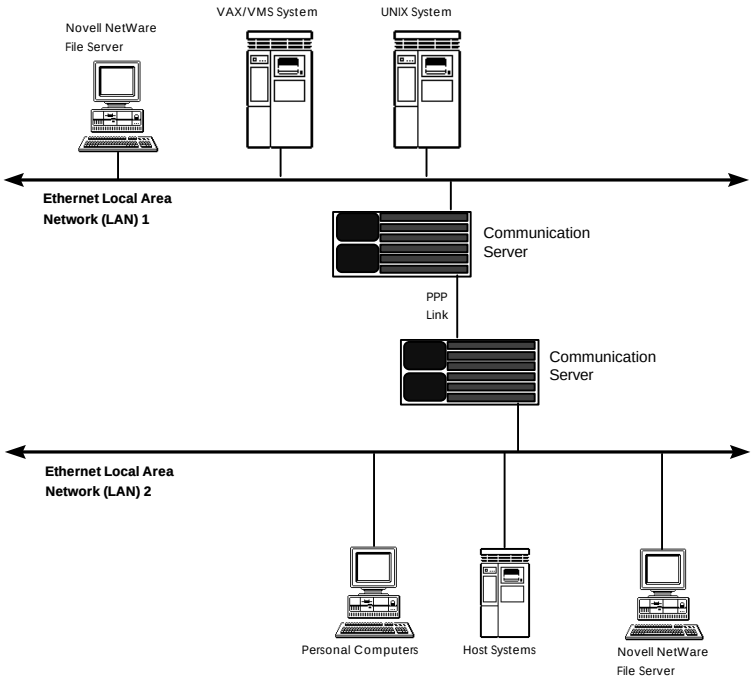


Figure 5. IPXCP "LAN-to-LAN" Configuration Using Communication Servers

- In this application, the users at one Ethernet LAN have access to all Novell services offered at the other Ethernet LAN, and vice versa. In Figure 2, the PPP link between the access servers would typically be a "null-modem" connection. The connection will normally be a permanent link. An asynchrouous IPX router which has dial-out capabilities could also be used in place of one of the access servers.

- Connecting a host, workstation, or PC directly to the network through a connection to a serial port (usually uses a "null-modem" cable). This application would be useful for connecting a device which does not support an Ethernet connection to the access server so that it has access to services available on the Novell network. The PC or workstation needs an IPX client program and the ability to communicate over a PPP link.

Key Features

- The access server can communicate with any RFC 1552-compliant IPXCP (IPX over PPP) client software implementation.

- Standard NetWare Addressing Methods. The IPX protocol specifies the address of each system using a network number, node number, and socket number. Network numbers identify NetWare network segments. Node numbers identify individual nodes on a network segment. Socket numbers identify the different applications within a single host. The access server software uses this standard addressing method. For information about IPX addressing, refer to the *Novell System Concepts* guide supplied with your Novell NetWare software.

- Use the IPX client setup and administration procedures to configure the remote IPX client. The IPX client set-up activities are described in the documentation supplied with your IPX client software package. Use Xyplex commands to configure the access server.

- IPX RIP and SAP Support. In some network configurations, an access server operates as an asynchronous IPX router. IPX routers exchange information about the networks where they are attached, and the networks they can reach, through IPX Router Information Protocol (RIP) packets. IPX routers use RIP information to route IPX packets. Each IPX router maintains a table of RIP information that it has received from other routers. IPX routers also broadcast RIP packets to neighboring routers periodically.

- Servers in an IPX network (e.g., file servers, print servers) advertise their services through Service Advertising Protocol (SAP) packets. IPX servers also answer requests by clients who are looking for their services. IPX routers are responsible for broadcasting SAP information to other IPX routers in the network, and functioning as a proxy for servers on other networks[1]. Each IPX router maintains a table of SAP information that it has received from neighboring routers and servers.
- RIP and SAP route-propagation is performed using a "split horizon algorithm." [2]
- IPX can be used in conjunction with all Xyplex access security methods.
- The IPXCP implementation can be managed via SNMP and includes support of Xyplex enterprise-specific MIB objects, and Control Point.

Using SLIP

The Access Server software enables a user to run Internet protocols over an asynchronous serial line, using the Serial Line Internet Protocol (SLIP). SLIP is specified by the Internet RFC 1055.

---

1 When a Netware client wants to connect to a service, the client broadcasts a request for the service. All IPX routers that have the service in their SAP tables respond to the request, based on the split horizon algorithm.

2 This algorithm indicates that when broadcasting RIP routes or SAP announcements to a given network, an individual router should only include data that the other network is not likely to know. For example, a router must not advertise a route to a network that it had learned from that network.

The Access Server software supports two models for the utilization of SLIP: the single-node model and the network model. Single-node SLIP applications include:

- Connecting a remote (dial-in) Workstation or PC to the network. This is a typical single node application.
- Connecting a host, workstation, or PC directly to the network through a connection to a serial port. (This single-node application usually requires "null-modem" cables.)

In the single-node model, a node is an intelligent device such as a PC, workstation, etc. Each node connected to an access server has a unique Internet address.

Using the network model, one connects a remote network to the local network through a serial port. In this application, the port functions as a simple static router connecting two networks.

In the network model, a network is a collection of internet nodes, each with a different internet address. Data communicated over the SLIP link is forwarded to a remote network. As shown in Figure 6, an access server passes data from the SLIP link to another local area network.

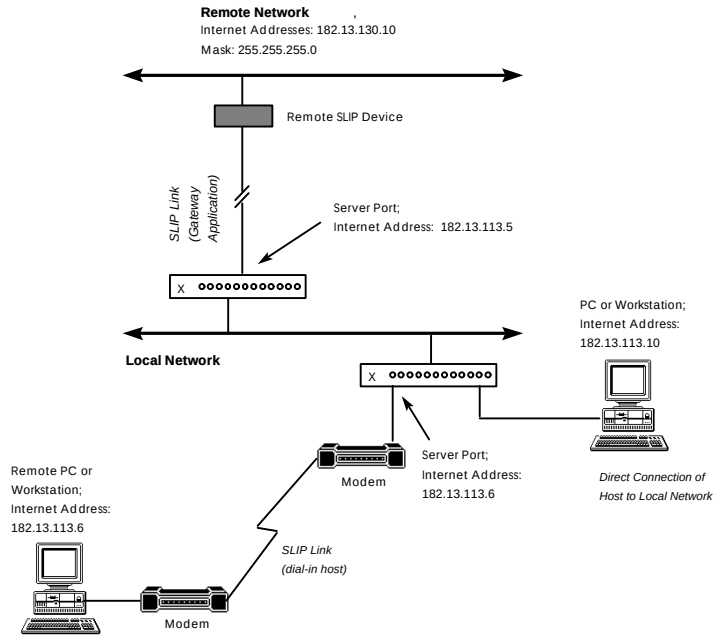


Figure 6. SLIP Connections to Remote Network, Remote PC

As shown, SLIP connections can be made from ports to a remote (dial-in) workstation or PC, direct connection of a host to the local network, and connection of a remote network.

SLIP Features

The following are some important features of the SLIP implementation:

- The server supports line speeds of 50 baud to 115,200 baud. Depending on the access server model and cabling you use, SLIP links can be configured to operate at speeds as high as 115.2 Kbps.

- SLIP links can transmit and receive packets that have been compressed using the Van Jacobson compression algorithm (refer to RFC 1144). Links using Compressed SLIP are referred to as CSLIP links. Compression allows SLIP links to operate with higher throughput (actual performance depends on your application). SLIP links can also transmit and receive uncompressed packets, since not all remote devices permit the use of compression.

- When a remote device initiates activity on the link, the port will automatically detect whether or not the remote device is using compressed SLIP packets. The port will use the same type (compressed or uncompressed) of packets as the remote device.

- When the port initiates activity on the SLIP link, you must specify whether or not the port can initiate communications with a remote device using CSLIP packets (using the DEFINE/SET PORT INTERNET CSLIP ENABLED/DISABLED command). When the use of compressed SLIP is enabled, the port will immediately begin transmitting compressed packets on the serial link.

- SLIP can be used in conjunction with all Xyplex security methods.

XREMOTE

The access server provides serial-line support for the NCD proprietary Xremote protocol. The Xremote protocol compresses the MIT X Windows™ protocol across a serial line. The Xyplex support for this protocol enables you to connect NCD Xterminals to a Xyplex access server, either directly or with a modem. In this configuration, Xterminal users have access to many resources on the LAN that may have previously been unavailable to them. In addition, the access server runs Xremote helper code, which normally runs on the host. Because of this, the host has more resources available to run applications.

In a conventional configuration, you either connect Xterminals to a host computer running the MIT X Windows protocol at the serial port on the host, or you connect the Xterminal directly to the LAN. When the Xterminal is connected to the access server, the serial port on the host computer is free for other uses.

Xremote Features

- Operates with NCD Xterminals having revision V2.2 and V2.3 Xremote server code in PROMs. An NCD Xterminal connected to a Xyplex access server with Xremote support is equivalent to the same terminal connected to a host running NCD Xremote helper code.
- Operates with Massachusetts Institute of Technology (MIT) X11R4 and X11R5 X Windows programs.
- Provides font loading from hosts using the Trivial File Transfer Protocol (`tftp`).
- Supports Xremote operation at line speeds of 9600 baud or greater.
- Permits nondedicated Xremote ports. An interactive user can choose one of several different types of connections including Xremote, SLIP, Multisessions, TN3270, or normal interactive capabilities on a serial port.
- Supports Xwindow Display Manager Control Protocol (XDMCP) notification of X Display Manager (XDM) hosts.

For more information about X Windows system, and how to install the XDM manager in particular, refer to *X Window System User's Guide Volume Three*, by Valerie Quercia and Tim O'Reilly, O'Reilly and Associates, Inc.

For general information about the Xremote protocol, refer to the *NCDware 2.3 Xremote User's Manual*, from Network Computer Devices, part number 9300137.

Figure 7 shows a conventional Xremote configuration with the host computer running Xremote helper code and the X Windows program.

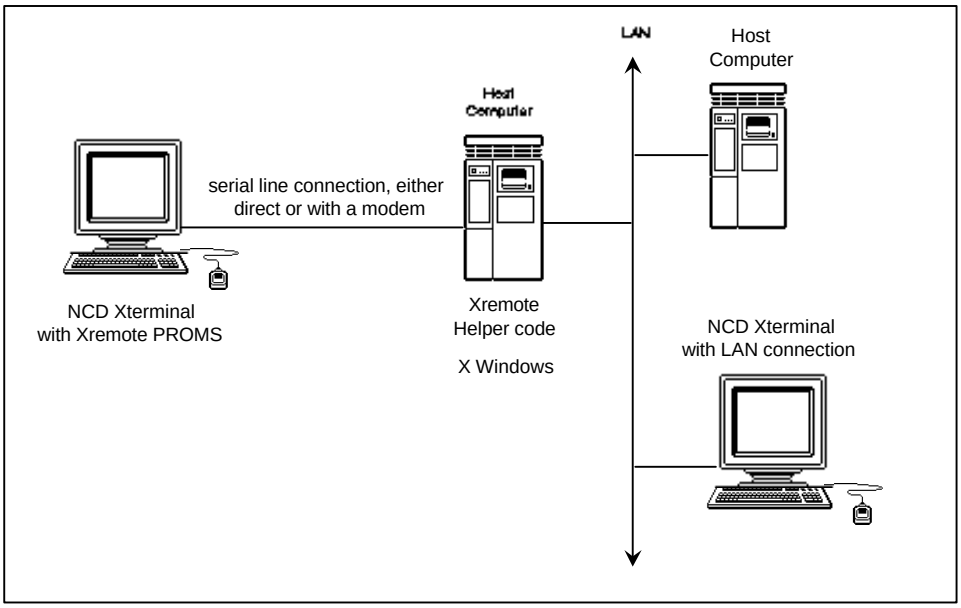


Figure 7. Conventional Xremote Implementation

In Figure 7, the Xterminal connected to the host must establish the initial Xsession with that host.

Figure 8 shows an example of an Xterminal connected to an access server.

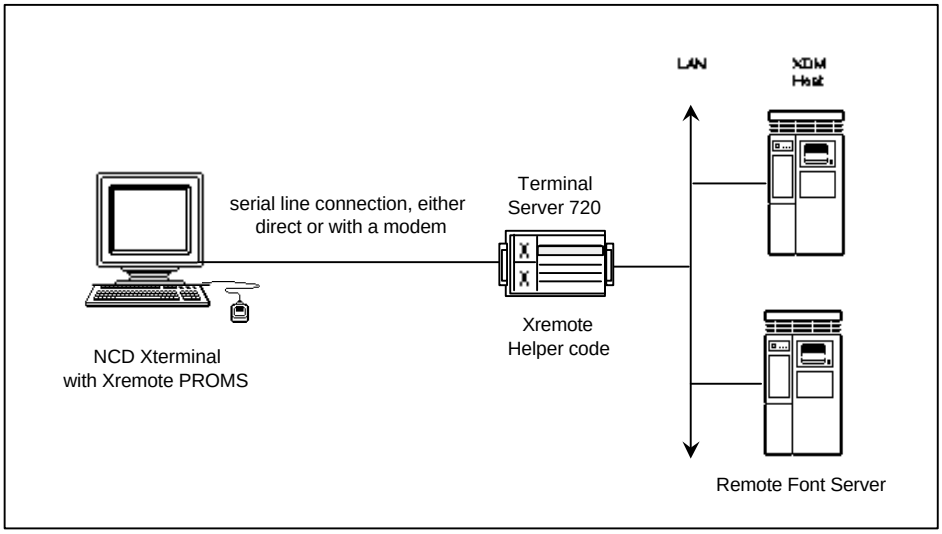


Figure 8. An Xterminal Connected to a Xyplex Access Server 720

The NCD Xterminal in Figure 8, which contains the Xremote protocol stack in PROMs, is connected to the access server with a serial line. The access server runs the Xremote helper code. A user at this Xterminal can gain access to different resources on the LAN, either directly or with a modem. The Xterminal can establish the initial Xsession with various hosts on the network, which run the XDM. The XDM host resides on the LAN, as does another host which acts as a remote font server.

In this configuration, the Xremote protocol runs across the serial line. The X Windows protocol runs on the LAN. The NCD helper code compresses X Window traffic from the LAN and sends it across the serial line. It also decompresses Xremote traffic from the serial line, and sends X Windows protocol across the LAN.

ARAP

AppleTalk Remote Access (ARAP) allows a user to connect a remote Macintosh computer to an AppleTalk network through a point-to-point modem link. A Remote Access server transfers AppleTalk packets between a remote Macintosh and an AppleTalk network so that the remote Macintosh acts as if it were directly connected to the network.

ARAP is a "keyed" software feature and requires a password to be enabled at a Xyplex access server. Contact your local Xyplex Sales Representative or distributor for more information about obtaining a password and the documentation which describes how to configure this feature on the Xyplex access server.

Notes

The following notes apply to the ARAP implementation:

- When there is no TFTP script server available on the network, Command Control Language (CCL) scripts and dial back scripts are unavailable.
- ARAP supports only one login password that is shared by all ARAP users. When Kerberos or SecurID authentication is performed, a username may be used that has an associated password and/or passcode.
- When Kerberos or SecurID authentication is not used, the server does not restrict access by user name. A user can login through Remote Access using any user name as long as the user specifies the correct server password. Specific user names are only used for locating a telephone number for dial back.
- To prevent AppleTalk “name collisions,” do not have more than one Remote Access Server with a given name on an AppleTalk network.

Figure 9 and Figure 10 show the differences between the standard Remote Access configuration, and an equivalent configuration using a Xyplex communications server. In the standard configuration (see Figure 9), a Macintosh computer is dedicated for use as a Remote Access server. The remote Macintosh computer has access to all AppleTalk zones that are available to the Macintosh computer that is configured as a Remote Access server. Each Macintosh computer (Remote Access server) can only service one modem, so multiple Macintosh computers are needed in order to support simultaneous connections to the network by several users.[1]

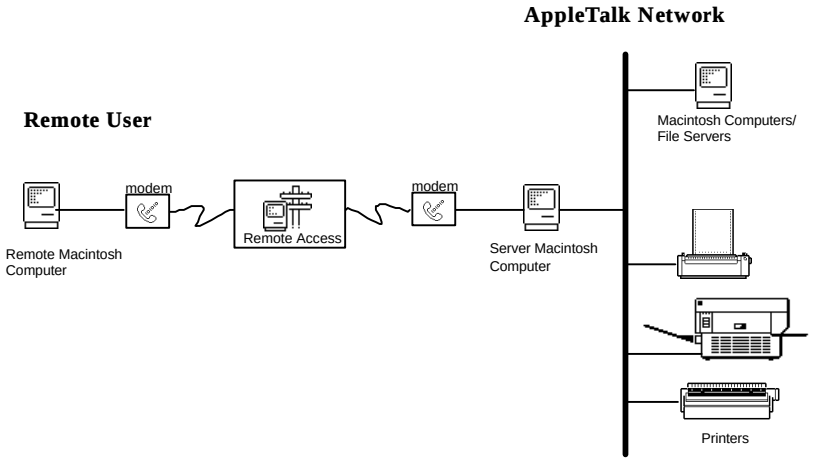


Figure 9. Standard AppleTalk Remote Access Configuration

1 This description applies to Version 1.X of AppleTalk Remote Access from Apple Computer Corporation. Version 2.0 of AppleTalk Remote Access permits a Macintosh computer to service more than one Remote Access connection when the Macintosh computer is equipped with a special hardware option. (Without this hardware option, V2.0 performs in the same manner as V1.X.) The Xyplex ARAP implementation is compatible with both V1.X and V2.0 of AppleTalk Remote Access software.

In the Xyplex configuration (see Figure 10), a port on the access server takes over the functions of the Macintosh computer (Remote Access server) and provides access to AppleTalk devices and networks. Using this approach, several users can simultaneously and cost-effectively be connected to the network via one multi-port access server device. Also, network administration is simplified, because an administrator only needs to manage a single access server, rather than multiple Macintosh computers (Remote Access servers).

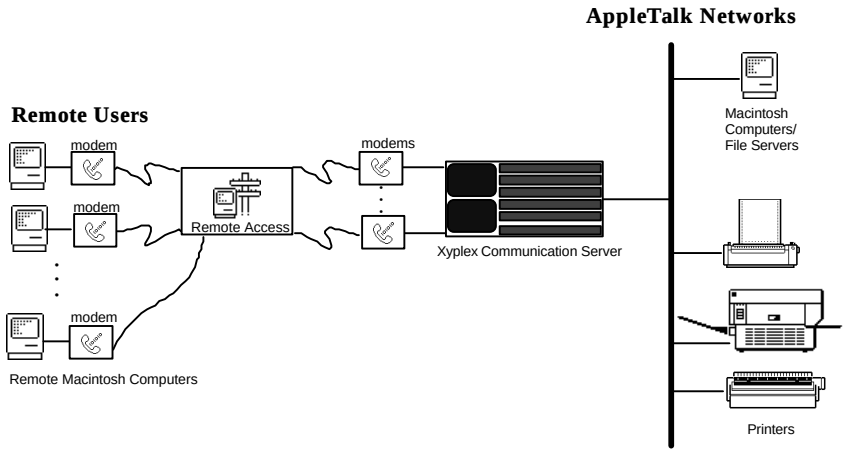


Figure 10. AppleTalk Remote Access Configuration Using Communication Servers

ARAP Features

Setup and administration using native procedures. You use the standard Macintosh setup and administration procedures to configure the remote Macintosh. The Macintosh set-up activities are described in the documentation supplied with your AppleTalk Remote Access software package, from Apple Computer, Inc. You use Xyplex commands to configure the access server. The Xyplex ARAP-related commands are described in the *Commands Reference Guide.*

NOTE: ARAP operates on access servers that support the Xyplex eight-wire cabling method. You must also use cables that are appropriate for

using the CTS/RTS flow control method. You can obtain modular cables and adaptors from Xyplex which provide the appropriate signals. The *Getting Started Guide* contains more detailed information about cabling..

Modem Support. ARAP can operate with any modem that supports a Hayes-compatible command interface. (Modem configuration for ARAP is performed by CCL scripts. CCL scripts are covered later in this section.)

SNMP Manageability. ARAP can be managed via SNMP and includes support of Xyplex enterprise-specific MIB objects, and Network Management Software.

Remote Access user dialback. ARAP supports user dialback. This feature provides a way of ensuring that only authorized users can connect to the network (a complete discussion of the available security methods is contained in “Setting UP ARAP””). When the remote user first connects to the port, the user logs in using a login name assigned by the network administrator. The port immediately disconnects and requests the dialback script for that user's name be downloaded from a script server. The dialback script contains the telephone number for the modem to dial. The modem then dials that number and attempts to establish a connection with that user.

IP/IPX Routing

Large Internet (IP) or Novell NetWare (IPX) networks with many hosts, file servers, or other devices which offer user services are often subdivided into smaller, separate networks to improve overall network performance and make the network easier to manage. These subnetworks, or subnets, can exist in the local or distributed locations. Sites with a small number of devices that are connected through routers or gateways to larger IP or IPX networks can also be divided into subnets.

The access server contains a list of routes. This list is called a routing table. The routes specify a preferred path where the access server can send traffic bound for a particular destination.

The Access Server Software supports some IP and IPX routing capabilities, such as:

- Limited IP routing. Servers collect ICMP (Internet Control Message Protocol) messages, which allow the server to "learn" IP routes in order to send packets to the appropriate destination. Servers can also be configured with a manager-specified, or "static" IP route.

- Full IPX routers. Servers collect IPX RIP (Routing Information Protocol) and SAP (Service Advertising Protocol) messages, which allow the server to "learn" IPX routes in order to send packets to the appropriate destination. Servers also broadcast IPX RIP and SAP messages and can be configured with a manager-specified, or "static" IPX route.

- The IP and IPX protocols support communication between devices on separate subnetworks through routers or gateways which are connected to two or more of the subnetworks. The routers and gateways communicate among themselves, forwarding network traffic to each other as well as between networks, according to their routing protocol. The gateways also select the most efficient path to a destination for communications sent by a host or access server. This is called "*dynamic routing*."

As conditions change, the path to a destination may also change. For example, when the Internet route changes, the hosts and access server are informed, through ICMP routing messages, that the packets they sent to a particular gateway have been forwarded to another gateway on the same network. As long as a host or access server sends traffic to a gateway that forwards traffic when necessary, the sender can be assured that the packets will eventually reach the destination. Internet routes to a destination which the server obtains in this manner are called learned Internet routes. Similar activities occur on IPX networks, where IPX RIP and SAP messages inform the access server of changes.

For some networks, dynamic routing may not be enabled or desirable. In this case, the sending host or access server must select the correct path to ensure that packets are forwarded to the right destination. To do this, the manager must specify database entries which correlate specific destination networks or hosts to the specific gateways that must be used to reach those destinations. This is called "static routing." There are access server commands which allow an administrator to specify static IP and IPX routes.

IP/IPX Filters

An access server has a single network interface (Ethernet connection) and multiple, separate, asynchronous interfaces (serial ports). Each interface can be configured with IP and IPX packet filters. These filters are used to allow certain IP or IPX traffic to pass through the server. The filters can specify network destination or source address, protocol, packet type, as well as other filter-specific criteria. Filtering is disabled by default.

The software can be configured to filter IP traffic, based on the following criteria:

- Source or destination address or network (internet-address and/or subnet).
- IP packet type (port number, protocol, setting of TCP SYN bit).

The software can be configured to filter IPX traffic, based on the following criteria:

- Source or destination address or network (IPX network number and node number).
- IPX packet type

The Access Server software also permits you to configure the software to limit broadcasting, learning, and use of IPX RIP route or SAP service information. You can create import and export filters for these packets.

Import filters enable you to control the information that a server adds to its IPX RIP or SAP Table. The server either accepts or discards routes and services that meet the criteria, based on the filters instructions.

Export filters enable you to control the routing or service information that a router sends to the network. The router either advertises or hides routes or services that meet the criteria, based on the filters instructions.

“Point-to-Point Protocol” covers filtering in more detail.

CCL Scripts

Command Control Language (CCL) scripts are files that contain commands which initialize a modem and configure communication between the modem and the device to which it is connected. The CCL script in an ASCII file. Since there are many types of modems, the CCL script "language" is flexible enough to accommodate them all.

CCLs were originally designed to be used with AppleTalk Remote Access. For ARAP connections, CCL scripts provide modem initialization commands to both modems that are part of a given AppleTalk Remote Access connection. (That is to say: the remote Macintosh computer provides initialization commands to its modem and the Remote Access server does the same for its modem.)

For Xyplex access servers, CCL scripts can be used to initialize ports for all types of modem connections and protocols. One could think of a CCL as an "alternate" method of autobauding, since the CCL will determine the appropriate port speed and set it accordingly. CCL scripts are required for ARAP connections, but can be used to initialize the port and modem for other types of connections (PPP, SLIP, interactive, etc).

CCL scripts are stored on script servers (hosts which can transfer files to the communications server via TFTP). Individual ports are configured to use a specific CCL script. The access server downloads the CCL script once, then executes the commands in the script when the access server is first initialized and when a connection is disconnected. That way, the modem is ready to accept the next incoming connection.

Xyplex supplies CCL scripts for use with a variety of modems that can be connected to access server ports and to remote Macintosh computers. These are listed in the *Software Kit Information* supplied with your software kit. CCL scripts for use with remote Macintosh computers can also be obtained from the manufacturer of the modem, or from public domain sources.

"Using CCL Scripts" describes the use of CCL scripts in more detail.

Protocols and Features

The Xyplex Multiprotocol Communication Server software offers many protocols and features. Which ones you use depend on the type of network you have and the amount of memory in the access server. Most sites do not require all possible features and protocols. In general, if a protocol is not needed, you should disable it to make more memory available for other uses.

Table 1 lists the Multiprotocol Communication Server access serving software protocols that you can enable or disable, as well as the amount of memory that will be used or freed up. Table 1 only lists the access serving protocols that one can enable or disable. There are access serving features (such as interactive connections, SLIP, or CSLIP) that do not need to be enabled. Also, there are protocols and features unrelated to access serving that can be enabled or disabled. A complete list of these can be found in Chapter 2 of the *Software Management Guide.* In some configurations, one might need to disable some protocols or features in order to make more memory available to enable an access serving protocol.

Table 1. Memory Usage For Features and Protocols

Protocol/Feature Name	Memory Used in Kilobytes	Type	Default	Comments
XREMOTE	22	Protocol	Disabled	This feature requires more memory for each open session. Requires Multi-Meg load image. Password required.
PPP	30, plus 5 packet buffers per port	Protocol	Disabled	IPCP is enabled when PPP is enabled. Requires Multi-Meg load image.
ARAP	180, plus 43 kilobytes per port	Protocol	Disabled	Requires Multi-Meg load image. Password required.
IPX	80, plus 5 packet buffers per port	Protocol	Disabled	PPP must also be enabled. IPXCP is enabled when both PPP and IPX are enabled. Requires Multi-Meg load image. Password required.
APD	5	Feature	Disabled	Requires Multi-Meg load image.

To enable or disable a protocol use the command:

`DEFINE SERVER PROTOCOL` *`protocol-name`* `ENABLED/DISABLED`

where *protocol-name* represents the name of a protocol listed in the first column of Table 1. For example, you would use the following command to enable PPP:

```
Xyplex>> define server protocol ppp enabled
```

NOTE: Many protocols require a password in order to be enabled. Table 1 lists the protocols which require passwords.

When you use one of these commands to enable or disable a protocol, the software will display a message similar to the following message, to indicate approximately how much memory remains available:

`-705- Change leaves approximately` *`nnnnn`* `bytes free.`

It is strongly recommended that you leave a minimum of 180 kilobytes of memory after all desired features have been enabled. If the memory needed for the desired features exceeds the amount of memory available on the unit, the server will display a message similar to the following message, to indicate approximately how much memory you need to free up in order to enable the feature:

`-708- Requires approximately` *nnnnn* `additional bytes; Change not done.`

Initialize the server after you have made all changes. When a protocol is enabled, the software sets all server or port characteristics associated with that protocol, meaning those characteristics set with DEFINE/SET SERVER and DEFINE/SET PORT commands, to their default values. When a protocol is disabled, the software changes all server or port characteristics associated with that protocol to reflect this.

Automatic Protocol Detection (APD)

Access server ports can be configured to accept connections made via different protocols, using the Automatic Protocol Detection Feature (APD).

APD Notes

To use APD, the access server port must be configured with PORT ACCESS set to LOCAL or DYNAMIC (applies only to dial-in connections).

To enable APD, and have the APD prompt display on a specific port, use the following command:

```
DEFINE PORT APD PROMPT ENABLED|DISABLED
```

The default prompt is “”.

Using APD, ports will automatically determine the protocol being used to make a connection and adjust port settings appropriately. If you do not enable APD, ports can be dedicated for use by a single protocol. Key features of APD include:

- An individual port can be configured to accept any connections made via ARAP, PPP (which includes IPCP and IPXCP), SLIP (which includes CSLIP), and interactive protocols, as well as all, none, or any combination of these.
- Ports can be configured to limit the amount of time spent in an attempt to determine which protocol is being used to make a connection. When the time expires, then the port will either default to a specific protocol, or logout the connection, as specified by the server manager.
- All access server security features (e.g., SecurID, Kerberos, etc) apply to ports configured with APD enabled.

NOTE: Do not use script logins on APD ports. The access server only executes login scripts for Interactive ports.

APD Setup

To configure server ports to accept different types of connections (i.e., using more than one protocol), issue the following command:

```
XYPLEX>> DEFINE SERVER APD ENABLED
```

Initialize the server. After you enable APD on the server, you must enable APD-related settings on individual ports. If you do not specify APD-related characteristics for the ports which use access serving protocols, the ports will default to permitting only interactive connections, unless configured with another protocol.

```
DEFINE SERVER APD MESSAGE [“message-string”]
```

For APD to work, you must first disable autobauding. Use the command:

```
DEFINE PORT port-list  AUTOBAUD DISABLED
```

For example:

```
Xyplex>> define port 6-12 autobaud disabled
```

Next, since autobauding is disabled, you must specify a port speed or use a CCL script to set the port speed when a call is made. “Using CCL Scripts” covers the procedure to configure a port to use a CCL script. To specify a port speed, use the command:

```
DEFINE PORT port-list  SPEED port-speed
```

For example:

```
Xyplex>> define port 6-12 speed 14400
```

APD

The following commands specify how APD will operate at a port:

```
DEFINE PORT port-list APD     [ALL]
                              [ARAP]
                              [DISABLED]
                              [INTERACTIVE]
                              [NONE]
                              [PPP]
                              [SLIP]
```

This command specifies the types of connections that will be allowed at the port. (This prevents non-enabled connection types.) The protocol-list can include: ALL, ARAP,DISABLED, NONE, PPP, SLIP, and INTERACTIVE. The default is DISABLED, which is the same as NONE. ALL permits any type of connection to be established at the port(s), while the remaining values (ARAP, PPP, SLIP, and INTERACTIVE) limit the port(s) only to connections of the types listed. For example, to permit PPP and ARAP connections, use the command:

```
Xyplex>> define port 6-12 apd arap,ppp
```

Note that ARAP and PPP must previously have been enabled for the server. It is not necessary that they be enabled at the port.

```
DEFINE PORT port-list  APD TIMEOUT time
```

This command specifies how much time the port can spend in an attempt to determine which protocol is being used to make a connection. Possible time values are numbers in the range 1 to 255 (seconds) or UNLIMITED, which means that the port can continue indefinitely. If a number between 1 to 255 is specified, then whenever the port is unable to determine the protocol within the specified time, the port will either default to a specific protocol, or logout the connection, depending on the setting of the DEFINE PORT APD DEFAULT command. For example, to permit the port to spend up to 30 seconds in an attempt to determine which protocol is being used to make a connection, use the command:

```
Xyplex>> define port 6-12 apd timeout 30
```

```
DEFINE PORT port-list  APD DEFAULT [LOGOUT]
                                   [ARAP]
                                   [PPP]
                                   [SLIP]
                                   [INTERACTIVE]
```

This command specifies the action that the port(s) will take in the event that the ports are unable to determine which protocol is being used to make a connection. The protocol can be: LOGOUT, ARAP, PPP, SLIP, or INTERACTIVE. The default is LOGOUT, which means that the port will be logged off if APD is unable to determine which protocol is being used to make the connection. The remaining values (ARAP, PPP, SLIP, and INTERACTIVE) indicate which protocol the port should assume is being used for the connection. The protocol specified for this command must be included in the list of possible protocols in the DEFINE PORT APD command. For example, to specify that the port should assume that a connection is a PPP connection after the APD TIMEOUT period has expired, use the command:

```
Xyplex>> define port 6-12 apd default ppp
```

After you have configured a port to accept multiple types of connections, you must specify the appropriate PORT characteristics for the permitted protocol(s).

NOTES: When using APD at a port, you do not need to enable specific protocols, such as PPP, IPX, ARAP, or SLIP/CSLIP at that port (only at the server for PPP, IPX and ARAP). APD will enable the protocol at the port when a connection is made. However, you must configure all appropriate PORT and/or SERVER characteristics (addresses, etc) that apply to that protocol.

When using APD at a port, you do not need to disable modem control in order to support ARAP connections. APD will automatically disable modem control when it detects that an ARAP connection is being made.

Authentication

If the APD feature has been enabled on a port, use this command to determine when user authentication is implemented: either before or after APD determines the user protocol being used (such as INTERACTIVE, PPP, SLIP). APD authentication is required in addition to protocol-level authentication mechanisms. If authentication will be done after protocol detection, PPP or SLIP users must use a protocol-level authentication such as PAP or CHAP.

```
DEFINE/SET PORT <port-list> APD AUTHENTICATION INTERACTIVE ONLY
                                                          [ENABLED]
                                                          [DISABLED]
```

APD PROMPT

Use this command to define whether or not the APD prompt will be displayed on a specific port.

Syntax

```
DEFINE PORT <port-list> APD PROMPT [ENABLED]
                                   [DISABLED]
```

Where	Means
ENABLED	The APD prompt will be displayed on the specified port(s). The default prompt is “AUTOMATIC PROTOCOL DETECTION - Begin Protocol or enter 4 returns for interactive mode.”.
DISABLED	No prompt will be displayed.

```
Example     SET PORT 20 APD PROMPT ENABLED
```

IP Address and Subnet Mask

This section describes how to assign an IP address and optional subnet mask, and to configure domain name server support.

To set up a server to operate as an Internet node, you need to assign it an IP address and subnet mask. When the Subnet Mask Autoconfigure setting is enabled — as it is by default — the server assigns a subnet mask automatically when you define the IP address. In this case, the subnet mask is determined by the class of the IP address (A, B, or C).

For example, the following command assigns the Class B address 172.19.1.1:

```
Xyplex>> define server internet address 172.19.1.1
```

The default subnet mask for a Class B address is 255.255.0.0; the server automatically assigns this mask. If you want use a different subnet mask, you must disable the Autoconfigure feature. Use these commands to define a subnet mask and enable/disable the autoconfigure setting:

```
DEFINE/SET SERVER INTERNET SUBNET MASK AUTOCONFIGURE [ENABLED]
                                                     [DISABLED]

DEFINE/SET SERVER INTERNET SUBNET MASK subnet-mask
```

Domain Name Server Support

For the server to operate with a domain name server (a network device that maps domain names to IP addresses), you need to define these settings:

- Internet Name
- Internet Domain Address(es)
- Internet Default Domain Suffix

The Server Internet Domain Address specifies the domain name server's IP address. You can define up to two domain name servers: a primary and a secondary.

The commands in the following example assign MAX5000.XYPLEX.COM as the domain name for an access server, and XYPLEX.COM as the default domain name suffix. This example assigns primary and a secondary domain name servers, which are located at the addresses 172.19.1.200 and 172.19.1.250.

```
Xyplex>> define server internet name max5000.xyplex.com
Xyplex>> set server internet name max5000.xyplex.com
Xyplex>> define server internet default domain suffix
         .xyplex.com
Xyplex>> set server internet default domain suffix
         .xyplex.com
Xyplex>> define server internet primary domain address
        128.3.0.200
Xyplex>> set server IP primary domain address
        128.3.0.200
Xyplex>> define server IP secondary domain address
        128.3.0.250
Xyplex>> set server IP secondary domain address
        128.3.0.250
```

You should also add the server's domain name and IP address to the database files at the domain name servers (Berkeley Internet Name Domain Server or NIS) for your network. In this way, requests for the server's domain name can always be resolved.

IP Broadcast Address

This setting specifies the server's IP address that is used in IP Broadcast messages. You cannot change this setting while the server has any active Telnet sessions. The default address is 255.255.255.255.

```
Xyplex>> define server IP broadcast address
         172.19.255.255
Xyplex>> set server IP broadcast address
         172.19.255.255
```

IP Primary and Secondary Gateways

An access server can use an IP gateway (or router) to send data packets to nodes on remote IP networks. You can use the following command to define up to two gateways, called the primary and secondary. The server first attempts to use the primary gateway; if it is unsuccessful (because the gateway is down or unreachable), it attempts to use the secondary. The default primary and secondary gateway addresses are 0.0.0.0.

```
Xyplex>> define server IP primary gateway 172.19.1.2
Xyplex>> set server IP primary gateway 172.19.1.2
Xyplex>> define server IP secondary gateway
         172.19.1.3
Xyplex>> set server IP secondary gateway 172.19.1.3
```

Show/List/Monitor Server IP Characteristics

Use this command to view the current settings for IP-related settings. Figure 11 shows a sample display:

```
Xyplex> show server ip char

MAXserver V6.0    Rom 440000 HW 00.00.00 Lat Protocol V5.2 Uptime: 3 06:18:21

Address:   08-00-87-03-45-67   Name:   X034567              Number:     0

Identification:  Xyplex Access Server

Internet Address:           172.18.240.23       Internet TTL:               64

Internet Broadcast Address: 255.255.255.255     Translation Table TTL:      60

Local Base:                 4000                Local Increment:           100

Routing Table Size:         64                  TCP Retransmit:            640

Domain Name:

Default Domain Suffix:

Domain TTL:                 0                   IP Reassembly:         DISABLED

Primary Domain Address:     172.18.130.200      TCP Resequencing:      DISABLED

Secondary Domain Address:   0.0.0.0             TCP Connect Timer:          32

Primary Gateway Address:    172.18.128.1

Secondary Gateway Address:  0.0.0.0

Gateway Timeout:            60

Subnet Mask:                255.255.128.0

Subnet Mask Auto-Configure: DISABLED
```

Figure 11. Server IP Characteristics Display

Configuring Username and Password Prompts

You can configure your username and password prompts. To do this, use the following command syntax:

```
SET/DEF PORT [port-list] USERNAME PROMPT ["string"]

SET/DEF PORT [port-list] PASSWORD PROMPT ["string"]
```

The default username/password prompt length is 26 characters.

If the server booted from the default parameters, the default values are, "`Enter username>`" and "`Enter user password>`."

If the server booted from an existing parameter file, the username prompt is, "`Enter username>`."

For the password prompt, the default value is "`Enter user password>`." However, if SecurID is enabled on the port, the default password prompt is "`Enter PASSCODE:`."

Use the SHOW PORT ALTERNATE CHAR command to display the current prompt settings.

Modem and Port Setup

This section explains how to configure the access server to support simple modem applications using any of several different types of modems. The typical applications performed over these modems include interactive activities such as terminal emulation, electronic mail, file transfers using Kermit, Xmodem, Microphone, TCP/IP, FTP, LAT, etc., or PPP or SLIP connections using low speed modems, etc. There are additional activities that you must perform for PPP and SLIP connections. These are described in later sections of this guide.

This section discusses the following topics:

- Basic Modem Port Setup

- Setting Up Dial-In Ports
- Setting Up Dial-Out Ports (also Dial-In/Dial-Out Ports)
- Setting Up Dial-Back Ports

The examples in this section only include the options that must be changed.

Basic Modem Port Setup

When connecting a modem to a port, you must configure the port so that its settings match those of the modem connected to it. Port characteristics are set with SET/DEFINE PORT commands. For most devices, the default settings for nearly all of the PORT characteristics are appropriate. This section highlights the changes that you will need to make. If a local service is available at several ports (for example, a modem pool), you must set the appropriate port characteristics for all the ports offering the local service.

Defining Ports Back to Defaults

A privileged user can define ports back to factory default settings. The following parameters are not changed (if enabled) when the ports are reset to defaults:

- IP security
- IP filters
- IPX filters

To reset ports to default settings, use the following command:

`DEFINE PORT [`*`port-list`*`] TO DEFAULTS`

The system will prompt you for verification on each specified port.

Press Return to reset the factory defaults or press any other key to terminates the process. When you press any other key, this terminates the default process from that port on. The ports that have already been returned to factory defaults will stay defaulted.

Log out from the port in order for the changes to take effect.

Modem Control issues. This refers to issues involving dialing and answer control, monitoring the DCD modem signal to determine when a session has been disconnected, and knowing when to assert the DTR modem signal to the modem. For most ports to which a modem is attached, standard modem control operation is used. An exception to this rule is a port which is configured to support only ARAP connection. In this case, modem control operation is disabled because ARAP uses a CCL script to control the modem activities (when APD is enabled at the port, even when only ARAP connections are accepted at that port, APD will disable modem control when it detects that an incoming call is an ARAP connection,).

Since modem control is disabled by default, typically you must enable modem control at the port, as shown in the following command (using port 8 as an example):

```
Xyplex>> define port 8 modem control enabled
```

NOTE: Do not use this command if the port is dedicated to ARAP connections. Use this command under all other circumstances where a modem is connected to the port.

You must usually disable DSRLOGOUT, since the DCD signal, not the DSR signal is used to determine when a modem session has been disconnected, as shown in the following command (using port 8 as an example):

```
Xyplex>> define port 8 dsrlogout disabled
```

You may also specify when the port should assert the DTR modem control signal with this command:

```
Xyplex>> define port 12 dtrwait value
```

For a modem which supports the Ring (RI) signal, set *value* to FORRING if you want the port to assert DTR only after the modem asserts RI, or to DISABLED if you want the port to always assert DTR. For a modem which does not support the Ring (RI) signal, set *value* to DISABLED.

Speed or Autobaud There are three ways in which to set a port speed, when making modem connections:

- "**autobauding**" **the port**. The Autobaud characteristic is enabled by default on all serial ports. When enabled, the port automatically matches the baud rate of the modem when the user presses the Return key a few times at initialization time. For the access server to use the autobaud feature, however, the modem must use 8-bit no parity or 7-bit even parity characters. If the characters cannot be set to 8-bit no parity or 7-bit even parity, you must disable the PORT AUTOBAUD characteristic, and individually set the PORT SPEED, CHARACTER SIZE, and PARITY characteristics to the appropriate values.

  You cannot use autobauding for APD ports, high-speed connections (connections where the port speed is 38,400 bps or higher), dial-out or dial-back connections, for ports which will make ARAP connections, or for ports where you plan to use a CCL script. If you need to disable autobauding, use a command similar to:

  ```
  Xyplex>> define port 6 autobaud disabled
  ```

- **Using a CCL script to ascertain and set the port speed**. One could think of a CCL as an "alternate" method of autobauding the serial port connected to the modem, since the CCL script will determine the appropriate port speed and set it accordingly. In effect, the modem performs the autobauding. You should note that the CCL script for your modem might not support all possible port speeds, particularly higher speeds.

  CCL scripts are required for ARAP connections. They can also be used to initialize the port and "program" the modem for other types of connections (PPP, SLIP, interactive, etc) even at ports where ARAP connections are not used. “Using CCL Scripts” covers the steps that you must take to configure a port to use a CCL script. If you plan to use a CCL script at a port, disable autobauding as described above.

- **Defining a fixed speed for the port**. You must use this method for high-speed connections (connections where the port speed is 38,400 bps or higher), or for dial-out or dial-back connections when modem control is enabled (i.e., all connections other than ARAP connections), for situations where the characters from the modem cannot be set to 8-bit no parity or 7-bit even parity, or where a CCL script is not used at the port. Typically, you will set the port speed to match the modem speed. The following example command would be used for this:

  ```
  Xyplex>> define port 6 speed 38400
  ```

Flow Control Flow control is often used in modem connections to prevent data from being lost. Appendix A covers flow control in more detail.

The default setting for the PORT FLOW CONTROL characteristic at all serial ports is XON (XON/XOFF). Typically, for modem connections, flow control is set to CTS (RTS/CTS flow control is used and XON/ XOFF flow control is turned off) or DISABLED (all flow control methods are turned off). When using a high-speed modem, you might need to use the CTS/RTS flow control (whether you do or not is modem-dependent - refer to the owner's manual supplied with the modem to determine the modem's flow control requirements). When using a low-speed modem, you can usually disable flow control. Both flow control methods prevent stray XOFF characters from stopping operations.

To alter the PORT FLOW CONTROL characteristic, use a command similar to the following:

`Xyplex>> define port` *`port-list`* `flow control cts (`or `disabled)`

When using modem or port speeds above 14,400 bps, one should use hardware flow control (DEFINE PORT FLOW CONTROL CTS) because XON/XOFF flow control characters can become embedded in the data stream and not be recognized.

NOTE: In PPP applications, it is possible to use a high-speed connection without using hardware flow control, by modifying the PPP Character Map to mask out the XON/XOFF characters (hardware flow control is much easier to use).

Ports which support the 8-wire cabling method can also use concurrent RTS/CTS hardware flow control. Refer to the section "Information about Xyplex Cabling Methods" for more information.

Setting Up Dial-In Ports

Dial-in ports provide local access connections to services on the network. Dial-in ports only accept connections made to the serial port, not connections originated from the local area network. Most of the default values for port characteristics support dial-in ports, but you do need to change the settings for some characteristics.

- Perform the Basic Modem Port Setup procedure (near the beginning of this section).
- Specify the type of access allowed to the port. Use the command:

```
Xyplex>> define port 8-12 access local
```

Dedicated Services

To ensure security, many dial-in ports limit connections to only one interactive host service (LAT or TELNET), called a *dedicated* service. The following are some examples of how to define a dedicated service at one or more ports:

This command assigns a dedicated service named ACCOUNTING to port 6:

```
Xyplex>> define port 6 dedicated service accounting
```

This command assigns a dedicated service with the Internet-address 192.12.119.184 to port 6:

```
Xyplex>> define port 6 dedicated service 192.12.119.184
```

This command specifies that when a user logs in to port 6, the port connects directly to the dedicated service:

```
Xyplex>> define port 6 autodedicated enabled
```

If you do this, you can also assign a permanent username for the port, so that you can identify the port more easily:

```
Xyplex>> define port 6 username "dial-in"
```

Setting Up a Dial-Out Port

Dial-out ports provide connections from devices on the local area network to devices accessible via telephone lines. The telephone call is initiated by a device connected to the local area network. Most of the default values for PORT characteristics are satisfactory. You will need to make the following changes:

- Perform the Basic Modem Port Setup procedure (near the beginning of this section).
- Specify the type of access allowed to the port. Any port that is defined as a service must be set up to accept remote connections. If you want the port to be able to originate connections, but not accept them from the modem, use a command such as:

  ```
  Xyplex>> define port 8 access remote
  ```

  If you want the port to be able to both originate and accept connections (for example, a modem that will be used for both dial-in and dial-out), use a command such as:

  ```
  Xyplex>> define port 8 access dynamic
  ```

- After the port is set up to communicate with the modem, you must configure a dial-out service at the server so that the network can communicate with the dial-out modem port. Dial-out support requires a program, such as FTP or Kermit running at a host or PC, which can connect to the local service, and then direct modem specific commands (such as dialing information) to the modem.

LAT Dial-Out Services To set up a LAT dial-out service, you must use SET/DEFINE SERVICE commands to specify service characteristics, as well as the SET/DEFINE PORT commands to specify appropriate port characteristics. For example, to define a dial-out service named MODEM, at port 1 of the server, you would use a command such as:

```
Xyplex>> define service modem port 1 enabled
Xyplex>> define service modem connections enabled
```

You could also assign an identification message for server displays, using the DEFINE SERVICE IDENTIFICATION characteristic. A user at a VMS host can connect to a LAT application port that maps to the LAT local service, as described in the next section.

TCP/IP Dial-Out Services To set up a TCP/IP dial-out service, you assign a Telnet remote port number to the ports which offer that service. For example, if two ports on the server offer the TCP/IP dial-out service (for example, a bank of dial-out modems), you would assign the same Telnet remote port number at both ports. Users (for example, at PCs or UNIX hosts) then connect to that service by connecting to the Internet-address of the server and specify the Telnet remote port number assigned to the ports. To assign a Telnet remote port number, use a command similar to this:

```
Xyplex>> define port 10 telnet remote port 3600
```

TCP/IP services can also be created by assigning an *internet-address* or *domain-name* to one or more ports (e.g., creating an Internet "rotary"). Use the DEFINE SERVER INTERNET ROTARY command. For example:

```
Xyplex>> define server internet rotary 112.132.11.1 5-8
```

- A user at a PC whose serial port is connected to a server port, can then connect to the dial-out service while running Kermit or a similar program. A user at a host that uses Internet Protocols, such as a UNIX host, would connect to the server *internet-address/telnet-remote-port* or *domain-name/telnet-remote-port* while running Kermit, FTP, etc.

Configuring a LAT Application Port at a VMS Host

To support dial-out ports, the only action that you need to take at the VMS host is to create a LAT application port using the LAT control program (LATCP). The basic steps are as follows:

1. Start the LAT Control Program.

   ```
   $ RUN SYS$SYSTEM:LATCP
   ```

   LATCP displays the LCP> prompt.

2. Create a LAT application port.

   LCP>CREATE PORT LTA*nnn*: /LOG

   where LTA*nnn*: represents a device, and *nnn* is a decimal number. You can also use the /NOLOG qualifier.

3. Map the applications ports to specific servers, or ports on the server.

   LCP>SET PORT LTA*nnn*: /APPLICATION /NODE=*node-name* - /SERVICE=*service-name* /PORT=*port-name*

   where *node-name* refers to the server, *service-name* refers to a local service offered by the server, and *port-name* refers to a server port. This example shows both a *service-name* and a *port-name*, although only one of these is necessary. (The *node-name* used in the remote connection request must match the server name specified by the DEFINE/SET SERVER NAME command. This name is not necessarily the same as the DECnet node name for the server.).

4. Exit from LATCP.

   ```
   LCP>EXIT
   ```

   In this configuration, a user at the LAT host can initiate the connection from a file transfer program like the VMS version of Kermit. In this case, the user would issue a Kermit connect command, such as:

   CONNECT *terminal-name*

   where *terminal-name* refers to a LAT application port, such as a device named LTA123:, that the system manager created with LATCP.

Setting Up Dial-Back Ports

Dial-back ports combine the characteristics of a dial-in port and a dial-out port. Like dial-in ports, they provide local access connections to services on the network. However, they provide this by having the server port instruct the modem to dial the telephone of the user who wants to log in.

The dialback feature uses the Network Command Script feature. The commands are contained in a file, called a script file, which is stored at a host called a *script server*. The script server can be a host system that supports the Trivial File Transfer Protocol (TFTP) or a Xyplex MAXserver unit that can load files over the network, such as a MAXserver 1800 or 1820 ACCESS SERVER. See the *Advanced Features Guide* for more information about creating scripts.

For a dialback port, you must create a dialback script, which contains the information that tells the modem which telephone number to dial when a specific user attempts to log on to the server through a modem. If no script file for the user is found, the user will not be able to login. If a script file is found for the user, the server will cause the modem to dial back that user at a designated telephone number. You can use the dialback script in conjunction with a login script for dialback ports.

This section describes how you set up a dialback port. The following specific activities are involved:

- Using Dial-Back Scripts on the access server
- Configuring port settings
- Setting Up a Dial-Back Script Server

Using Dial-Back Scripts on the Access Server

To use script files from the access server, you must specify the Internet destinations (internet-address or domain-name) and directory locations where the server can request script files. You also specify which ports will use or require a script file for login. The following procedure describes the steps to take at access servers which use scripts.

At the access server which will use script files, define one or more script servers, using the privileged DEFINE SERVER SCRIPT SERVER command. For example, the following commands designate a script server where all username directories and the common script are located in the directory path `/tftpboot/SCRIPTS`. The `/tftpboot` directory is the TFTP home directory of the host which has the domain-name UNIXHOST.XYPLEX.COM.

```
Xyplex>> define server script server unixhost.xyplex.com
"/tftpboot/SCRIPTS"

Xyplex>> set server script server unixhost.xyplex.com
"/tftpboot/SCRIPTS"
```

You can specify up to four script servers for each server unit.

Configuring Port Settings

Most of the default values for port settings are satisfactory. You need to make the following changes to dialback port and modem-related characteristics. The following examples use port 12:

- Perform the Basic Modem Port Setup procedure (near the beginning of this section).
- Specify the type of access allowed to the port. Dial-back ports both originate and accept connections. Therefore use a command such as:

  ```
  Xyplex>> define port 12 access dynamic
  ```

- Specify that the port is a dial-back port with the command:

  ```
  Xyplex>> define port 12 dialback enabled
  ```

- To change the amount of time which the remote modem has in which to respond to a dial-back attempt, use this command:

  ```
  Xyplex>> define port 12 dialback timeout time
  ```

  where *time* is between 0 and 60 seconds (default is 20 seconds).

Setting Up a Dial-Back Script Server

To use the network command script feature, you must specify information at the script server and the access server. Complete the following steps to configure each script server:

NOTE: To use scripts, Telnet must be enabled on the server, and an internet-address, and optionally a domain name must be specified for the server.

a. Determine which UNIX host system or MAXserver 1800/1820 access servers will be the script servers. You can use multiple hosts for backup, which can be a combination of script server types. Each access server can have up to four script servers.

b. Set up directories to contain script files at each script server. For a UNIX host script server, you need to consider the TFTP guidelines in the next section, as well.

   Create a directory to contain the dialback script file for each user who will have one. The directory name must match the name that user will specify when logging on to the port at the `Enter username>` prompt. At a given script server, all username directories must be located in the same directory. For ease of management, you can create a directory just for script files, rather than use a directory that already contains many files, such as `/usr`, `/bin`, `/tftpboot`, or `/etc` on a UNIX host. The username and the directory name cannot include space or tab characters.

Figure 12 illustrates how to set up the directories to contain script files at a UNIX host. In the figure, the user whose username is "gjones" has both a login script file and a dialback script file which contains the information that tells the modem which telephone number to dial when the user gjones attempts to log on to the server through a modem. For example, when a user logs in as gjones, the access server request the file `/tftpboot/SCRIPTS/gjones/dialback` from this script server (in this example, `/tftpboot` is the TFTP home directory for this host). Then the connection is dropped, the dialback script executes, and the user is dialed back.

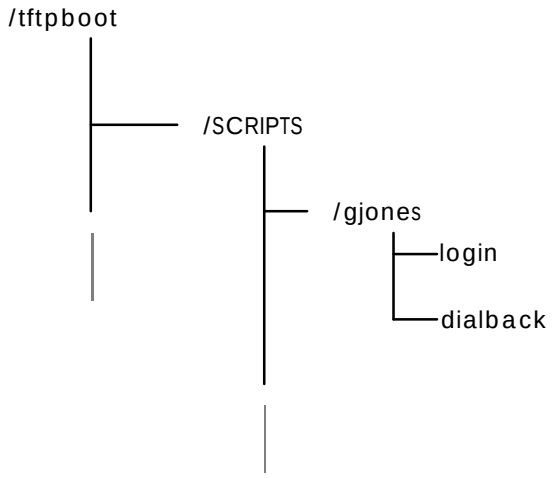


Figure 12. Example Script Server Directory Structure

The port username must match the directory name at the host for the access server to locate a custom script file for a user. Therefore, users need to type in the correct user name when they log in to use their custom login script file.

The following examples creates a directory named SCRIPTS, and a username directory for a user whose login name will be "gjones" on a UNIX host and a MAXserver 1800/1820:

UNIX Host

% mkdir SCRIPTS
% cd SCRIPTS
% mkdir gjones
% cd gjones

MAXserver 1800/1820

Take the MAXserver system disk to a DOS-based personal computer (PC) to create a directory for each user. For example:

```
C:> mkdir SCRIPTS
C:> cd SCRIPTS
C:> mkdir gjones
C:> cd gjones
```

For additional users, the directory for each username would be a sub-directory of the SCRIPTS directory.

c. At the UNIX host or PC, use a text editor to create the script file, which contain the instructions that permit the server to dialback to another modem. The name of file is dialback. At a UNIX host the file name must consist of all lower-case letters (`dialback`). When creating the file, follow the syntax rules listed in the next section. Refer to the *Software Management Guide* for more information about Scripts.

Script File Structure and Guidelines

Observe the following rules when developing a dialback script file:

- The first line in the script is always the following:

  ```
  #control_script
  ```

- You can include commands which require user input, such as a command that requires a password. The unit will prompt the user for the password or other input before continuing (the user prompt will be displayed, regardless of the setting of the PORT SCRIPT ECHO characteristic).
- Each line of a script file can be up to 132 characters long. Each line of a script file must contain only one command. Each command must be on only one line.
- Within command scripts, the server software recognizes the character (#) as a flag for special operations. When the pound character is the first non-space character on a command line, the server attempts to treat the contents of the line as control information that it must interpret. When followed by a space or tab, the pound character indicates a comment; the server ignores the remainder of the line.

  In a dialback script, the pound character, when followed by the word modem and one or more spaces, specifies a modem command that the server will pass on to a modem. The phrase "#modem" must be in lower case letters.

NOTE: If you plan to use a dialback script for a port that is configured to use AppleTalk Remote Access Protocol (ARAP), refer to “Setting Up ARAP” for additional information.

Directory Requirements

Script files are downloaded to units through the TFTP protocol. UNIX systems usually require that you locate all files that TFTP will transfer on the network in the TFTP "home directory." Most UNIX systems provide a way for you to specify the TFTP home directory or use a default home directory. The default TFTP home directory varies from system to system. Follow the configuration instructions for the TFTP daemon (tftpd) in the system documentation (MAN pages, etc) to determine how to locate the TFTP home directory.

On Sun Workstations, for example, the MAN page for `tftpd` says that the home directory is specified in the `/etc/inetd.conf` file, and that the factory default home directory is `/tftpboot`. Therefore, you would examine the tftp entry in the `/etc/inetd.conf` file to see if the host is using the default home directory or a user-specified home directory. Place the script files (and/or script sub-directory) in the TFTP home directory.

For ease of configuration (for example, adding users) or to prevent the TFTP home directory from becoming too cluttered, it may be desirable to locate script files in a directory other than the TFTP home directory. To do this, you must create a link from the directory containing the script files to the TFTP home directory, so that the TFTP daemon will know where to locate the files. Create this link and give it appropriate file permissions using commands in the form (note, you must be superuser):

```
# cd tftp-home-directory
# ln -s tftp-home-directory script-directory
# chmod 777 script-directory
```

For example, on Sun Workstations, using the default TFTP home directory, `/tftpboot`, and a directory named `/SCRIPTS` as the top-level directory in which script files are stored, you would use the commands:

```
# cd /tftpboot
# ln -s /tftpboot SCRIPTS
# chmod 777 SCRIPTS
```

Determine if any TFTP security mechanisms exist on your UNIX system. Some TFTP implementations do not limit the directories that TFTP can access, which can present a security risk at some sites. Other implementations do limit TFTP to certain directories. In this case, you must place all files in a particular home directory, or in a subdirectory of the home directory. If the files are not located there, TFTP will not find them. For example, SunOS, and some others, use a TFTP daemon -s (secure) option, that restricts TFTP to a particular directory and its subdirectories. Sun Workstations are normally configured with this option enabled. If you examine the `/etc/inetd.conf` file, you will see an entry similar to `-s /tftpboot` in the `tftpd` entry. Other vendors may use a different method. Read the MAN page on `tftp`, `tftpd`, and `inetd.conf` to determine the directory/security requirements on your UNIX system.

Script File Execution and Processing

The server executes script files either when the user logs in to an appropriately configured port, or when the user issues the SCRIPT command. The following steps describe what happens during script file processing and execution:

1. The user attempts to log on to a dialback port. The user specifies a username when the `Enter Username>` prompt appears. The server immediately disconnects the telephone connection. The server uses the port username to locate the script file and to reauthenticate the user and the port speed.

2. The access server requests the TFTP process at each script server to download a specific script file. The file to be downloaded is determined, as follows:

   The access server requests a script file named "dialback". from a directory location which is based on two items: the *pathname* specified in the DEFINE/SET SERVER SCRIPT SERVER command and the username of the port (the server removes spaces from the username to locate the script file). The DEFINE/SET SERVER SCRIPT SERVER command designates the top-level directories to be searched, the username designates the lowest directory to be searched. For example, the following command to specifies a script server at address 192.12.119.184 and a top-level path name of `/usr/xyplex`

```
Xyplex>> define server script server 192.12.119.184
   "/usr/xyplex"
```

   When a user named "John A. Smith" logs on to a port, the server requests the script file `/usr/xyplex/JohnA.Smith/dialback` from the script server at address 192.12.119.184. If the script is not found in the first directory location, the TFTP process searches the directory immediately above it.

3. If the access server finds the file at a script server within thirty (30) seconds, the script server downloads the script file to the server through TFTP.

   If the access server does not find the file at any script server within thirty (30) seconds, it logs out the port.

4. The access server reads the entire script file into its memory, before it executes the commands in the script. The port passes the dialing information to the modem which then dials the remote modem. The remote modem has only a limited time to respond (the amount is set by the PORT DIALBACK TIMEOUT characteristic). If the remote modem does not respond within the specified time, or if the line is busy, the server logs out the port and drops the connection. If the remote modem does respond within the specified time, the server begins the normal login sequence. When the `Enter Username>` prompt appears again, the name the user enters must match the name originally entered in Step 1, or the port is logged out and the connection is dropped. If the port is set up to use or require a login script, the server unit requests and executes this script. Refer to the *Advanced Configuration Guide* section which describes login scripts in detail.

   Kerberos and other security measures can provide additional security.

   The following is an example of a dialback script:

```
#control_script

# This is a dialback script.

#modem atdt5551978
```

Port Settings

Use the SHOW PORT CHARACTERISTICS command to display the current settings for a port. If you do not specify a port number, The settings for port 0 display.

```
Port 4:                                    02 Dec 1998  10:50:51

Character Size:            8           Input Speed:       38400
Flow Control:            XON           Output Speed:      38400
Parity:                 None           Modem Control:  Disabled
Access:                Local           Local Switch:       None
Backwards Switch:       None           Name:             PORT_4
Break:                 Local           Session Limit:         4
Break Length:          250ms           Type:                Soft
Forwards Switch:        None
CCL Modem Speaker: Inaudible           CCL Name:            None
APD Timeout:       Unlimited           APD Default:       LOGOUT
APD:                Disabled
Dialout Action:       Logout
APD Authentication
Interactive Only:   Disabled

Preferred Service: None

Authorized Groups: 0
(Current)  Groups: 0
Enabled Characteristics:
Autobaud,  Autoprompt,  Broadcast,  Input Flow Control,  Internet
Connections,
Line Editor,  Loss Notification,  Message Codes,  OutboundSecurity,
Output Flow Control,  ULI,  Verification
```

PPP Support

The Access Server Software supports two PPP Network Control Protocols (NCP) which are used to establish and configure network layer protocols. The NCPs supported include the IP Control Protocol (IPCP, also known as IP over PPP) and , also known as IPX over PPP).

This section describes how to set up and enable IPCP and IPXCP on the access server. It also describes several typical network configurations that use PPP to support connections between different devices on Internet networks (IP) or Novell NetWare networks (IPX). The specific topics that are covered include:

- Enabling Protocols on the Server
- Configuring a PPP Port for Modem Support
- Configuring PPP
- Configuring IPCP Connections
- Configuring IPXCP Connections
- Configuring IP and IPX Filtering

NOTES: PPP requires at least 2 megabytes of memory and the enhanced load image on MAXserver Access Servers.

Enabling Protocols On the Server

PPP and IPX are configurable features, which are disabled by default. PPP must be enabled on the server in order to use IPCP or IPXCP. IPX must also be enabled on the server in order to use IPXCP. (When PPP is enabled, IPCP support is automatically enabled.)

- The following example shows how to enable PPP on the access server:

  ```
  Xyplex>> define server protocol ppp enabled
  ```

- The following example shows how to enable IPX on the access server:

  ```
  Xyplex>> define server protocol ipx enabled
  ```

  The server responds with the following prompt:

  ```
  Press <RETURN> to modify configuration, any other key
  to abort.
  ```

  Press the RETURN key when you see this prompt. The server displays the following message:

  ```
  -705- Change leaves approximately nnnnn bytes free.
  Xyplex>>
  ```

- Use the SHOW SERVER PARAMETER command to verify that all parameter servers are "Current." Then re-initialize the unit, so that the change takes effect. You can use the command:

  ```
  Xyplex>> initialize delay 0
  ```

Configuring a PPP Port for Modem Support

You must make sure to configure the proper modem-related settings and to use the correct cabling. See the Getting Started Guide for cabling information.

Configuring PPP

The basic steps for setting up a server to support IPCP and IPXCP connections are:

1. Enable PPP at specific ports, or use APD.
2. Specify optional PPP port settings.

3. After the port has been configured for PPP operation, you must perform additional steps that are specific to the NCP (IPCP or IPXCP) being configured. These are covered in the sections "Configuring IPCP Connections" and "Configuring IPXCP Connections."

Enabling PPP at Specific Ports

After you enable PPP on the access server, you must enable it on individual ports. This can be done either by setting up the port to accept multiple protocols with APD or setting up the port so that only PPP is used on it, using one of the following commands:

```
DEFINE PORT port-list  PPP ENABLED/DISABLED
SET PORT PPP ENABLED/DISABLED
```

The DEFINE command dedicates the port to PPP. The SET command only enables PPP until the user disconnects from the port.

Examples:

```
Xyplex>> define port 6-12 ppp enabled

Xyplex>> set port ppp enabled
```

After you enable PPP on one or more ports, you can also specify PPP characteristics, although the default values for these characteristics may be appropriate for your implementation.

NOTE: If you use a SET command at your port to enable PPP, PPP processing begins immediately. You will not see the Xyplex> prompt until the port is logged out and logged on again.

Optional PPP Port Settings

There are several optional PPP port settings available, depending on the needs of your site. See the *Commands Reference Manual* for a detailed description of these commands.

- Enable negotiation options with remote devices

  DEFINE/SET PORT [*port-list]* PPP ACTIVE ENABLED/DISABLED

- Reset port PPP settings to default values

  DEFINE PORT [*port-list]* PPP DEFAULTS ENABLED

- Specify the time limit that a user can be logged in to a port, regardless of the activity on a port.

  DEFINE/SET PORT [*port-list*] LOGIN DURATION [*time-logged-in*]

  The valid values are from 0 to 480 minutes. This is a privileged command and can only be applied to ports in local access mode. The default setting is 0, which indicates no time limit is set.

- Specify a PPP port to be mapped to a small subnet of IP addresses.

  DEFINE/SET PORT [*port-list*] IP MASK [*ip-address*]

- Specify a range of IP addresses that cannot be overwritten by remote clients

  DEFINE/SET PORT [port-number] PPP IP LOCAL ADDRESS RANGE [0.0.0.0 - 255.255.255.255]

- Specify how many seconds the port will wait to retry negotiations

  DEFINE/SET PORT [*port-list]* PPP RESTART TIMER [*number-of-seconds]*

- Specify how many attempts the port will make to negotiate.

  DEFINE PORT *port-list* PPP CONFIGURE LIMIT [*number-of-attempts*]

- Specify how many times the port can refuse a proposed PPP option, before rejection.

  DEFINE PORT *port-list* PPP FAILURE LIMIT [*number-of-refusals*]

- Specify the ASCII control characters that the port can negotiate to control how data is transferred between the two sides of the PPP connection.

  DEFINE/SET PORT *port-list* PPP CHARMAP [*nnnnnnnn*]

- Specify how often the specified port(s) will send a Link Control Protocol (LCP) echo request packet over the PPP link to the connection partner.

  DEFINE/SET PORT *port-list* PPP KEEPALIVE TIMER [*time]*

- Specify how many seconds the specified port(s) should wait to receive a Link Control Protocol (LCP) echo reply packet from the connection partner before terminating the PPP link.

  DEFINE/SET PORT *port-list* PPP KEEPALIVE TIMEOUT [*time]*

- Specify whether or not PPP negotiation packets will be logged in the verbose accounting log, and the format in which they will be logged. Valid values for setting include NONE, INTERPRETED, or RAW. The default is NONE. This should only be used as a diagnostic tool in the event of interoperability problems.

  DEFINE/SET PORT *port-list* PPP LOGGING [*setting]*

Configurable Username and Password Prompts

You can configure your username and password prompts. To do this, use the following command syntax:

```
SET/DEF PO # USERNAME PROMPT "string"
SET/DEF PO # PASSWORD PROMPT "string"
```

The default username/password prompt length is 26 characters.

If the server booted from the default parameters, the default values are, "`Enter username>`" and "`Enter user password>`."

If the server booted from an existing parameter file, the username prompt is, "`Enter username>`."

For the password prompt, the default value is "`Enter user password>`." However, if SecurID is enabled on the port, the default password prompt is "`Enter PASSCODE:`."

These new prompts are displayed on the SHOW PORT ALTERNATE CHAR screen.

```
XYPLEX>> show port alt char

Port 0:  a                          05 Jan 1900  09:54:04

Resolve Service:    Any_Lat      DTR wait:           Disabled
Idle Timeout:              0            Typeahead Size:           128
SLIP Address:              N/A          SLIP Mask:                N/A
Remote SLIP Addr:   N/A          Default Session Mode:     Interactive
TCP Window Size:    256          Prompt:                   X021812
DCD Timeout:        N/A          Dialback Timeout:         N/A
Stop Bits:          N/A          Script Login:
       Disabled
TCP Keepalive Timer:N/A          Username Filtering:       None
Nested Menu:        Disabled     Nested Menu Top Level:    0
Command Size:              132          Clear Security Entries:
       Disabled
Rlogin Transparent Mode:       N/A          Login Duration:           0
Xon Send Timer:     N/A          RADIUS Accounting:        Disabled

Username Prompt:           Enter username>
Password Prompt:           Enter user password>
```

Configuring IPCP Connections

After the port has been configured for PPP operation, you must configure IPCP. The basic steps to configure IPCP include:

1. Assigning Local and/or remote IP Addresses
2. Specifying optional IPCP PORT characteristics.
3. Optionally, you might want to configure static IP routes.
4. Optionally, you might want to configure a unit to use IP filtering features. (Covered later in this section.)

This section also shows sample IPCP single-node and network configurations.

Assigning Local and Remote IP Addresses to PPP Ports

The network topology at your site determines whether you need to assign local or remote IP addresses to PPP ports. You can, for example, specify a remote IP address at a PPP port so that the interface will assign that address to a PPP device that connects to the port. Later in this section, the section that describes a network with a PC having no configured Internet address explains how to use a remote IP address in this situation.

Most of the time you do not need to assign a local IP address to a port because the PPP interface uses the access server's Internet address as a local address. The local IP address can be useful in certain two-node configurations where you have serial connections at two PPP ports.

The format for the commands that assign these addresses are the following:

```
DEFINE/SET PORT port-list PPP IP REMOTE ADDRESS [internet-address]
DEFINE/SET PORT port-list PPP IP LOCAL ADDRESS [internet-address]
```

Generally, for dial-in ports, you will want to assign a REMOTE ADDRESS. If you do not do this, the user can configure the remote PC to have any internet-address. This can pose a security risk or result in the remote PC being assigned to an incorrect subnet or duplicating an existing address.

Specifying Optional IPCP Port Characteristics

There are several optional PPP port settings available depending on the needs of your site. See the *Commands Reference Manual* for a detailed description of these commands. The optional commands include:

- DEFINE/SET PORT *port-list* PPP IP ENABLED/DISABLED

  The command specifies whether or not a PPP port can negotiate use of the IP protocol (IPCP). Enabled (the default) means that the port will negotiate use of the IP protocol when the user attempts to connect via IPCP, effectively allowing the connection. Disabled means that the port will not negotiate use of the IP protocol when the user attempts to connect via IPCP. One might disable IPCP if the port is to be used exclusively for IPXCP connections, or to temporarily disable IPCP connections.

- DEFINE PORT *port-list* PPP IP BROADCASTS ENABLED/DISABLED
  SET PORT *port-list* PPP IP BROADCASTS ENABLED/DISABLED

  These commands specify whether or not a port will transfer Internet broadcast packets over the PPP link.

- DEFINE PORT *port-list* PPP IP VJ COMPRESSION ENABLED/DISABLED
  SET PORT *port-list* PPP IP VJ COMPRESSION ENABLED/DISABLED

  These commands specify whether or not a port will negotiate the use of Van Jacobsen (VJ) data compression on the Internet link.

- DEFINE PORT *port-list* PPP IP VJ COMPRESSION SLOTS [*n*]
  SET PORT *port-list* PPP IP VJ COMPRESSION SLOTS [*n*]

  These commands specify the number of data channels which will use VJ data compression.

- DEFINE PORT *port-list* PPP IP REMOTE ADDRESS RANGE *addr-range*
  SET PORT *port-list* PPP IP REMOTE ADDRESS RANGE *addr-range*

  These commands specify the range of internet-addresses that the PPP link will allow to be negotiated. Internet addresses outside the range will not be permitted by the link. Valid values for *addr-range* are two internet-addresses separated by a hyphen. The first internet-address in the *addr-range* represents the lowest acceptable address. The second internet-address in the *addr-range* represents the highest acceptable address.

Specifying IP Static Routes

The Commands Reference Guide provides a detailed description of the DEFINE/SET SERVER INTERNET ROUTE command. You use this command to specify static IP routes.

Examples of IPCP Single-Node Configurations

This section includes three examples of single node configurations. The differences among them depend on whether or not a PC running PPP has an assigned Internet address and whether or not the PC exists within the same subnet as the access server. The three configurations are these:

- A PC With an Internet Address Within the LAN Subnet
- A PC With an Internet Address Outside of the LAN Subnet
- A PC With No Configured Internet Address

In the diagrams in this section, PPP is enabled on a Xyplex access server. A PC and an unspecified device are connected to asynchronous ports on the access server. The access server is attached to a LAN with other IP devices, such as various UNIX hosts and Internet Routers. The access server has an Internet address and a default subnet mask, which the access server assigns automatically when you specify the access server's Internet address. Some devices exist within the same subnet as the access server and some do not.

A PC With an Internet Address Within the LAN Subnet

Figure 13 shows a PC attached to the access server with an Internet address within the same subnet as the access server. The PC connection can be direct or through a modem. A router is attached to the LAN. The PPP protocol is enabled on the access server and the appropriate asynchronous ports.

The PPP port on the access server "learns" the IP address of the PC when the PC gains access to the port. The destinations that the PC can reach through the access server depend on whether or not the router is defined as an Internet Gateway on the access server.

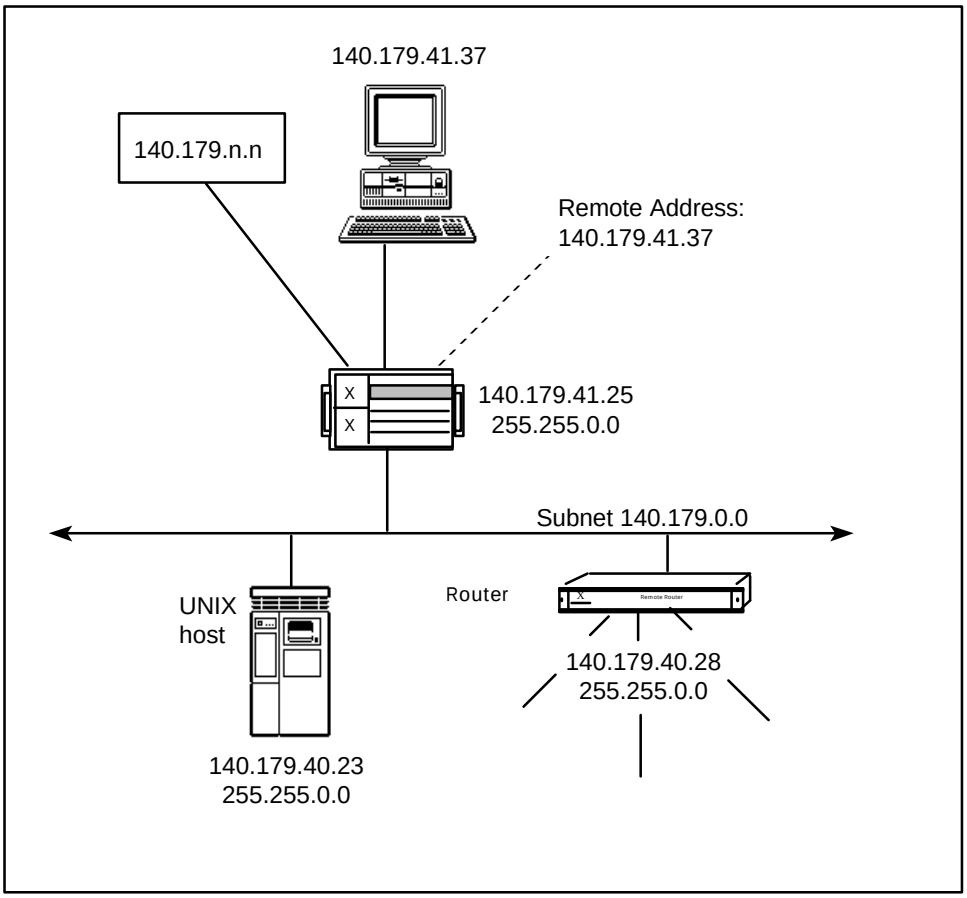


Figure 13. A PC with an Internet Address Within the LAN Subnet

Without a defined Gateway, the PC can use PPP to reach the access server, other devices directly attached to the access server on a serial line such as device 140.179.n.n, and all devices on the LAN within the same IP subnet (140.179.0.0). With the Router defined as a primary Internet gateway on the access server, the PC can also reach IP addresses outside of the local subnet through the Router.

Using the Internet address of the remote router in Figure 13, the command has this form:

```
Xyplex>> define server internet primary gateway address
140.179.40.28
```

The command interface assigns a default subnet mask when you define the gateway address.

A PC With an Internet Address Outside of the LAN Subnet

Figure 14 shows a PC attached to the access server with an Internet address that is not within the same subnet as the access server. The PC connection can be direct or through a modem. A router is attached to the LAN. The PPP protocol is enabled on the access server and the appropriate asynchronous ports.

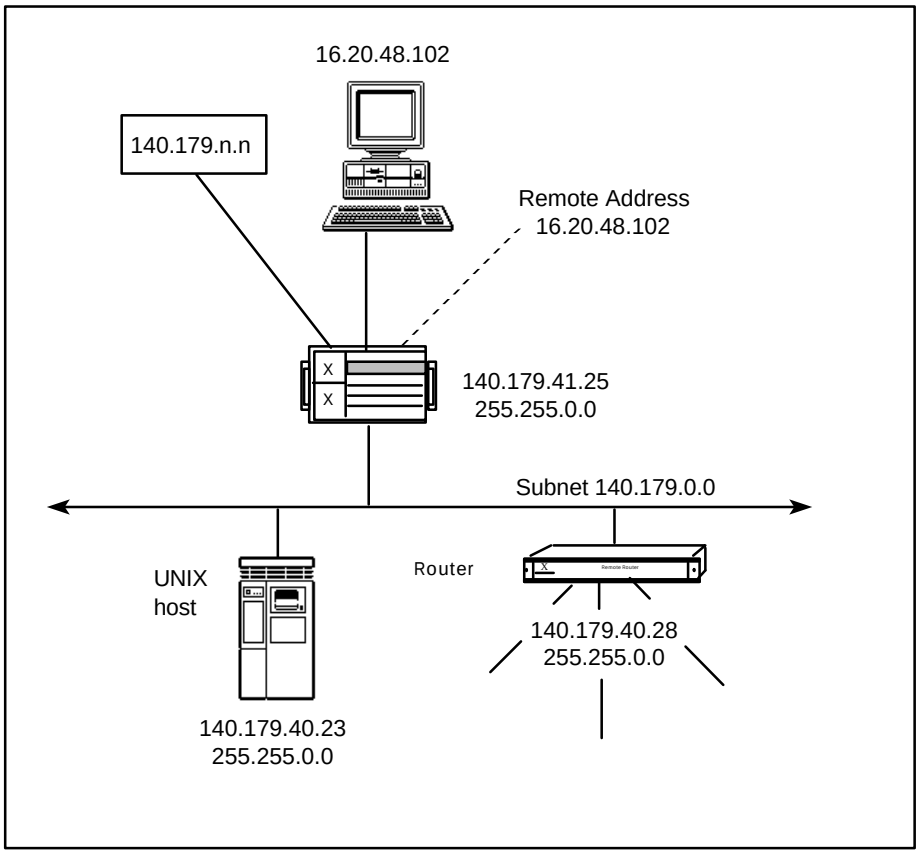


Figure 14. A PC With an Internet Address Outside of the LAN Subnet

When the PC is not on the same subnet as the access server, you must configure a routing entry for the PC on the access server. You also must specify routing information on the LAN devices or on a router if one exists on the LAN. The access server can then identify the address from the remote network and act as a router for the remote PC when the PC attempts to access devices on the access server's local subnet. The LAN devices can send network traffic from the local network back to the PC either through the access server or through the router.

Using the Internet addresses of the access server and the PC in Figure 14, the following command defines the access server as a router for the remote PC:

```
Xyplex>> define server internet route 16.20.48.102 gateway
140.179.41.25
```

If you use the access server as a router for the LAN devices, you can locally configure a route-to-host entry at each UNIX device on the LAN subnet. This specifies the path to the PC on the remote subnet. Most UNIX devices support a `route add host` command, which can identify the access server (140.179.41.25) as the router to use to gain access to the PC (16.20.48.102).

Instead of defining a route-to-host entry on each LAN host device, you can define a route-to-host entry on the router to act as an Internet gateway. In this case, a UNIX device on the LAN sends PPP traffic to the default router, which then forwards the traffic to the access server. If you also configure the router as the Internet gateway on the access server as in the previous example , the PC has access to Internet addresses available through the router.

A PC With No Configured Internet Address

Figure 15 shows a PC without an Internet address attached to the access server. The PC connection can be direct or through a modem, and the PC can reside in the same subnet as the access server or in a remote subnet. A router is attached to the LAN. The PPP protocol is enabled on the access server and the appropriate asynchronous ports.

When a the PC does not have an Internet address, the PPP port on the access server can assign an address to the PC when it negotiates the PPP link. You specify the address at the PPP port prior to link negotiation time. For example, this command assigns the address 140.179.41.37 to PPP port 10 on the access server:

```
Xyplex>> define port 10 ppp ip remote address 140.179.41.37
```

When the PC on the remote subnet attempts to connect to port 10, the port assigns this address to it.

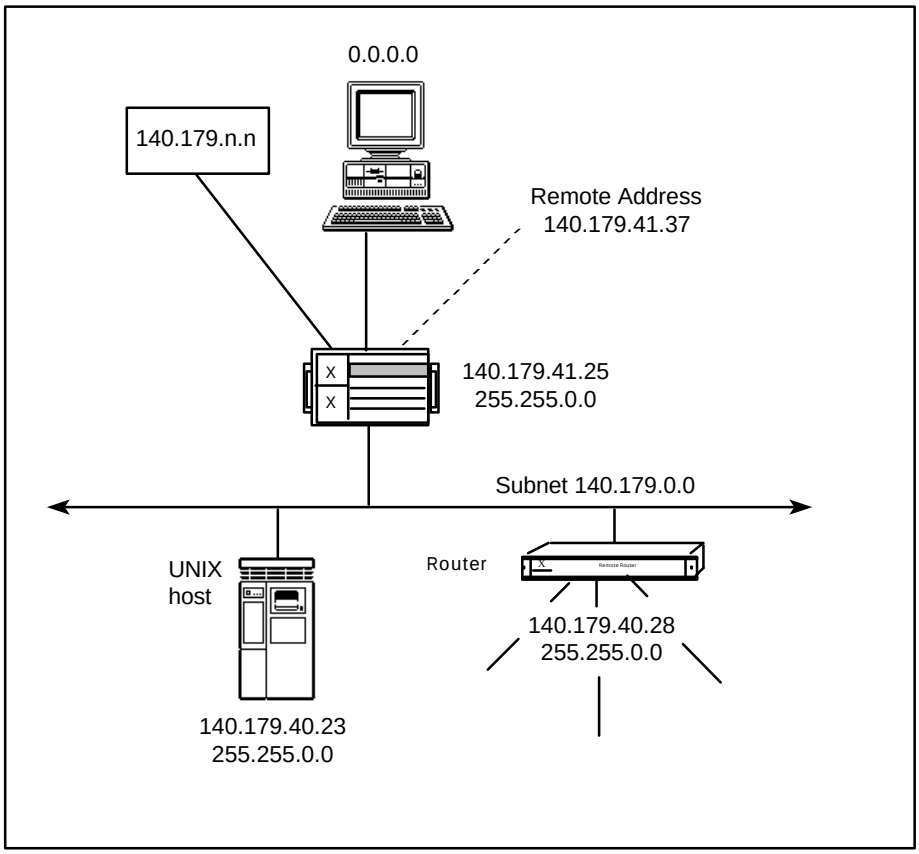


Figure 15. A PC With No Configured Internet Address

Similar routing issues as in the last example apply for IPCP devices without a configured Internet address as for those with a configured Internet address. If you assign an Internet address in a remote subnet to the PPP device, however, you can configure routing information on the other network devices prior to the Initial PPP connection. If the PC has an Internet address, however, you must wait for the initial connection when the access server "learns" the Internet address of the device.

Example of an IPCP Network Configuration

This section shows an example of a network configuration. This configuration requires two access servers, connected over a serial line. The two access servers connect separate LANs through PPP. Figure 16 shows the two LANs, LAN A and LAN B, connected by two access servers running PPP.

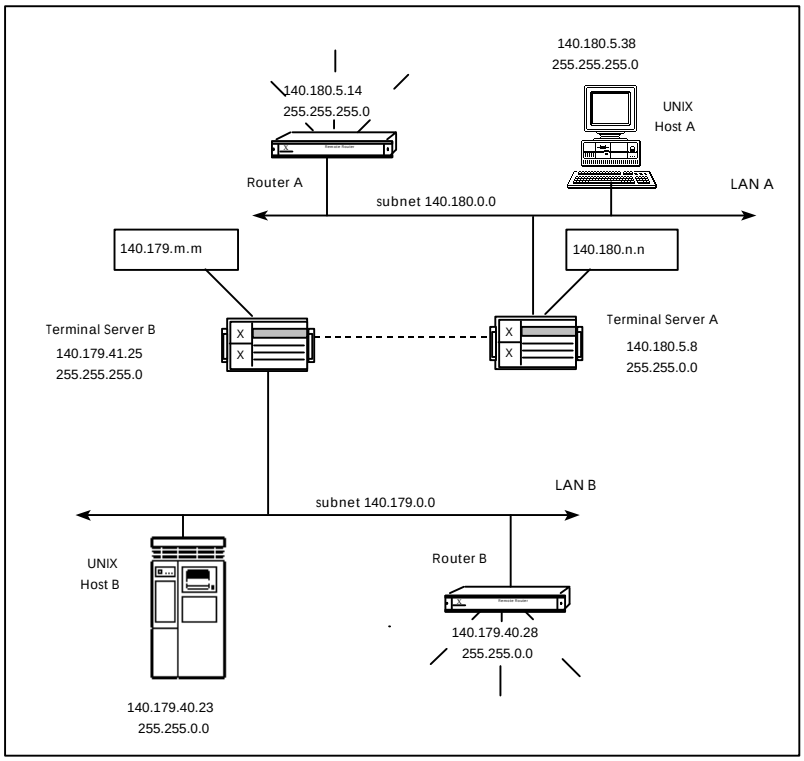


Figure 16. Two Communication Servers in a Back-To-Back Gateway

To configure LAN-to-LAN connectivity with two access servers running PPP, you must define Internet addresses on both access servers, and assign static routes on each access server to identify the path to the remote subnets. You then assign routing entries on the LAN devices which either identify the access server as the router to the remote subnets, or define a default router, if one exists on the LAN.

Using the Internet addresses in Figure 16, the following command defines access server A as the router to the subnet where access server B resides:

```
Xyplex>> define server internet route 140.179.41.25 gateway
140.180.5.8 mask 255.255.0.0
```

The following command defines access server B as the router to the subnet where access server A resides:

```
Xyplex>> define server internet route 140.180.5.8 gateway
140.179.41.25 mask 255.255.0.0
```

To gain access to a remote subnet, a device on the LAN must use the access server as a router, or send network traffic to a router on the LAN, if one exists, which can then send the traffic to the access server. Most UNIX hosts support a `route add host` command which identifies devices which act as routers to a remote subnet.

In Figure 16, for example, you can define access server B (140.179.41.25) as the router for UNIX host B to use when it attempts to gain access to UNIX host A (140.180.5.38) on LAN A. You can also define Router B (140.179.40.28) as the default router to use when UNIX host B attempts to reach a device on LAN A (subnet 140.180.0.0).

Configuring IPXCP Connections

Overview

After the port has been configured for PPP operation, you must configure IPXCP characteristics. The basic steps for configuring this application are:

1. Specify SERVER characteristics.
2. Specify PORT characteristics.
3. Configure IPX clients with the client software. Refer to the documentation supplied with the IPX client software package for more information.

   The access server can communicate with any RFC 1552-compliant IPXCP (IPX over PPP) client software implementation (e.g., a version of the Stampede Remote Office client software). Using this software, users at the remote IPX clients, such as remote (dial-in) workstations or PCs, have access to the Novell services offered on the Novell Netware network (unless the network manager chooses to limit that access). The user dials in when he or she needs access to the services, and disconnects when the services are no longer needed. This is a typical "remote office" or "user-to-LAN" application.

4. Optionally, you might want to configure a unit to use static IPX RIP routes or SAP services.
5. Optionally, you might want to configure a unit to use IPX routing or filtering. (This is covered later in this section.)

Specify IPXCP-Related SERVER Settings

You must specify a number of server settings which allow the server to operate as an IPX node on the Ethernet network. The following is a summary of these SERVER characteristics. Refer to the *Commands Reference Guide* for more information about these commands.

NOTE: In order for the changes, specified by the DEFINE SERVER commands listed below, to take effect, you must re-initialize the server after issuing the commands.

The Xyplex access server can accept two packet types over an IPX Interface: Ethernet packets and IEEE 802.3 (MAC) packets. You can only use one of these types at a time on a server. (Ethernet packets and IEEE 802.3 packets have different formats[1].) By factory default, the server is configured to use Ethernet-type packets for IPX. Use the following command to specify the IPX protocol used:

```
DEFINE SERVER IPX PROTOCOL ETHERNET/MAC ENABLED/DISABLED
```

Example:

```
Xyplex>> define server ipx protocol ethernet enabled
```

The IPX protocol specification requires that IPX networks be identified by a network number. This permits efficient routing of packets to their destinations. Each device in a given IPX network must know its network number. Communications servers can obtain a network number in one of two ways: the server can "learn" its network number from other IPX devices (such as a Novell file server) that is connected to the same Ethernet network, or the server manager can assign a network number.

An access server actually uses a minimum of three unique network numbers. One network number is used for traffic that is sent or received on the Ethernet network. Another network number is used for traffic that is sent over a given PPP link (setting this up is covered later), and a third network number is an "internal" network number, which is used inside the server for transferring information between the Ethernet network and the PPP link(s). This internal network number must not be used elsewhere in the Novell NetWare network (i.e., must be unique).

---

1 IEEE 802.3 (MAC) packets have a 2-byte LENGTH field, where Ethernet packets have a 2-byte TYPE field.

Use the following command to specify an IPX network number to be used for communication between the server and devices on the Ethernet network, or to specify that the server should learn its network number from other IPX devices that is connected to the same Ethernet network:

```
DEFINE SERVER IPX NETWORK network-number
```

Valid values for *network-number* are hexadecimal numbers between 0 (the default) and FFFFFFFE. When the *network-number* is set to 0, the server will learn its network number from other IPX devices on the Ethernet network to which it is connected. You would tend to specify a network-number when the server is connected to an Ethernet network that does not include other IPX devices (i.e., a "quiet" network).

Use the following command to specify an internal IPX network number:

```
DEFINE SERVER IPX INTERNAL NETWORK network-number
```

Valid values for *network-number* are hexadecimal numbers between 1 (the default) and FFFFFFFE. The network number must not be used elsewhere in the Novell NetWare network.

Example:

```
Xyplex>> define server ipx network fffffffe
Xyplex>> define server ipx internal network 2
```

SERVER IPX RIP Settings

The following are optional SERVER characteristics which control RIP-related activity on the Ethernet connection of the access server:

- `DEFINE/SET SERVER IPX RIP [BROADCAST] setting`

  This command specifies whether or not the server will broadcast RIP information to other devices on the Ethernet network, and if the information is broadcast, how much information the server will send. Valid choices for *setting* are: FULL, CHANGE, and NONE. FULL means that the server will broadcast the entire contents of the RIP table. CHANGE means that the server will only broadcast new or changed routing information. NONE means that the server will not broadcast any routing information. The default is FULL.

- `DEFINE/SET SERVER IPX RIP [BROADCAST] TIMER timer`

  This command specifies how frequently the access server will broadcast RIP information on the Ethernet network. Valid values for *timer* are whole numbers between 0 and 4294967295 (seconds). The default interval is 60 seconds.

- `DEFINE/SET SERVER IPX RIP [BROADCAST] DISCARD TIMEOUT timer-multiple`

  This command specifies how long the server keeps RIP information that it receives from other devices connected to the Ethernet network. The *timer-multiple* that you specify is multiplied by the value you specify in the DEFINE/SET SERVER IPX RIP [BROADCAST] TIMER *time* command. Valid values for *timer-multiple* are whole numbers between 0 and 4294967295. The default is 3.

- DEFINE SERVER IPX RIP [MAXIMUM] TABLE SIZE *table-size*

  This command specifies the maximum number of entries in the IPX Router Information Protocol (RIP) table. If you change this value, the change will take effect after you re-initialize the server. Valid values for *table-size* are whole numbers between 0 to 16000. If you specify 0 (the default) the server can maintain an unlimited number of entries.

SERVER IPX SAP Settings

- DEFINE/SET SERVER IPX SAP [BROADCAST] *setting*

  This command specifies whether or not the server will broadcast SAP information to other devices on the Ethernet network, and if the information is broadcast, how much information the server will send. Valid choices for *setting* are: FULL, CHANGE, and NONE. FULL means that the server will broadcast the entire contents of the SAP table. CHANGE means that the server will only broadcast new or changed SAP information. NONE means that the server will not broadcast any SAP information. The default is FULL.

- DEFINE/SET SERVER IPX SAP [BROADCAST] TIMER *timer*

  This command specifies how frequently the access server will broadcast SAP information on the Ethernet network. Valid values for *timer* are whole numbers between 0 and 4294967295 (seconds). The default interval is 60 seconds.

- DEFINE/SET SERVER IPX SAP [BROADCAST] DISCARD TIMEOUT *timer-multiple*

  This command specifies how long the server keeps SAP information that it receives from other devices connected to the Ethernet network. The *timer-multiple* that you specify is multiplied by the value you specify in the DEFINE/SET SERVER IPX SAP [BROADCAST] TIMER *time* command. Valid values for *timer-multiple* are whole numbers between 0 and 4294967295. The default is 3.

- `DEFINE SERVER IPX SAP [MAXIMUM] TABLE SIZE` *`table-size`*

  This command specifies the maximum number of entries in the IPX Service Advertisement Protocol (SAP) table. If you change this value, the change will take effect after you re-initialize the server. Valid values for *table-size* are whole numbers between 0 to 16000. If you specify 0 (the default) the server can maintain an unlimited number of entries.

Specify PORT Characteristics

PORT settings control IPX-related activity over PPP links. The following is a summary of these PORT characteristics that you must set to allow user-to-LAN connections:

Basic PORT IPX Characteristics

- The following command enables a PPP port to negotiate use of the IPX protocol:

  `DEFINE/SET PORT` *`port-list`* `[PPP] IPX ENABLED/DISABLED`

  Enabled means that the port will negotiate use of the IPX protocol when the user attempts to connect via IPX/PPP, effectively allowing the connection. Disabled means that the port will not negotiate use of the IPX protocol when the user attempts to connect via IPX/PPP.

- As mentioned previously, IPX networks are identified by a network number, and the server uses a minimum of three unique network numbers, one of which is used for traffic that is sent over a given PPP link. Servers can obtain the network number for traffic that is sent over a PPP link in one of two ways: the server can "learn" its network number from other IPX devices (such as a Novell file server), or the server manager can assign a network number. Use the following command to configure the IPX network number for the port (i.e., the PPP link):

DEFINE/SET PORT *port-list* [PPP] IPX *network-number*

Valid values for *network-number* are hexadecimal numbers between 0 (the default) and FFFFFFFE. A *network-number* of 0 means that the port will learn its network number from the remote PPP device(s). The network number must not be used elsewhere in the network.

- Individual devices within a Novell NetWare network are identified by node-numbers. The server can either learn the node number by which it will be identified over the PPP link, or the server manager can specify a permanent node-number. The server notifies its connection partner of its node-number when the link is being established. The following command configures the IPX node number for the port (i.e., the PPP link)

  DEFINE/SET PORT *port-list* [PPP] IPX [REMOTE] NODE *node-number*

  Valid values for *node-number* are hexadecimal numbers between 0 (the default) and FFFFFFFFFFFE. When the *node-number* is set to 0, the port will learn its node number from the remote PPP device(s). The combination *network-number* and *node-number* must not be used elsewhere in a given Novell NetWare network.

PORT IPX RIP Characteristics.

- DEFINE/SET PORT *port-list* IPX RIP [BROADCAST] *setting*

  This command specifies whether or not the server will broadcast RIP information over the serial link to the remote partner, and if the information is broadcast, how much information the server will send. Valid choices for *setting* include: FULL, CHANGE, and NONE. FULL means that the server will broadcast the entire contents of the RIP table. CHANGE means that the server will only broadcast new or changed RIP information. NONE means that the server will not broadcast any RIP information. The default is CHANGE.

- DEFINE/SET PORT *port-list* IPX RIP [BROADCAST] TIMER *timer*

  This command specifies how frequently the access server will broadcast RIP information over the serial link to the remote partner. Valid values for *timer* are whole numbers between 0 and 4294967295 (seconds). The default interval is 60 seconds.

- DEFINE/SET PORT *port-list* IPX RIP [BROADCAST] DISCARD TIMEOUT *timer-multiple*

  This command specifies how long the server keeps RIP information that it receives over the serial link to the remote partner. The *timer-multiple* that you specify is multiplied by the value you specify in the DEFINE/SET SERVER IPX RIP [BROADCAST] TIMER *time* command. Valid values for *timer-multiple* are whole numbers between 0 and 4294967295. The default is 3.

PORT IPX SAP Characteristics

- DEFINE/SET PORT port-list [PPP] IPX SAP [BROADCAST] *setting*

  This command specifies whether or not the PORT will broadcast SAP information over the serial link to the remote partner, and if the information is broadcast, how much information the PORT will send. Valid choices for *setting* include: FULL, CHANGE, and NONE. FULL means that the PORT will broadcast the entire contents of the SAP table. CHANGE means that the PORT will only broadcast new or changed SAP information. NONE means that the PORT will not broadcast any SAP information. The default is CHANGE.

- DEFINE/SET PORT *port-list* [PPP] IPX SAP [BROADCAST] TIMER *timer*

  This command specifies how frequently the communication PORT will broadcast SAP information over the serial link to the remote partner. Valid values for *timer* are whole numbers between 0 and 4294967295 (seconds). The default interval is 60 seconds.

- DEFINE/SET PORT *port-list* [PPP] IPX SAP [BROADCAST] DISCARD TIMEOUT *timer-multiple*

  This command specifies how long the server keeps SAP information that it receives over the serial link to the remote partner. The *timer-multiple* that you specify is multiplied by the value you specify in the DEFINE/SET SERVER IPX SAP [BROADCAST] TIMER *time* command. Valid values for *timer-multiple* are whole numbers between 0 and 4294967295. The default is 3.

Specify Static Routes and Services

- DEFINE/SET SERVER IPX RIP *interface* NETWORK *network-number* [HOPS *hops*] [TIME *time*] [FORWARDING ROUTER *router*]

  This command specifies a static route. The *interface* can be either ETHERNET or *port-number*. The *network-number* identifies the unique IPX network where the destination device is located. Valid values for *network-number* are hexadecimal numbers between 1 (the default) and FFFFFFFE. Hops refers to the number of IPX routers that the packet must pass through in order to reach the destination. Valid values for *hops* are 1 through 15. The default is 10. Time refers to the number of timer "ticks" necessary to reach the final destination. Valid values for time are between 1 and 65535. The default is 400. A forwarding router is one through which a destination network can be reached. Valid values for router are hexadecimal numbers between 1 (the default) and FFFFFFFFFFFE.

- DEFINE/SET SERVER IPX SAP [SERVICE] "*name*" TYPE *type* NETWORK *network-number* NODE *node-number* SOCKET *socket-number* [HOPS *hops*]

  This command specifies a static service. Valid names can be between 1 and 47 characters long and contain characters a through z (both upper- and lower-case), the numbers 0 through 9, the underscore character (_), the hyphen character (-), and the at-sign character (@). Valid values for *network-number* are hexadecimal numbers between 1 (the default) and FFFFFFFE. Valid values for *node-number* are hexadecimal numbers between 1 and FFFFFFFFFFFE. Hops refers to the number of IPX routers that the packet must pass through in order to reach the destination. Valid values for *hops* are 1 through 15. The default is 10.

Configuring Ports to Use SLIP and CSLIP

The Access Server software enables a user to run Internet protocols over an asynchronous serial line, using the Serial Line Internet Protocol (SLIP). SLIP is defined by the Internet RFC 1055. SLIP is automatically enabled when the TELNET feature is enabled.

SLIP links can transmit and receive packets that have been compressed using the Van Jacobson compression algorithm. Links using Compressed SLIP are referred to as CSLIP links.

This section covers the following topics:

- Configuring Ports To Use SLIP and CSLIP
- SLIP Sessions
- Example Configurations

You must configure ports appropriately to support SLIP connections. The settings that must be used depend on your SLIP application. Also, refer to "Information About Xyplex Cabling Methods" to make sure that you are using the correct cables.

The basic activities include:

- **Configuring Modem Support for SLIP Links.** This is only necessary when using a dial-in SLIP application.
- **Enabling SLIP/CSLIP at Specific Ports.**
- **Assigning SLIP Addresses.** Not all SLIP applications require this.

These activities are covered in the remainder of this section.

Configuring Modem Support for SLIP Links

You must make sure to configure the proper modem-related characteristics and to use the correct cabling. "Information About Xyplex Cabling Methods" provides cabling details. "Port Settings" covers how to set up a port to support dial-in, dial-out, or dial-back capabilities.

Enabling SLIP/CSLIP at Specific Ports

You must enable SLIP/CSLIP on individual ports. This can be done either by setting up the port to accept multiple protocols with APD, or setting the port up so that only SLIP/CSLIP is used on it, using one of the following commands:

```
DEFINE/SET PORT port-list  INTERNET SLIP ENABLED/DISABLED
DEFINE/SET PORT port-list  INTERNET CSLIP ENABLED/DISABLED
```

The difference between these commands only matters in applications where the port will initiate communication over the link. For situations where the port initiates activity on the SLIP link, you must specify whether or not the port can initiate communications with a remote device using CSLIP packets (using the DEFINE/SET PORT INTERNET CSLIP ENABLED/DISABLED command). When the use of compressed SLIP is enabled, the port will immediately begin transmitting compressed packets on the serial link.

NOTE: In situations where the remote device initiates activity on the link, the port automatically detects whether or not the remote device is using compressed SLIP packets. The port uses the same type (compressed or uncompressed) of packets as the remote device.

When compression is in use, a number of sessions (or slots) using higher-level protocols, such as TCP/IP, can operate across a CSLIP link. This can happen, for example, when the link is used in a gateway configuration that supports several users, or in a configuration where a single node (such as a dial-in PC) is connected to the port and the single node has several windows in use. RFC 1144 allows a CSLIP link to use a maximum of 16 slots. (This is because the compression mechanism is very memory intensive. If too many slots use compression, the server or the remote device could run out of memory resources to perform other tasks.) When compression is in use on a link, the server will allocate sufficient memory to support 16 slots (the maximum permitted), regardless of the number of slots that will actually be used on the link. If the remote device only supports fewer slots, that number will be the actual number of slots used on the link.

You can examine the "Enabled Characteristics" field on the SHOW/LIST/MONITOR PORT CHARACTERISTICS display to determine if the port can initiate activity on the SLIP link using compressed SLIP packets. If it is enabled, "CSLIP" will be listed.

NOTE: If you use a SET command at your port to enable SLIP/CSLIP, processing begins immediately and you will not see the Xyplex> command prompt until the port is logged out and logged on again.

Examples:

```
Xyplex>> define port 6-12 internet slip enabled
Xyplex>> define port 6-12 internet cslip enabled
Xyplex>> set port internet slip enabled
```

Automatic Sending of SLIP Information

Use this command to enable/disable automatic sending of SLIP address information. With this command enabled, the following addresses are returned when you issue the `SET PORT IP SLIP ENABLE` command:

- SLIP remote address

- SLIP local address
- SLIP Mask address

Use the SHOW PORT ALT CHARACTERISTICS command to display the current status of SLIP Autosend.

NOTE: A "Set" can only be done on the port you are currently on. All other ports are define only.

Syntax

```
DEFINE PORT <port-list> IP SLIP AUTOSEND [ENABLED]
                                         [DISABLED]
```

Where	Means
ENABLED	Allow SLIP addresses to be automatically sent.
DISABLED	Do not allow SLIP addresses to be automatically sent.

Example `DEFINE PORT 4 IP SLIP AUTOSEND ENABLED`

Assigning SLIP Addresses to Ports

Both the port (the local end of a SLIP connection) and the remote device must each have an Internet address assigned to them for the purpose of establishing a connection and forwarding data. The Internet address of the port is referred to as the local address. The Internet address of the remote device is referred to as the remote address. During the period when the SLIP link is being established, both sides of the link communicate their addresses to each other. In some configurations, one side of the link might not have a pre-assigned Internet address. When the port has been configured this way, the port will learn its address from the partner. In this case, the port will assume the address of the remote device, which is contained in the first packet sent to it by the remote device.

When the remote device has been configured this way, it can learn its address from the server if it is capable of making a bootp request. (SLIP links cannot be established if the remote device does not have an Internet address and is incapable of requesting one.)

The network topology at your site determines whether you need to assign local and/or remote SLIP addresses to SLIP ports. The format for the commands that assign these addresses are the following:

```
DEFINE PORT port-number INTERNET SLIP ADDRESS port-address
REMOTE remote-address MASK network-mask
```

In this command, the *port-address* represents a local Internet address that the port will use. If you do not specify a unique port-address, the link will use the address of the access server itself. The *remote-address* is the Internet address that the port will assign to a remote device that does not know its address. The port will communicate this information while the link is being initialized. The *network-mask* specifies the Internet addresses on the local area network to which the remote device may have access. The server discards packets forwarded to it by the remote device which do not match the network-mask. The server passes packets which do match the network-mask to the local area network.

Most of the time you do not need to assign a local SLIP address to a port because the SLIP interface uses the access server's Internet address as a local address. The local SLIP address can be useful in certain network configurations where you have serial connections at two SLIP ports.

SLIP Sessions

Ports can be configured with Automatic Protocol Detection enabled or can be dedicated only for SLIP/CSLIP connections. For ports which are dedicated for SLIP/CSLIP connections, when you enable SLIP on a port, the port expects only SLIP or CSLIP packets from the remote location. Each packet is transformed into an IP packet and then forwarded to the destination Internet address. All packets received from the local network, and destined for the device or network connected to the port, are put in SLIP or CSLIP packets and forwarded over the serial link.

To terminate a SLIP session, you disconnect the dialup link or log out the SLIP port through another port on the access server. SLIP processing terminates when you log out the port.

If a port has a dedicated connection to the remote location, you can use the DEFINE PORT INTERNET SLIP ENABLED command to establish a permanent SLIP link. In this case, the only way to disable SLIP on the port is to use the DEFINE PORT INTERNET SLIP DISABLED command and then log the port out from another port.

Sample Configurations

The access server software supports two models for the utilization of SLIP: the single-node model and the network model. The following sections contain examples of each.

Single-Node Applications

Direct Connection of a Host to a Serial Port

This configuration is used to connect a host, workstation, or PC directly to the network through a connection to a serial port. One might use this configuration in order to connect a host that does not have an Ethernet connection to the network. Figure 17 depicts this configuration.

NOTE: If this is a DTE to DTE connection, the configuration will use "null-modem" cabling.

To configure this connection, assign the local Internet address to the SLIP port. The remote device supplies its own address. Since the idea is to allow the remote device to be part of the network, a special SLIP *network-mask* (subnet mask) is not needed. The access server will assign 255.255.255.255 as the SLIP *network-mask*. For example, to assign a local address of 182.13.130.1 to port 8 of the access server shown in Figure 17, and assign no *remote-address* (meaning that the remote device must supply its own address) or special *network-mask*, use the command:

```
Xyplex>> define port 8 internet slip address 182.13.130.1
```

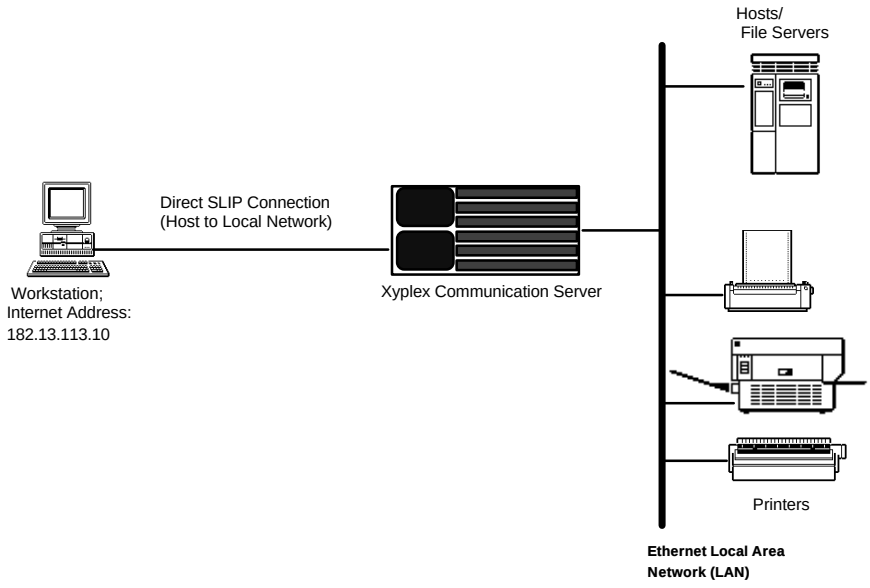


Figure 17. Direct SLIP Connection

Dial-In SLIP Connection

This configuration connects a remote (dial-in) Workstation or PC to the network. Figure 18 depicts this configuration. To configure this connection, use 0.0.0.0 (the default) as the local Internet address of the SLIP port, the remote-address, and SLIP network-mask. When the PC/workstation dials in and initiates a SLIP session, the port learns the PC/workstation's Internet-address and assigns the local-address and remote-address to be the Internet-address learned from the remote device. The server sets the SLIP network-mask to be 255.255.255.255. To do this, use the command:

```
Xyplex>> def port 8 intern slip addr 0.0.0.0 remote 0.0.0.0 mask
0.0.0.0
```

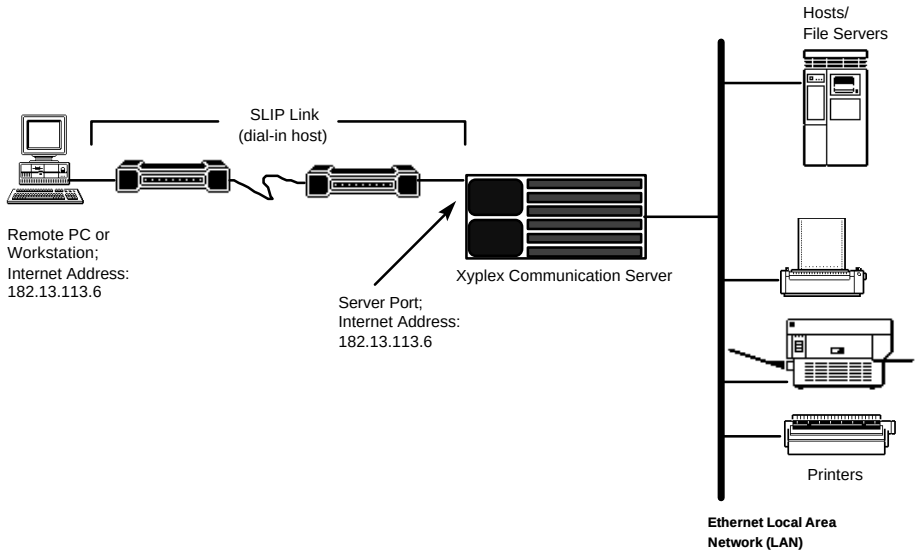


Figure 18. Dial-In SLIP Connection

This configuration allows different PC/Workstations with different Internet-addresses to dial in to the same port without having to reconfigure SLIP information each time. It requires that the first connection must be initiated from the PC/Workstation, not the network. This is because the server does not know the SLIP information until it receives the first packet from the PC/Workstation.

In the example shown in Figure 18, the first packet will contain the Internet address of the remote PC/Workstation (182.12.113.6) and will assign that address as the local address for the link. Packets addressed to 182.12.113.6 will be forwarded over the SLIP link to the remote device.

Network Applications

This configuration is used to connect a remote network to the local network through a serial port. In this application, the port functions as a gateway connecting two networks. Figure 19 depicts this configuration. To configure this connection, you must assign a local Internet address, a remote-address, and a SLIP network-mask to the SLIP port.

For example, to forward packets between the local network (Internet addresses 182.13.113.x) to a remote network (Internet addresses 182.13.130.x), you would use the command:

```
Xyplex>> def port 8 intern slip addr  182.13.113.5 remote
182.13.130.5 mask 255.255.255.0
```

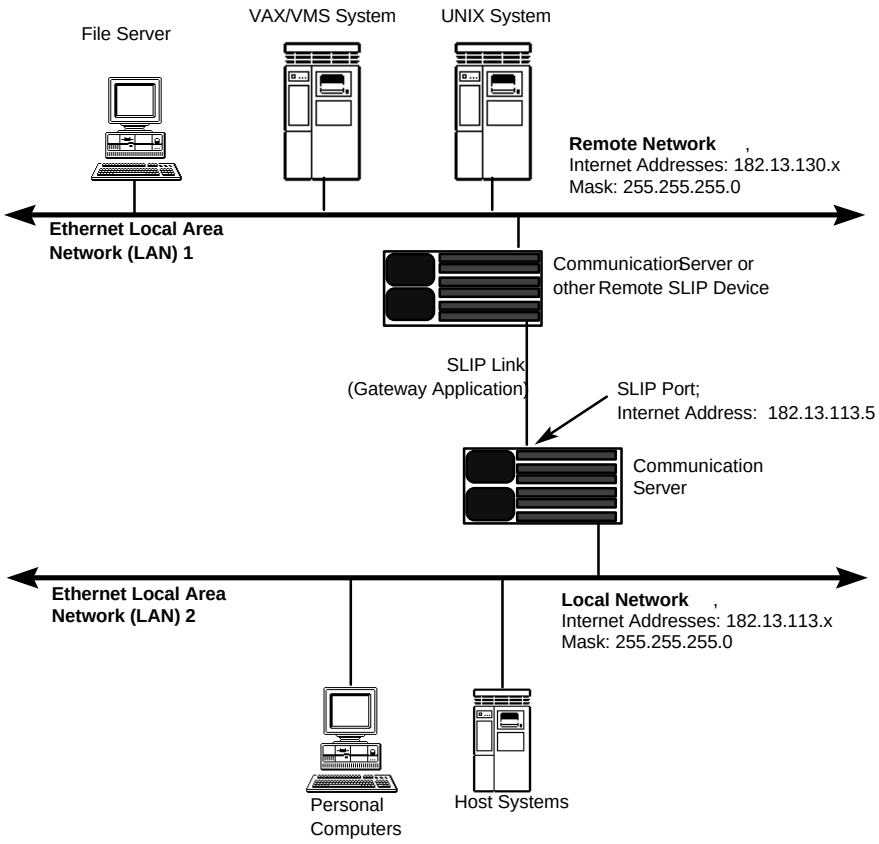


Figure 19. SLIP Connections to Remote Network

ARAP Configuration

This section describes how to configure the AppleTalk Remote Access Protocol (ARAP) on an access server. The topics contained in this Section are:

- ARAP Setup
- Using ARAP With Authentication and Dialback Features
- Modifying Dialback Scripts for ARAP Ports
- ARAP Planning Considerations
- Diagnostic Cabling

To configure an Access Server to support AppleTalk Remote Access connections is fairly straight-forward. The basic steps for setting up Remote Access clients are:

1. Enable the server ARAP protocol
2. Specify SERVER settings
3. Specify PORT settings
4. Install CCL scripts (portions of this topic are covered in “Using CCL Scripts”).

Optional Steps can include:

5. Configure Authentication Methods for Server, Ports, and Hosts
6. Edit CCL scripts to support authentication. This is only needed if you are using Kerberos and/or SecurID authentication and you have a CCL script which does not contain Xyplex modifications to support these authentication methods. Xyplex supplies CCL scripts for many modem models which are already modified appropriately. This topic is covered in “Using CCL Scripts”.

Each of the activities listed above is covered in this section or in “Using CCL Scripts”. This section also includes information about using ARAP and security or authentication methods concurrently.

ARAP is a configurable feature, which is disabled by default. You must obtain a password from Xyplex to enable ARAP. For information about obtaining a password, contact your local Xyplex Sales Representative or distributor.

When enabled, ARAP occupies approximately 160 Kbytes of server memory. Each port which has ARAP enabled requires 43 Kbytes of server memory. You may also need to increase the number of packet buffers available to the server for buffering data. (This is covered in the section titled "Specify SERVER Characteristics.")

Use the following command to enable the ARAP protocol on the access server:

```
Xyplex>> define server protocol arap enabled
```

The server will respond with the following prompt:

```
ARAP Password>
```

Enter the protocol password at this password prompt. The server will not "echo" the protocol password to the display. Press the <RETURN> key. When you supply the correct password, the following messages appear:

```
Press <RETURN> to modify configuration, any other key to abort.
```

Press the RETURN key when you see this prompt. The server displays the following message:

```
-705- Change leaves approximately nnnnn bytes free.
```

Use the CHECK PARAMETER SERVER command to store parameters on all parameter servers. (You can verify that all parameter servers are "Current" by examining the SHOW SERVER PARAMETER SERVER display.) Then re-initialize the unit, so that the change takes effect. You can use the command:

```
Xyplex>> initialize delay 0
```

Specify Server Settings

You must specify a number of SERVER characteristics which allow the server to operate as an AppleTalk node. The following is a summary of these SERVER characteristics. Refer to the *Commands Reference Guide* Supplement for more information about these commands.

NOTE: In order for the changes, specified by the DEFINE SERVER commands listed below, to take effect, you must re-initialize the server after issuing the commands.

- `DEFINE SERVER ARAP NODE NAME "`*`node-name`*`"`

  Specifies the server's AppleTalk name. This is the name that will be displayed in the Remote Access Status window of the Macintosh computer, when a user connects to the server using Remote Access. The name can be up to 32 characters in length and may not contain the double-quote (") character. If you do not specify a node-name, the server will use the default ARAP node-name, which is the server-name specified by the SET/ DEFINE SERVER NAME command or, if one is not specified, a seven-character name in the form X*nnnnnn*, where *nnnnnn* represents the last 6 digits of the server Ethernet address. (For servers that operate with a parameter server that is a VAX/VMS node, the default name is the DECnet node name that has been assigned by the system manager of that node.)

- DEFINE SERVER ARAP DEFAULT ZONE "*zone-name*"

  Specifies the AppleTalk zone that the server will attempt to join when it is initialized. The zone name may be up to 32 characters in length and may not contain the double-quote (") character. The default is None (not "NONE" which would be a zone-name), which means that the server will join the default zone for the attached EtherTalk segment.

- DEFINE SERVER ARAP PASSWORD "*password-string*"

  Specifies the password that registered (non-guest) ARAP users must type when they connect using remote access. The password can be up to 8 characters in length and can not contain the double-quote (") character. The *password-string* is case sensitive. The default ARAP password is access. There is only one ARAP login password per server.

- DEFINE SERVER PACKET COUNT *packet-buffers*

  Valid values for *packet-buffers* are whole numbers in the range of 80 to 1088; the default is 80. The server allocates 1556 bytes of memory for each additional packet buffer.

  You can determine the current number of *packet-buffers* available by examining the "Packet Count" field on the SHOW SERVER CHARACTERISTICS display. The server may use up to 12 packet buffers for each port at which ARAP is enabled. Since this decreases the number of packet buffers available for other applications, you will probably need to increase the number when you enable ARAP. (For example, six ports configured for ARAP will use up nearly all of the available packet buffers when the server is configured to use the default value of 80.) It is recommended that you increase the number of *packet-buffers* available by 12 for each port configured for ARAP.

Specify PORT Settings

Configuring an ARAP Port for Modem Support

You must make sure to configure the proper modem-related characteristics and to use the correct cabling. "Information About Xyplex Cabling Methods" provides cabling details. "Port Settings" covers how to set up a port to support dial-in or dial-back capabilities.

AppleTalk Remote Access (ARAP) Notes

The following notes apply to the ARAP implementation:

- When there is no TFTP script server available on the network, Command Control Language (CCL) scripts and dial back scripts are unavailable.
- ARAP supports only one login password that is shared by all ARAP users. When Kerberos or SecurID authentication is performed, a username may be used that has an associated password and/or passcode.
- When Kerberos or SecurID authentication is not used, the server does not restrict access by user name. A user can login through Remote Access using any user name as long as the user specifies the correct server password. Specific user names are only used for locating a telephone number for dial back.
- To prevent AppleTalk “name collisions,” do not have more than one Remote Access Server with a given name on an AppleTalk network.

Enabling ARAP at Specific Ports

You must enable ARAP on individual ports. This can be done either by setting up the port to accept multiple protocols with APD (covered in "Automatic Protocol Detection"), or setting the port up so that only ARAP is used on it, using the following command:

```
DEFINE PORT port-list ARAP ENABLED
```

Specifying Optional ARAP Port Settings

You may also want to alter PORT characteristics which affect ARAP sessions. The following is a summary of these PORT characteristics. Refer to the *Commands Reference Guide* for more information.

- DEFINE PORT *port-list* ARAP ZONE ACCESS *value*

  You can permit or restrict remote users from having access to various AppleTalk zones with this command. The *value* can be ALL (the default) for access to all AppleTalk zones, NONE for access to no AppleTalk zones, LOCAL for access only to the zone that the server is in, or a single *zone-name*, for access to a specific AppleTalk zone in addition to the zone that the server is in. A *zone-name* can be up to 32 characters in length and must be enclosed in the double-quote (") character (you cannot use the double-quote character as part of the *zone-name*).

- DEFINE PORT *port-list* ARAP MAXIMUM CONNECT TIME UNLIMITED/*time*

  You can limit the amount of time that users can remain connected, or allow users to remain connected for an unlimited amount of time using this command. If you specify a *time* (in minutes), the Remote Access client will be disconnected after being connected for the specified amount of time. You can also specify UNLIMITED, which means that the user can remain connected for an indefinite amount of time. UNLIMITED is the default.

- SET PORT *port-list* ARAP TIME REMAINING UNLIMITED/NONE/*time*

  UNLIMITED means that users at the port can now remain connected for an indefinite amount of time. NONE means that users at this port will be disconnected immediately (i.e., they have no more time). A value for *time* means that users at the port can now remain connected only for the specified amount of time. Specify the amount of time in minutes. The user will be notified of the change.

- DEFINE PORT *port-list* ARAP GUEST LOGINS ENABLED/DISABLED

  Specifies whether or not users can login to the server via ARAP as a "Guest" user (no password is required to log in as a guest user), rather than as a "registered" user. ENABLED means that a user at the port can login as a guest user. DISABLED means that a user at the port can not login as a guest user and must be a registered user. This is the default.

Install CCL scripts

CCL Scripts are required at ports which use ARAP. “Using CCL Scripts” covers CCL scripts in more detail.

Using ARAP With Authentication and Dialback Features

Xyplex communications servers offer several security features that control access to ACCESS SERVER ports and access to devices on the network. You can use these features individually, or combine them to achieve different levels of network security. (The *Advanced Features Guide* describes these security features, and some of the issues you may want to consider before you implement them at your site.) This section summarizes the operation of Xyplex security features at ports which are configured for AppleTalk Remote Access connections.

The authentication steps that are performed are somewhat complex. The actual steps that the access server performs in any given situation depend on the manner in which the remote user attempts to login (as a "Guest" or a "Registered" user) and on the settings for a number of DEFINE/SET PORT and SERVER settings affect the behavior of the Xyplex access server. These settings include:

```
DEFINE/SET PORT ARAP GUEST LOGINS
DEFINE/SET PORT CCL NAME
DEFINE/SET PORT DIALBACK
DEFINE/SET PORT KERBEROS
DEFINE/SET PORT SECURID
```

```
DEFINE PORT USERNAME
DEFINE/SET SERVER ARAP PASSWORD
```

NOTE: If you are using ARAP with Kerberos or SecurID authentication, or with dialback scripts, the AppleTalk "registered" user name must be the same as the Kerberos and/or SecurID user name, and/or the name of the dialback script.

You should note that methods of controlling access to LAT or TCP/IP resources on the network do not apply at ports which are configured for AppleTalk Remote Access connections:

- LAT Authorized Groups
- Limited View (LAT)
- Service Passwords (LAT)
- Internet Security

A system administrator must make a number of decisions about the manner in which the communications server will authenticate user logins. Figure 20 shows the activities associated with the use of these security methods. The diagram explains the entire process that the server performs in sufficient detail to make these decisions. The general order in which the Xyplex unit performs authentication or security-related activities at these types of ports is as follows:

- Kerberos authentication
- SecurID authentication
- Remote Access login
- Dial-back script execution

To users at remote Macintosh computers, Remote Access login appears to be the first operation that is performed. For the user, this activity is actually under control of the CCL script. During the process of establishing the connection, the Macintosh computer passes various information (login name, authentication passwords, etc) to the Xyplex unit. Figure 20 does not depict the role of the remote Macintosh computer in these activities.

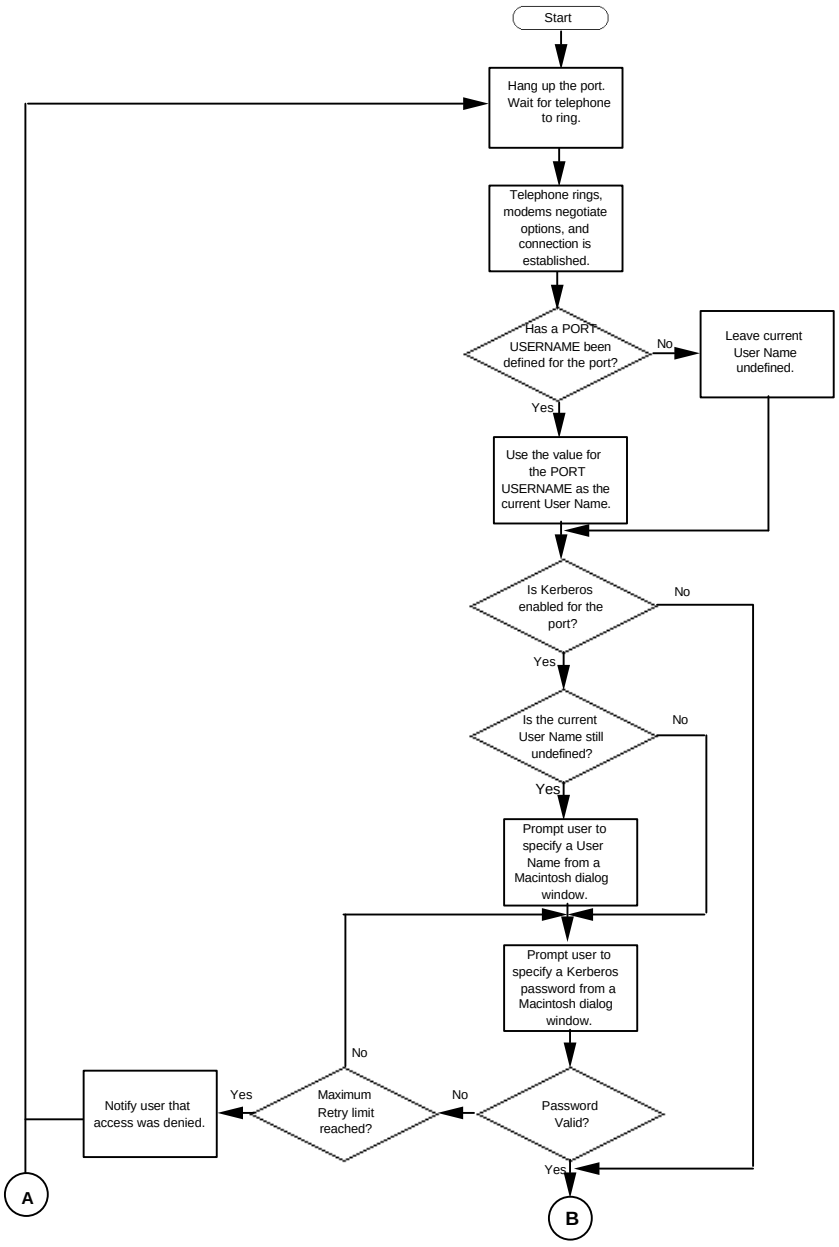


Figure 20, Part 1. Operation of Authentication and Security Methods

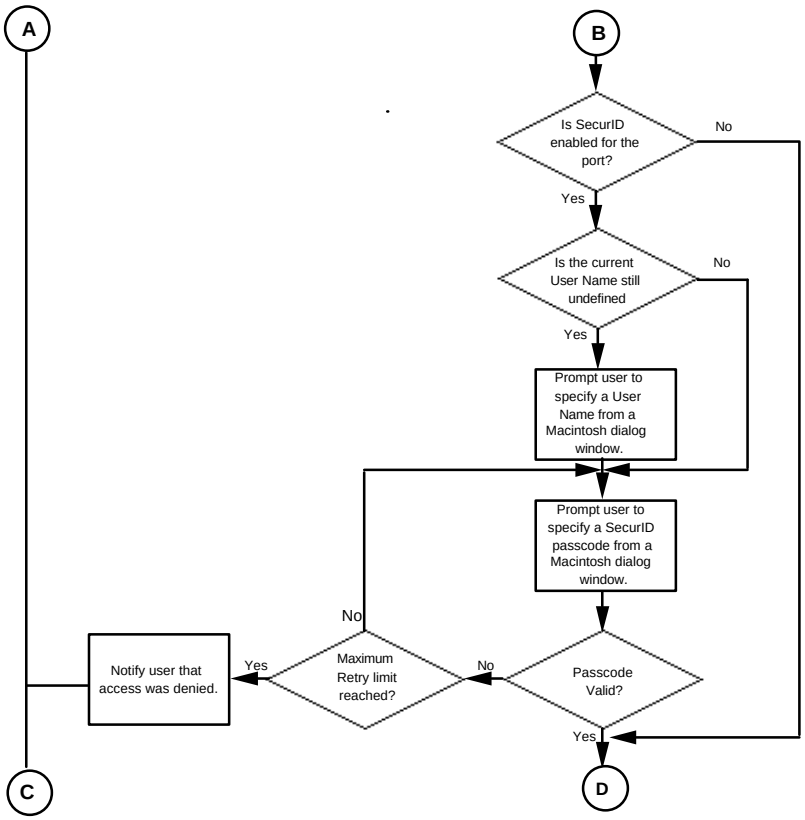


Figure 20, Part 2. Operation of Authentication and Security Methods.

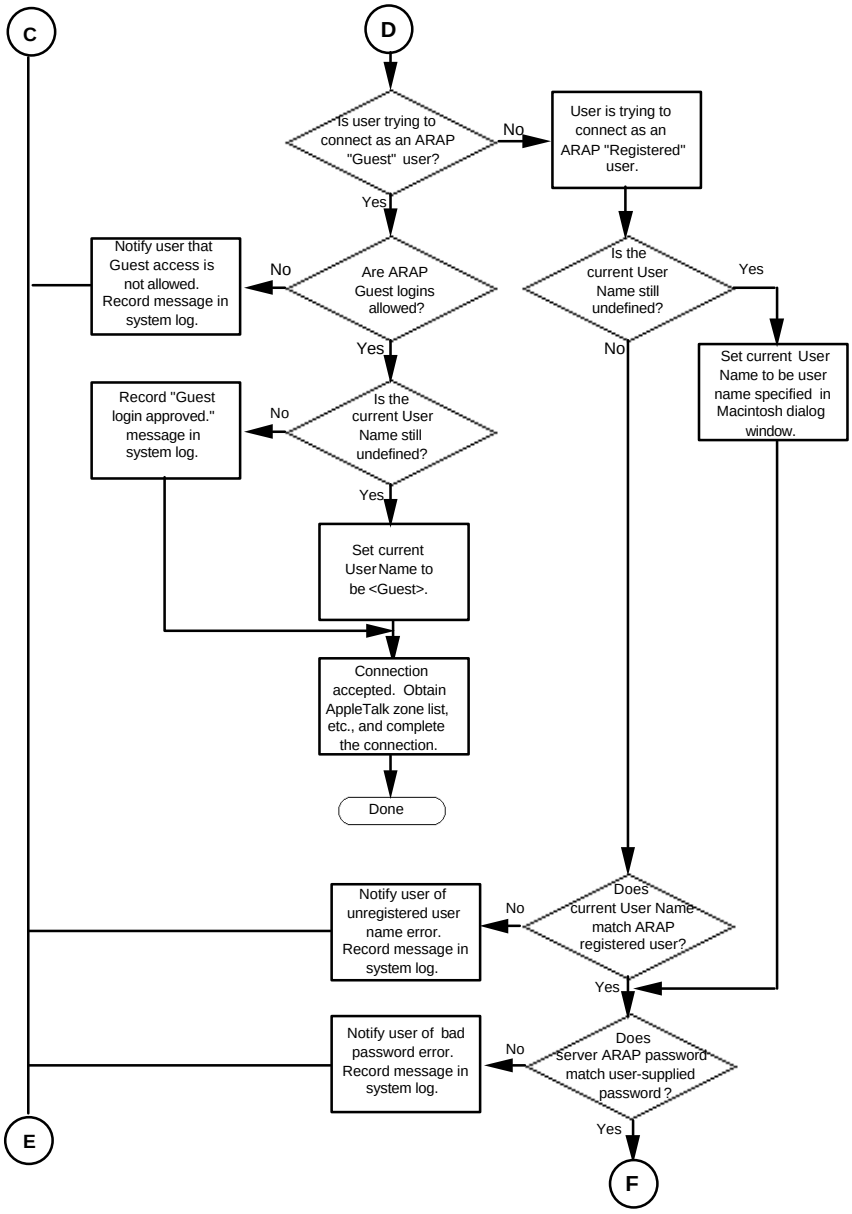


Figure 20, Part 3. Operation of Authentication and Security Methods.

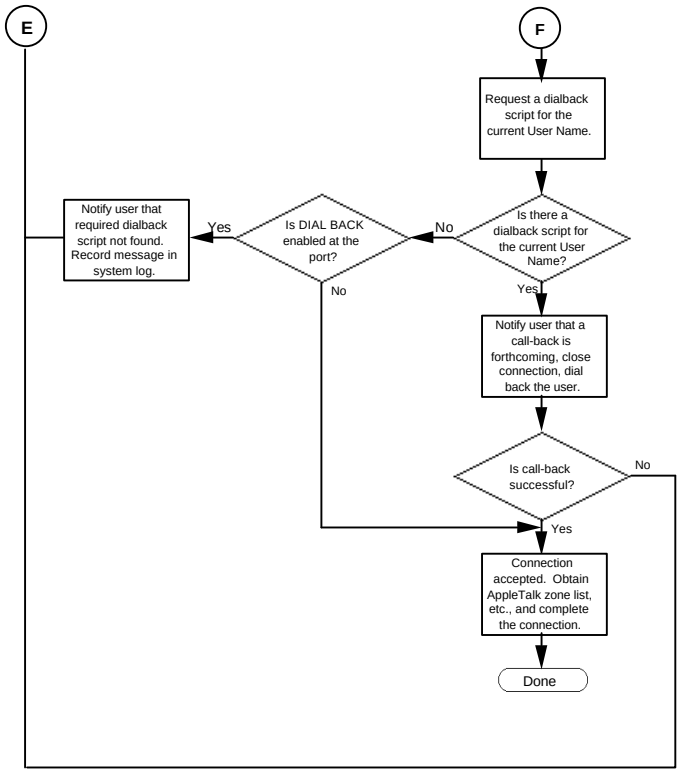


Figure 20, Part 4. Operation of Authentication and Security Methods.

Modifying Dialback Scripts for ARAP Ports

"Port Settings" describes the syntax for a dialback script in detail. However, there are differences between dialback scripts for use at ports which support AppleTalk Remote Access connections and those which do not. These differences are mainly due to the fact that ports which support AppleTalk Remote Access connections use CCL scripts to control modem and connection activity. For ports which support AppleTalk Remote Access connections, the main purpose of a dialback script is to pass a telephone number on to the CCL script, which then handles dialing and connection activity.

Observe the following guidelines for dialback scripts that are to be used at ports which support AppleTalk Remote Access connections:

- The first line in the script is always the following:

  ```
  #control_script
  ```

- The pound-sign character (#), when followed by the phrase ARAP_modem is used to specify to the CCL script the telephone number to be dialed. Do not include modem control commands, such as an ATDT command with the #ARAP_modem command. You can include commas or spaces in the telephone number, as permitted or required by your modem.

- Each line of a dialback script file can be up to 132 characters long. Each line must contain only one command. Each command must be on only one line.

- Within command scripts, a pound-sign character followed by a space or tab indicates a comment; the server ignores the remainder of the line.

- At ports which support AppleTalk Remote Access connections, the server ignores "#modem" commands. At ports which support AppleTalk Remote Access connections, only the #ARAP_modem command is processed.

At ports which are not configured to support AppleTalk Remote Access connections, the #ARAP_modem command is ignored. This allows you to configure one dialback script for a user, and allow that user to connect both to ports which support AppleTalk Remote Access connections and those which do not.

The following is an example of a dialback script that would be used only at ports which support AppleTalk Remote Access connections:

```
#control_script
# This is an ARAP-only dialback script.
#ARAP_modem 5551978
```

The following is an example of a dialback script that would be used only at ports which support AppleTalk Remote Access connections:

```
#control_script
# This is a generic dialback script.
#ARAP_modem 5551978
#modem atdt5551978
```

ARAP Planning Considerations

The Xyplex Remote Access implementation provides a cost-effective way to connect remote Macintosh computers to a home-office AppleTalk network. The number of ports that you can allocate for ARAP connections and the throughput that you can reasonably expect to achieve from this implementation depends on many factors. The factors include: CPU capacity and utilization, modem line speeds in use, and link utilization.

The probable maximum for the number of ports that can be assigned for ARAP connections can be determined using the following formula[1]:

$$\textit{number of ports} = \frac{\textit{unit-ARAP-capacity-rating}}{\textit{average-modem-line-speed} \times \textit{average-link-utilization}}$$

The actual number of ports that can be used with ARAP is reduced by factoring in overhead associated with other normal access server activity.

Unit-ARAP-capacity-rating is approximately 100,000 bits per second (bps) for a MAXserver 1620 or 1640 ACCESS SERVER or a Network 9000 ACCESS SERVER 720. For a MAXserver 800 or 1600 ACCESS SERVER, this value is 50,000 bps. These values were ascertained in actual tests.

---

1 The formula assumes that the Xyplex ACCESS SERVER is dedicated for making ARAP connections (i.e., no other optional features are enabled on the unit), that there is no Ethernet traffic to contend with, and that the processor will not be required to perform data compression activities. The theoretical maximum also depends upon the presence of "clean" telephone connections, so that the link does not need to retransmit garbled data. These factors should be taken into account when planning for "real-world" applications, however.

Average-modem-line-speed refers to the average speed at which the modems connected to the serial ports will operate. Most of the popular high-speed modems that are used by Macintosh computer owners operate at 14,400 bps. Typically, the lower-speed modems operate at 2,400 bps. For testing purposes, this number would be easy to calculate, since one would commonly use the same line speed for all modems.

Average-link-utilization is determined by examining how much traffic crosses a modem link for various types of applications. Tests performed at Xyplex indicate that interactive applications typically demand about 30% (.3) of a single link's available capacity. This type of traffic loading is typical of applications such as electronic mail, terminal emulation, text editing, etc. Applications such as large program and file transfers can demand about 50 to 60% (.5 to .6) of a single link's capacity.

Example

Assume a Network 9000 ACCESS SERVER 720, using high-speed modems operating at 14,400 bps, and users who are all using interactive-type applications. The theoretical maximum number in this example is:

$$\textit{number of ports} = \frac{100000 \text{ bps}}{14400 \text{ bps} \times .3} = 23.15$$

Experiments performed at Xyplex largely confirm these performance expectations for "real-world" applications. Factoring in the overhead associated with other normal access server activity, the results indicate that a MAXserver 1620 or 1640 ACCESS SERVER or a Network 9000 ACCESS SERVER 720 can comfortably handle traffic for 8-10 ports running simultaneously, using all high-speed modems (14,400 bps) and heavily-utilized links. These same units can comfortably handle 16-20 ports of interactive traffic. A MAXserver 800 or 1600 ACCESS SERVER was able to support roughly half the number of ports for the same types of traffic.

Diagnostic Cabling

"Information About Xyplex Cabling Methods" shows the wiring diagram of the 8-wire cabling that is needed to connect an access server serial port to a modem for ARAP applications. Figure 21 is a wiring diagram w i h shows the cabling that is needed to connect a server serial port directly to a Macintosh computer. You could use this configuration for debugging the Remote Access configuration on the Macintosh, or for familiarizing yourself with Remote Access operations. You can purchase modular cables and adaptors shown in the figure from Xyplex, or make your own cables based on the wiring diagram.

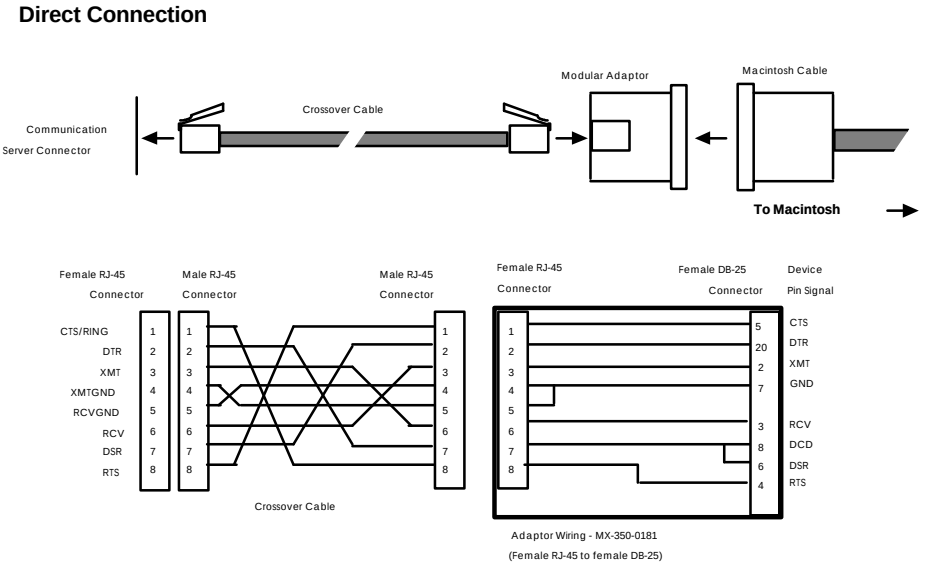


Figure 21. Modular Cables for Connecting a Macintosh Computer

Xyplex Support for the Xremote Protocol

This section describes the Xremote features that the access server supports, how to set up an XDM host and remote font servers, and how to configure an access server for Xremote support. This information is in the following sections:

- Starting Up the XDM Host
- Configuring the Communication Server for Xremote Support
- Notes on Memory Requirements for Xremote

Starting up the XDM Host

The X Display Manager (XDM) starts up the Xserver and the initial login window on an Xterminal or other display device which is either local or remote. The XDM prompts for a username and password, and manages the user's sessions. The access server requests management services from the XDM host using the X Display Manager Control Protocol (XDMCP) on behalf of the remote Xterminal.

Usually, you start XDM from the host system startup file `/etc/rc` . In a typical setup, XDM reads a configuration file when it starts. In this example, the default file is this:

```
/usr/lib/X11/xdm/xdm-config
```

Table 2 lists the typical default files that usually reside in the default directory `/usr/lib/X11/xdm` and are listed in `xdm-config.` These files can reside in any directory, however.

Table 2. Default Files

File	Purpose
`Xservers`	Contains a list of servers to start, which do not run XDMCP.
`xdm-errors`	Receives error output from the XDM. Examine this file when an Xterminal cannot connect to the XDM host.
`Xresources`	Contains default resources for the XDM login window.
`Xstartup`	Contains an optional program or script that runs after a user has entered a valid password.
`Xsession`	The default session manager program that starts up the user's Xwindow environment. It usually runs the `.xsession` file in a user's home directory, if this file exists, or a default session if it does not exist. The Xsession program is usually a shell script, and you can customize it for many tasks.
`Xreset`	An optional program that runs when a user logs out of the Xsession.
`Xdm-pid`	Contains the process id for XDM.

NOTE: The filenames on your host may be different.

Font files reside on the font server, which can be the XDM host or another host. The default font directory is usually `/usr/lib/X11/fonts`. Make sure that each font subdirectory includes a `fonts.dir` file and a `fonts.alias` file. Important font directories include `misc` and `100dpi` .

X Windows terminals that support the XDMCP protocol do not generally require special configuration on the XDM host. Because the access server supports this protocol, you need not configure the NCD Xterminals on the

XDM host.

You need not install or run NCD's `xinitremote` program or the `Xremote` program on the XDM host. You also need not install the file `.xinitremoterc` in the user's home directory. The access server code has the Xremote process embedded in it, so you need not install Xremote separately. The section Establishing an Xremote Session, later in this section, explains how the access server starts up the Xremote process when a user enables it at an access server port.

For more information about XDM host requirements, refer to these documents:

X Window System User's Guide Volume Three, by Valerie Quercia and Tim O'Reilly, O'Reilly and Associates, Inc.

MIT X Window system release notes and other documents are available through anonymous `ftp` on the Internet at `export.lcs.mit.edu` or `18.24.0.12`. When you reach this address, use `anonymous` as the username and password and go to the `/pub/R4` or `/pub/R5` directory.

For general information about Xremote, refer to the *NCDware 2.3 Xremote User's Manual*, from Network Computer Devices, part number 9300137.

Configuring the Communication Server for Xremote Support

The access server has certain parameters and port characteristics that support the Xremote protocol. In addition, you must define or set many general port characteristics in specific ways to support Xremote operation. Table 3, later in this section, lists these characteristics.

This section includes these topics

- Enabling the Xremote protocol on the Server
- Defining Remote Font Servers
- `tftp` Security on Font Servers

- Specifying Xremote Characteristics at Server Ports
- Establishing an Xremote Session
- Using a Script to Configure the Server for Xremote Support
- Enhancing Security for Xremote Users

Enabling the Xremote Protocol on the Server

This command enables the Xremote protocol in the permanent database of the access server :

```
DEFINE SERVER PROTOCOL XREMOTE ENABLED/DISABLED
```

```
Xyplex>> define server protocol xremote enabled
```

For Xremote to function properly, be sure to set all access server Internet characteristics for Internet protocol operation. See the *Advanced Features Guide* for more information about Internet characteristics.

Defining Remote Font Servers

To use fonts other than the ones available on your terminal, you must specify at least one remote font server, although you can specify two: a primary font server and a secondary font server. The XDM host can be one of the font servers, but you still have to specify it as a font server. Each time the Xterminal requests a font file, the access server requests the file from both the primary and the secondary font servers. It retrieves the file from the server that responds first.

Figure 22 shows a network with the XDM host defined as the primary font server and another host defined as the secondary font server. The NCD Xterminal is connected to the ACCESS SERVER 720 with a modem.

In Figure 22, the access server polls both the primary and the secondary font server. The primary font server, which is also the XDM host, responds first, so the access server loads the fonts from this host.

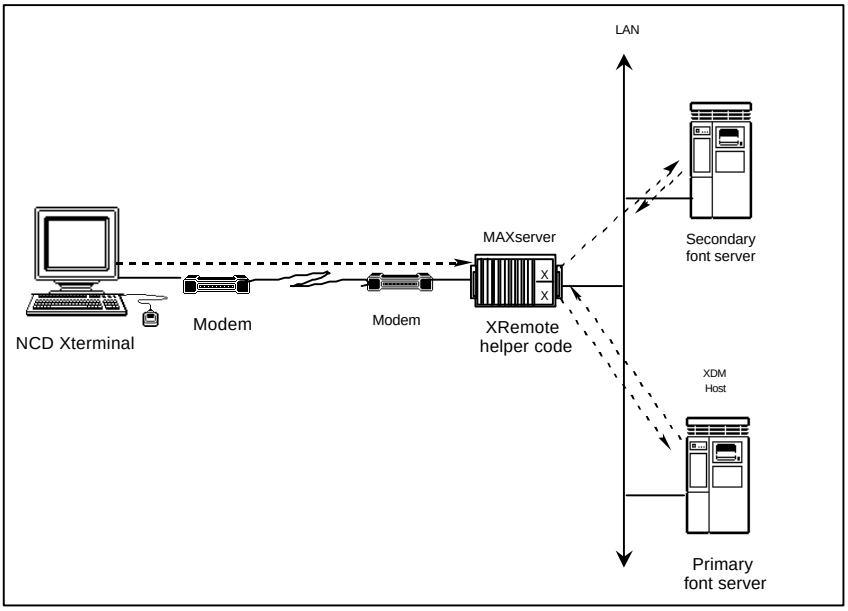


Figure 22. Font Servers

These commands specify the font servers:

DEFINE/SET SERVER XREMOTE PRIMARY FONT SERVER [*domain-name/internet-address/*NONE]

DEFINE/SET]SERVER XREMOTE SECONDARY FONT SERVER [*domain-name/internet-address/*NONE]

You can use either a domain name or an Internet address to specify a font server. The keyword NONE removes a previously specified domain name, and Internet address 0.0.0.0 removes a previously specified Internet address.

To use a remote font server once you establish an Xremote session, you must load the fonts from the server with the appropriate command from the Xterm window.

This is an example of a command which loads fonts from the `misc` directory:

```
xset fp+ /usr/lib/X11/fonts/misc
```

If a subsequent Xclient requires a font file within the `misc` directory, then the specific file is loaded through `tftp`.

You can add the `xset` command to the `Xsession` file in the XDM directory, or to the `.xsession` files in the user's home directory. Doing so loads the font lists for a user automatically at session initialization time.

Errors may occur during the font loading process. For example, the `tftp` file transfer may time out, `tftp` may not find the file, or `tftp` may not have access to the directory where the font files reside on the remote font server. Check the NCD Setup Menu Diagnostic Session for errors.

tftp Security on Font Servers

Because the access server uses `tftp` to transfer fonts from the font server to the access server and then across the NCD serial line, you need to ensure that `tftp` has access to the font file directories on the font server. In many X Windows environments, `tftp` runs with the secure option disabled. If the secure option is enabled, however, be sure that all of the font files are in subdirectories of the secure `tftp` home directory. Check the Internet configuration file on the UNIX font server to determine whether `tftp` runs with the secure option enabled or disabled. Refer to the man page for `tftpd` for information on how to set up `tftpd` on your UNIX system.

This example shows a SUN OS.4.1 system configured to run with `tftp` in secure mode. On this font server, the Internet configuration file `/etc/inetd.conf` has a command line that starts up the `tftp` server daemon, `tftpd`, with the secure option:

```
tftp dgram udp wait root /usr/etc/in.tftpd in.tftpd -s
/tftpboot
```

In this example, the `tftp` daemon starts with the secure `-s` option, and searches for files within `/tftpboot`, which is the default `tftp` home directory. When a font server such as this runs with the secure option, all font files must be in subdirectories of the `tftp` home directory, such as the font directory `/tftpboot/fonts/misc` . The directory `/tftpboot` is the default home directory for `tftp` files, but you can edit the file `inetd.conf` to change this.

This example shows the same SUN OS.4.1 system configured to run without `tftp` security.

```
tftp dgram udp wait root /usr/etc/in.tftpd in.tftpd
```

In this example, the `tftp` daemon can search for any file on the system. The `tftp` daemon should be able to find `/usr/lib/X11/fonts` on request from a remote `tftp` client.

Defining Xremote Characteristics at Server Ports

When an access server port requests an Xremote session, the access server software immediately begins searching for an XDM host. The software obtains the Internet address or domain name of the host either from the permanent database of the access server, from a name that the user enters at the Xyplex command interface, or through a broadcast request to the network. You can configure an access server port to search for the XDM host in any of these ways.

Use the DEFINE PORT XREMOTE ENABLED command to specify that when a user logs in to a port, the access server bypasses the Xyplex command interface and immediately begins searching for a predefined XDM host, or searches the network for an XDM host using the Internet Broadcast address. You specify an XDM host and query type with the DEFINE PORT XDM [HOST/QUERY] commands.

You can allow a user to specify an XDM host with the XCONNECT command from the Xyplex command interface after the user logs in to the port. The user specifies a domain name or an Internet address, and the access server software searches for that XDM host. The user can also enter the XCONNECT command without specifying an XDM host *if* you have used the DEFINE PORT XDM [HOST/QUERY] command to define a host, or the BROADCAST query type.

Examples

These examples show the different ways you can configure ports to search for an XDM host. When the access server software locates the host, and the host agrees to manage the session, the XDM establishes an initial master session with a log in window on the Xterminal. The next section, Establishing an Xremote Session, describes this process.

The first example defines the Xremote characteristics at ports 8-16. The query type is the default type (SPECIFIC), so the command line does not include a query type.

```
Xyplex>> define ports 8-16 xdm host 129.70.110.83
```

This example enables the Xremote process at ports 8-10.

```
Xyplex>> define ports 8-10 xremote enabled
```

These are DEFINE commands, so the network manager logs out these ports for them to take effect.

When users at ports 8-10 log in, the access server software automatically activates the Xremote process and searches for the XDM host at the Internet address 129.70.110.83. If the access server is successful, an XDM login window appears on the screen.

When users at ports 11-16 log in, the Xyplex access server prompt appears on the screen. These users must enter the XCONNECT command to establish an Xremote session. Users can provide the domain name or Internet address of an XDM host or simply enter the XCONNECT command to use the previously defined host and query type.

The following command causes the access server to search the permanent database for an XDM host or the broadcast query type for this port. In this example, the XDM host has been defined as 129.70.110.83.

```
Xyplex> xconnect
```

The following command specifies an XDM host at the address 130.63.110.79.

```
Xyplex> xconnect 130.63.110.79
```

Establishing an Xremote Session

When an access server port requests an Xremote session, the access server either sends XDMCP messages to the XDM host, or broadcasts XDMCP messages to the network if the query type is BROADCAST. If a host agrees to manage the display, the Xterminal automatically switches from ANSI emulation mode to Xterminal window mode. (The serial-session window disappears at this point.) The XDM establishes an initial master session, and the XDM login window appears after a few seconds. This uses two active access server sessions.

When you log in at the XDM login window, the XDM runs the Xsession file which usually executes the `.xsession` file in your home directory. This usually starts up additional windows and a window manager. You can also connect to other X Windows hosts and open windows from those hosts.

An Xclient process on a host running X Windows connects to the NCD Xserver on the NCD Xterminal through the access server. This accounts for one access server session. Each access server session corresponds to one Xclient process, and each window you open accounts for one Xclient process. The SHOW/MONITOR SERVER XREMOTE display shows the total number of active Xclients on the access server. If you want to observe Xclients or active access server sessions on a specific port, use the SHOW/MONITOR SESSIONS PORT *x* command.

If an XDM host refuses to manage the display, or the XDMCP request times out, the Xterminal remains in ANSI emulation mode. If the query type is SPECIFIC or you specified a host with the XCONNECT command, an error message appears on the screen. (See the section on Error Messages, at the end of these Release Notes.) If the query type is BROADCAST or INDIRECT, the access server searches for another XDMCP host. If it does not find one after repeated attempts, an error message appears on the display. The access server remains in ANSI terminal emulation mode, and you can enter other commands or log out of the port. You can define a different INTERNET XDM HOST for a specific query or use the XCONNECT command with a different XDM host, and attempt to reenable the Xremote session.

To disable the process, you log out of the port from the XDM host or hang up the modem. The session also becomes disabled if the XDM host refuses to manage a display or if the session times out.

Several port characteristics affect whether or not a user can successfully run an Xremote session. Table 3 lists these port characteristics and their recommended settings.

Table 3. Settings for Port Characteristics

Characteristic	Setting	Notes
MODEM CONTROL	ENABLED	This setting ensures proper port shutdown during disconnection. Be sure that other characteristics related to modems, such as DSRLOGOUT, DTRWAIT, and DIALBACK, are set appropriately for your modem.
ACCESS	DYNAMIC	This setting allows an interactive user login, followed by the posting of a passive network session, which Xremote requires.
SESSION LIMIT	16	The value must equal or exceed the maximum number of windows to be supported. Xyplex recommends 16 as the value for the SESSION LIMIT setting when running Xremote. Be sure that the session limit on the access server is equal to or greater than the sum of the session limits for each port which you plan to use. The maximum number of sessions on a server is either 128 or 255, depending on the type of unit you have.
TYPEAHEAD SIZE	1024	The value must be appropriate to the quantity of data being transferred, and should be twice the size of the INTERNET TCP WINDOW SIZE. While all allowable values are valid, Xyplex recommends the value 1024 when running Xremote.

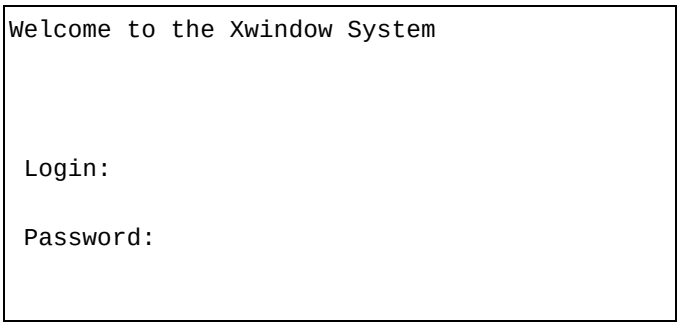
INTERNET TCP WINDOW SIZE	512	The value must be appropriate to the quantity of data being transferred. While all allowable values are valid, Xyplex recommends the value 512 when running Xremote.
TELNET REMOTE	6000 + *port-number*	The value of *port-number* must equal the physical port number on the access server.
SPEED	9600 or greater	To ensure the correct port speed, you can either set it with the SPEED characteristic, or set the AUTOBAUD characteristic to ENABLED so that when you enable Xremote, the current speed is in a valid range. The Port Characteristics display lists the current port speed in the "input speed" and "output speed" fields. NCD does not recommend using port speeds below 9600 baud, and Xyplex does not support port speeds below 9600 baud for Xremote.

If any of the MODEM CONTROL, ACCESS, TELNET REMOTE, or SPEED port characteristics are set incorrectly, you cannot enable Xremote, and an error message appears on the terminal indicating which characteristic is causing the error. If the SESSION LIMIT, TYPEAHEAD SIZE, or INTERNET TCP WINDOW SIZE characteristics are set incorrectly, you can still enable Xremote, but the session may not run properly. The access server does not generate an error message.

Example

In this example, a user enters the XCONNECT command with the domain name of an XDM host from the Xyplex command interface. This is a typical example. The messages and displays on your system may be different.

```
Xyplex> xconnect 234.179.70.155
```



Enter your login username and password. When you do this, the login window disappears, and the X Display Manager executes the `.xsession` file in your home directory, which typically contains one or more Xwindows and a window manager.

Logging Out of the X session

To log out, exit from the last process listed in the `.xsession` file, which is either the window manager or an Xwindow, or exit from each process separately. Be sure to close all open windows and the window manager before you exit from the X session, or they will remain open. These open processes can prevent you from reconnecting to the XDM host at a later time.

Using a Script to Configure the Server for Xremote Support

This section includes an access server script that specifies server parameters and port characteristics for Xremote support. This is a sample script, but you can modify it for the implementation at your site. It assumes that you have enabled Xremote on the access server. This script is installed on a UNIX host on the network. You associate a script with a port with the SET PORT SCRIPT command. See the *Advanced Features Guide* for more information about how to create and install access server scripts.

Comment lines begin with #. Comment lines that appear on the user's screen begin with #echo.

```
#control_script
#echo  This script initializes the ACCESS SERVER for running Xremote
#echo
#echo  Make sure to enable the Xremote protocol on the access server

#echo  before you execute the script with the DEFINE SERVER PROTOCOL
XREMOTE

#echo  ENABLED command.  You must enter a password to enable Xremote.  See
your

#echo  Xyplex sales representative if you need a password.
#
#  Enter privileged mode and specify server characteristics.
#  Customize font servers for the access server.
set priv system
define server xremote primary font server 123.123.123.123
set server xremote primary font server 123.123.123.123
#
define server xremote secondary font server 123.123.123.124
set server xremote secondary font server 123.123.123.124
#
#echo  This script initializes ports 1-4 for Xremote.
#echo  To view configuration changes, use these commands:
#echo  show server xremote, show port,
#echo  show port alternate characteristics,
#echo  show port telnet characteristics
#
define port 1-4 modem enabled
define port 1 telnet remote 6001
define port 2 telnet remote 6002
define port 3 telnet remote 6003
define port 4 telnet remote 6004
define port 1-4 typeahead size 1024
define port 1-4 internet tcp window size 512
define port 1-4 session limit 16
```

```
define port 1-4 access dynamic
#
#  Customize XDM hosts for each port.
define port 1-2 xdm host 123.123.123.123
define port 3 xdm host 123.123.123.124
define port 4 xdm host 123.123.123.125
#
#  Logout ports so that defined characteristics become working
characteristics.
logout port 1-4

#echo  Set port speeds

#  End of script
```

Enhancing Security for Xremote Users

The DEFINE PORT XREMOTE ENABLED command enhances security at access server ports because it causes the port to bypass the Xyplex command interface after a user logs in to a port. You can further enhance security on access server ports with Xremote enabled, or add security to ports without Xremote enabled, with these features: an access server password, the SecurID authentication system, the Kerberos security system, and dialback scripts. This section briefly describes these features as they apply to Xremote, but for more information about these features and access server security in general, see the Advanced Features Guide.

The Access Server Password

An access server password requires a user to enter a predefined password when the user attempts to log in to an access server port. The access server software does not begin its search for the XDM host until the user enters the correct password and logs in to the port. To use this feature, you enable the password requirement at specific ports and specify the access server login password. Use the following commands:

DEFINE/SET PORT *port-list* PASSWORD ENABLED/DISABLED

DEFINE/SET SERVER LOGIN PASSWORD *password*

The SecurID Authentication System

SecurID is a system of server software, client software, and accompanying SecurID cards from Security Dynamics Technologies, Inc[1]. The system is designed to secure a TCP/IP computer network, preventing unauthorized users from gaining access to resources on a TCP/IP network, but allowing authorized users to gain access easily to these resources.

Using SecurID authentication, the user must specify a SecurID personal identification number (PIN) and the password (PASSCODE) shown on a SecurID card in order to log on to the server. Once the user is logged on to the server, the user can connect to resources on the network. These resources can also be protected using authentication or other security mechanisms.

The Kerberos Security System

Kerberos is an Internet network authentication service that provides a central database of encrypted data, such as passwords, that access servers can use to verify login requests. A Kerberos system includes a Kerberos master host and one or more Kerberos server hosts. The master host maintains the database of encrypted data for a network organization called a realm. The master host provides data for the server hosts when clients in the realm query the server hosts for Kerberos verification. The network manager provides Kerberos passwords for access server users by entering them on the Kerberos master host.

With the Kerberos system running, the access server requires a user to enter a password before the user logs in to the port. You can provide unique passwords for each access server user. This can provide greater security than the server login password, which is the same for all access server users.

---

1 SecurID, PASSCODE, and PINPAD are trademarks of Security Dynamics Technologies, Inc.

NOTE: Kerberos requires compatible host software running at the TCP/IP host that is the Kerberos Master. Please contact your Xyplex sales representative if you want this software.

Login Scripts and Dialback Scripts

Login scripts are collections of access server commands that reside on a host computer. A user can execute a script, or a network manager can configure a port to execute a script automatically when a user logs in to the port. In an Xremote implementation, for example, the script could contain the XCONNECT command. When a user logs into the port associated with this script, the script bypasses the Xyplex command interface and automatically begins searching for the XDM host.

Dialback scripts are scripts that authenticate modem users. When a modem user dials in to the access server, the access server saves the username, disconnects the user, finds the dialback script for the user, and establishes the phone connection again with the dialback script. The user must then reenter the original username. If the user enters an incorrect name, the access server breaks the connection and returns the port to an idle state.

Notes on Memory Requirements for Xremote

Xremote is a configurable feature and uses a significant amount of free memory when you enable it. If you plan to use Xremote, be sure that any other configurable features that you have enabled are absolutely necessary. Otherwise, the configured image may not have enough memory to establish Xremote sessions, which can require almost 80 Kbytes to establish with only one window. See “Selecting Protocols and Features” for more information on configurable features.

How Xremote Can Affect Server Performance

Running several simultaneous Xremote sessions at different ports, each with multiple windows, may affect the performance of the access server. To improve performance somewhat, especially at line speeds above 9600 baud, you can increase the INTERNET TCP WINDOW SIZE port characteristic. If you do, be sure to increase the TYPEAHEAD SIZE so that it is twice the amount of the TCP window size.

If the access server is low on memory, you can reduce the Internet TCP window size and the typeahead size. This reduces the X Window memory requirements. See the SHOW/MONITOR SERVER ALTERNATE STATUS display for information on free memory utilization.

See Table 3, earlier in this section, for more information about the recommended values for port characteristics that can also affect the performance of the access server during Xremote sessions, including SESSION LIMIT, TYPEAHEAD SIZE and INTERNET TCP WINDOW SIZE.

Memory Requirements for Sessions and Windows

The following figures show the memory requirements to establish an initial Xremote session and each additional Xwindow. These values assume a port TYPEAHEAD SIZE set to the recommended amount of 1024 bytes and an INTERNET TCP WINDOW SIZE set to the recommended amount of 512 bytes:

Initial session requirement: Xremote requires at least 78,300 bytes of memory *per initial session*, which includes the initial X connection and the XDM login window.

Window requirements: Xremote requires at least 2,700 bytes of memory *per window* after you establish the initial session.

The access server automatically opens one login window with the initial X connection when you enable Xremote, and each of these uses one session on the Xyplex access server. After login, the login window disappears, and the session it used is terminated. The software then calls up a window manager, which uses one session. Assuming that the port has a session limit of 16, there can be a maximum of 14 working windows, one window manager, and one initial X connection.

If you established an Xremote session at a port and then opened four additional windows, the access server would require approximately 87 Kbytes:

76.5K for the initial X connection and 1 XDM login window (the login window disappears, but another session is used by the window manager).

10.5K for 4 additional active windows (2700 bytes each).

———

87K for a port with 4 working windows and a window manager.

Notes and Restrictions

V2.3.1 Xremote Server code Multiprotocol Communication Server Software V4.4 (and later) operates with revision V2.3.1 Xremote server code, but does not support all V2.3.1 features. In particular, the remote restart from the local window manager is not supported. The XDM host is not informed of the restart.

X11R5 X Windows code Multiprotocol Communication Server Software V4.4 (and later) operates with MIT X11R5 windows protocol, but does not support all new features.

X11R5 font service The font loading server provided in Multiprotocol Communication Server Software V4.4 (and later) uses `tftp` so it can work with hosts running X11R4. This font loading service is independent of the new X font server available with X11R5.

CCL Scripts

Command Control Language (CCL) scripts are files that contain commands which initialize a modem, configure communication between the modem and the device to which it is connected, and manage call-answering and call-hangup activities.

CCL scripts were originally designed to be used with AppleTalk Remote Access (ARAP). For ARAP connections, both the Macintosh computer and the access server require a CCL script. Typically, separate CCL scripts are used to initialize the remote Macintosh computer's modem, and the modem connected to the access server port.

CCL Notes (Using Modem-Based Compression)

The following notes apply to the CCL Notes:

ARAP connections cannot use modem-based compression. Compression must be done by the communication server. Typically, CCL scripts contain commands that prevent the modem from negotiating V.42 LAM-M error correction or V.42bis compression. To use modem-based V.42 LAM-M error correction or V.42bis compression for connections that are made using particular protocols (excluding AppleTalk Remote Access Protocol (ARAP)), use CCL scripts which permit this feature to be negotiated.

Modem-based MNP error correction is not supported on ports using CCL scripts.

CCLs are not supported on a port with RADIUS Authentication enabled.

While CCL scripts are required for ARAP connections, they can also be used to initialize the port and modem for other types of connections (PPP, SLIP, interactive, etc). There are some benefits to using CCL scripts even at ports where ARAP connections will not be used. First, one can use CCL scripts as an easy way of "programming" the modems. Second, one could think of a CCL as an "alternate" method of autobauding the serial port connected to the modem, since the CCL will determine the appropriate port speed and set it accordingly.

For non-ARAP connections (PPP, SLIP, interactive, etc), the server manager only needs to install a CCL script at a script server. A CCL script is not typically used by the remote device for these types of connections.

The topics contained in this section are:

- Types of CCL Scripts Available
- Communication Server Setup
- Script Server Setup
- Installing CCL scripts at Macintosh computers
- Modifying CCL scripts for Macintosh computers
- Modifying CCL scripts for Communication Servers

Available Script Types

Many modem vendors supply CCL scripts for use with their products. There are also public domain sources for these CCL scripts. Xyplex supplies CCL scripts for use with a variety of modems and makes them available publicly and as part of your software kit (depending on the type of kit you have ordered). Xyplex supplies CCL scripts in a UNIX tar archive and on a Macintosh formatted floppy diskette. CCL scripts supplied by Xyplex are listed in the *Software Kit Information* supplied with your software kit.

Xyplex also supplies CCL scripts which have been modified to:

- take advantage of Kerberos and SecurID authentication features when making ARAP connections.

  CCL scripts which contain these modifications are only installed at Macintosh computers. Xyplex-supplied CCL scripts which contain these modifications are designated with the prefix "`s.`" followed by the name of the generic CCL script for the modem. For example, for a Microcom 4232 Series Modem, the generic CCL script name is `Microcom_4232_series`. The modified version is named `s.Microcom_4232_series`.

  For "non-ARAP" connections (PPP, SLIP, interactive, etc), you can install the generic script used with the given modem at the script server. For these types of connections, no special CCL instructions are needed to support Kerberos or SecurID authentication.

- take advantage of V.42 LAPM error correction or V.42bis data compression for "non-ARAP" connections (PPP, SLIP, interactive, etc) at ports which use Automatic Protocol Detection (APD) to support both ARAP and non-ARAP connections.

  AppleTalk Remote Access (Version 1) required that modem connections be made without using any error correction or data compression methods. Therefore CCL scripts for modems which can use these features included modem instructions to refuse negotiation of these options. The modified CCL scripts (supplied by Xyplex) allows the server to accept ARAP connections without using error correction and data compression, while non-ARAP connections can negotiate these options. Scripts with these modifications only apply to access server ports.

  MNP error correction is not supported at ports which use CCL scripts.

Xyplex-supplied CCL scripts which contain these modifications are designated with the prefix "`l.`" followed by the name of the generic CCL script for the modem. For example, for a Microcom 4232 Series Modem, the generic CCL script name is `Microcom_4232_series`. The modified version is named `l.Microcom_4232_series`.

You do not need to use a CCL script with these modifications if APD is not enabled at the port, or if ARAP is not among the connection types (protocols) that will be accepted at a port where APD is enabled.

This section contains sections which describe how you can modify CCL scripts which are not among the Xyplex-supplied scripts.

Specify Script Server Settings

You must configure a script server for the access server. (Refer to the *Commands Reference Guide* and the *Advanced Features Guide* for information about script servers.) Use either of the following commands:

`DEFINE SERVER SCRIPT SERVER` *`domain-name`* "*`directory-path`*"

`DEFINE SERVER SCRIPT SERVER` *`internet-address`* "*`directory-path`*"

The *directory-path* specifies the name of the directory where script files are located. A valid *directory-path* can be a string up to 40 characters long. Separate the *directory-path* from the *internet-address* or *domain-name* with a space. Enclose the *directory-path* in quotation marks. For example:

```
Xyplex>> define server script server 140.179.224.10
"/tftpboot"
```

NOTES: CCL scripts are actually stored in the /CCL sub-directory of whatever directory you specify using the above command. This is described in the section titled "Install CCL Scripts." The complete directory path name where the CCL scripts are located would be: /tftpboot/CCL.

In order for this change to take effect, either re-initialize the

server or use a SET SERVER command as well as the DEFINE SERVER command.

Specify PORT Settings

Define which ports use which CCL files with a command of the format:

```
DEFINE PORT port-list CCL NAME "ccl-name"
```

ccl-name represents the file name which usually indicates the type of the modem connected to the port. The file is located in a directory at the script server. (Refer to the discussion about the location of CCL scripts which begins on the next page.)

Specify whether or not the modem speaker should be audible while it establishes a connection.

DEFINE PORT *port-list* CCL MODEM *value*

You can set *value* to be AUDIBLE or INAUDIBLE (the default).

Script Server Setup

Units download CCL scripts and dial-back scripts via the TFTP protocol. Typically, UNIX systems require that you locate all files that TFTP will transfer on the network in the TFTP "home directory" of your UNIX system or one of its sub-directories. Xyplex requires that CCL scripts be in a /CCL sub-directory of the TFTP home directory.

Most UNIX systems provide some mechanism that allows you to specify the TFTP home directory or use a default home directory. The default TFTP home directory varies from system to system. You will need to follow the configuration instructions for the TFTP daemon (tftpd) that are contained in the system documentation (e.g., MAN pages, etc) to determine how to locate the TFTP home directory.

For example, on Sun Workstations, the MAN page for tftpd says that the home directory is specified in the /etc/inetd.conf file, and that the factory

default home directory is /tftpboot. Therefore, you would examine the tftp entry in the /etc/inetd.conf file to see if the host is using the default home directory or a user-specified home directory.

You should be aware of any TFTP security mechanisms available on your UNIX system. Some TFTP implementations have no method of limiting the directories that TFTP has access to, which can present a security risk at some sites. Other implementations do have a method of limiting TFTP to certain directories, which means that you must place all files in a particular home directory, or in a sub-directory of the home directory. If the files are not located there, TFTP will not be able to find them. For example, SunOS (and some others) uses a TFTP daemon -s option (-s for secure) that restricts TFTP to having access to a particular directory and its sub-directories. Sun Workstations are normally configured with this option enabled. If you examine the /etc/inetd.conf file, you will see an entry similar to "-s /tftpboot" in the tftpd entry. Other vendors may use a different method. You should read the MAN page on tftp, tftpd, and inetd.conf to find out directory/security requirements on your UNIX system.

Installing CCL Scripts at Script Servers

For the access server, you specify a CCL for a given port (described in the section titled "Specify PORT Characteristics"). You also install the CCL script in a directory at the script server. For communications server ports, use the standard CCL script for the modem, or a CCL script which has been modified to support V.42 LAPM error correction and V.42bis compression. Do not use a CCL script that has been modified to include the extensions that are needed to support Kerberos and SecurID authentication (CCL scripts which contain these modifications are only used at a Macintosh computer).

Xyplex supplies a UNIX tar archive which contains CCL scripts for use with a variety of modems. These are listed in the *Software Kit Information* supplied with your software kit. Most modem vendors also supply a CCL script when you purchase a modem.

To install the CCL scripts that are supplied by Xyplex on a UNIX host, complete the following steps:

a. Change to the TFTP home directory, using a command of the form:

   `% cd` */tftp-home-directory*

   For example, on Sun Workstations using the default TFTP home directory, /tftpboot, use the command:

   ```
   % cd /tftpboot
   ```

b. Create a /CCL sub-directory. Use the command:

   ```
   % mkdir CCL
   ```

c. Change to the /CCL sub-directory. Use the command:

   ```
   % cd CCL
   ```

NOTE: In this example, the complete directory path name where the CCL scripts are located would be: /tftpboot/CCL.

d. Load the distribution tape onto a tape drive, then copy the desired UNIX tar archive from the distribution tape to the TFTP home directory using a "tar" command of the form:

   `% tar xfv /dev/`*(your tape drive name)* `ccl`

NOTE: For nine-track tapes, make sure that you use the correct *tape-drive-device-name* to match the format (QIC11 or QIC24) of the tape.

For example, on Sun Workstations to extract the UNIX tar archive named ccl from a QIC24 tape, use the command:

```
% tar xfv /dev/rst8 ccl
```

e. Extract the CCL files from the UNIX tar archive(s) copied from the distribution tape to the TFTP home directory (in step b, above), using a "tar" command of the form:

   `% tar xfv` *tar-archive-name*

   For example, on Sun Workstations to extract the CCL scripts contained in the UNIX tar archive named ccl, use the command:

   ```
   % tar xfv ccl
   ```

   You can delete the tar archive when you have completed extracting the CCL scripts from it. Later, you can delete unused CCL scripts if you need the space.

If you have a CCL script that is not included in the tar archive, you can simply copy the script file into the /CCL sub-directory.

NOTE: The section "Specify SERVER characteristics" (earlier in this section) covers the procedure to configure the unit to use the UNIX host as a script server.

For the remote Macintosh computer, you configure the CCL script by putting the CCL scripts into the Extensions folder (in the System Folder), and using the Remote Access Setup Control Panel. You access this window by running the Remote Access program and selecting the Remote Access Setup choice from the Setup menu.

Unless you are using Kerberos or SecurID authentication, you should use the CCL script provided by the modem vendor on the remote Macintosh computer. If you are using an authentication method, then you will need to use a CCL script that has been modified to include "extensions" which handle the prompts and messages needed to obtain passwords and information needed by the authentication method.

Xyplex supplies many CCL scripts for use with a variety of modems which have been modified appropriately. These are listed in the *Software Kit Information* supplied with your software kit. If you need a modified CCL script that is not included in this kit, you can create the script yourself, by following the instructions described in the section titled "Modifying CCL scripts for the Macintosh computer."

Modifying CCL Scripts

Modifying a CCL script to include the Xyplex Kerberos and/or SecurID Authentication extensions is very easy. Xyplex includes the necessary extensions in a file that is contained in the CCL UNIX tar archive, or on the Macintosh floppy diskette. You will find an example of the Xyplex extension and the text for a typical CCL at the end of this section. (The typical CCL shown does not contain the Xyplex extensions.) Refer to the documentation supplied by the APDA division of Apple Computer, Inc. for a description of the CCL script command language.

Basic Format of a CCL Script

CCL scripts consist of a number of sections, which are described here.

NOTE: The description shown here is a summary of the contents of a CCL script. The labels shown in this description are included only to illustrate basic CCL contents. In an actual script there will be number values for the labels, instead of the descriptive text strings shown below. For example, in strings such as:

```
@LABEL "answer"

IFANSWER (goto label "answer")
```

"answer" will actually be a number. While you are examining the CCL script, it is important that you examine the label numbers that are used. The Xyplex extensions were written using the labels 100 through 116, since most CCL scripts do not use labels in this range.

The first portion of a typical CCL script deals with originating or answering a call. The first commands in a CCL script are:

```
@ORIGINATE
@ANSWER
```

These commands are followed by a series of modem-specific commands which:

- configure the serial port (speed, bits per character, parity, stop bits, flow control)
- reset the modem to factory defaults
- configure modem operation for Remote Access (normal or direct mode, RTS/CTS flow control or no flow control, no error correction, no compression, echoing off, DTR handling, speaker on/off, etc)

The modem-specific commands are followed by the CCL command:

```
IFANSWER (goto label "answer")
```

which is followed by a modem specific "ATDT" command to dial the telephone number. This command instructs the script to jump to the portion of the script that deals with answering an incoming call. Otherwise, the script continues by executing the command on the next line and dialing the specified telephone number.

The dial command is followed by "match strings." An example match string is:

```
matchstr 5  14 "CONNECT 9600\13\10"
```

Match strings define all the possible responses that are expected from modem (such as CONNECT at a particular speed, no answer, busy, no carrier or dial tone, or some other error). Each match string also includes a label to which the program should go when the particular condition specified by the match string is met. The match strings are followed by a "matchread" command which tells the script to read data from the serial port and compare the data to the match strings. An example matchread command is:

```
MATCHREAD 700
JUMP "error-exit label"
```

Following the matchread command, there are labels and commands for each possible "CONNECT" response that is defined by a match string. Typically, these commands inform the user of the progress of the connection attempt. For example, when a connection is made, the user may be informed that the modem is "Communicating at *nnnn* bps." If the connection is not made, the user will be informed why the connection attempt failed. Optionally, the program can set the serial port speed. Each of these labels are followed by a jump to a common "success" label.

A success label will look like this:

```
@LABEL "success"
IFANSWER (goto label "exit-0")
PAUSE 30
@LABEL "exit-0"
EXIT 0
```

NOTE: Other commands can be included in this label, so it could take a moment to identify this label. Take note of its location. You will need to modify this portion of the CCL in order to use authentication methods with Remote Access.

The success label is followed by:

```
@LABEL "answer"
```

This portion of the CCL script deals with answering calls. The "answer" label is followed by modem-specific commands which:

- set the modem to autoanswer
- set up match strings to define all the possible responses that are expected from modem (such as CONNECT at a particular speed, no answer, busy, no carrier or dial tone, or some other error), similar to the match strings that are used when the CCL script is originating a call. These match strings usually jump to the same labels that are used for originating calls.

The match strings are followed by another "matchread" command which tells the script to read data from the serial port and compare the data to the match strings. An example of this matchread command is:

```
@LABEL "answer-2"
MATCHREAD 700
JUMP "answer-2"
```

This string is followed by labels and commands to exit when errors are encountered, and the command:

```
@HANGUP
```

which is followed by modem specific commands that handle hanging up the telephone.

Modifying a CCL Script for Macintosh Computers

To modify your CCL script to include Xyplex authentication extensions:

NOTE: The modifications described below only apply to CCL scripts which will run at the remote Macintosh computer.

1. Make the CCL script an editable text file. For example, you can use the ScriptSwitcher program (supplied on the Xyplex kit) or standard Macintosh programs such as the ResEdit program to change the file type to text. This permits you to edit the file using a program such as TeachText, Mockwrite, or any other text-editing program.

2. Examine the labels used in the CCL script. The Xyplex extensions were written using seventeen consecutive labels in the range of 100-116. Most CCL scripts do not use labels in this range. If any of the labels 100 through 116 are already being used by the script, either change the labels in the script or change the labels in the Xyplex extensions.

3. Locate the lines of the script that correspond to the "success" area of the script, comment out the "PAUSE" command (if any) by placing an exclamation point character at the beginning of the line. Then, add the command "JUMP 100" after the commented PAUSE command. Both of these are shown here:

   Before:

   ```
   @LABEL "success"
   IFANSWER (goto label "exit-0")
   PAUSE 30
   ```

```
@LABEL "exit-0"
EXIT 0
```

After

```
@LABEL "success"
IFANSWER (goto label "exit-0")
! PAUSE 30
JUMP 100
@LABEL "exit-0"
EXIT 0
```

(If you had to modify the label numbers in the extensions, change the 100 to whatever number you used as the first label in the extensions.)

NOTE: If the script does not follow the typical format described in the "Basic format of a CCL" section, you must do the following additional steps:

- Find all "EXIT 0" commands in the script that can be reached from the @ORIGINATE entry point.
- For each occurrence, replace the "EXIT 0" command with a "JUMP 100" (or whatever is the first label for the Xyplex extensions) command.

4. Insert the extensions at the end of the script. You can use the Macintosh "paste" function to do this.
5. Configure the Remote Access program on the Macintosh computer to use the modified CCL script. This procedure was described earlier in the section titled "Installing CCLs at Macintosh Computers.""
6. Change the CCL script back to an non-editable file. For example, you can use the ScriptSwitcher (supplied in the Xyplex kit) or ResEdit program and change the file type to mlts.

Modifying a CCL Script to Use Error Correction or Compression

To modify your CCL script to support V.42 LAPM error correction and V.42bis compression:

NOTES: The modifications described below only apply to CCL scripts which will be used at access server ports.

In order to perform this procedure, you will need to consult the owner's manual or programming manual for your modem.

1. Consult the documentation supplied with your modem to verify that the modem can perform V.42 LAPM error correction (independent of MNP error correction) or V.42bis compression.

2. Copy the CCL script to the script server or unpack the CCL tar archive (if you have not already done so).

3. Examine the CCL script using `more` or a text editor. Locate the lines of the script that deal with disabling V.42/V.42bis and/or MNP compression features. (Exact terminology varies from modem to modem). This text is usually located near the beginning of the CCL script. For example, you might see text similar to the following:

```
! Note: When the "\Nn" commands of the TP Serial are like those of the PP we
!       will need to change the \Nn setting to allow a V.42 connection !
       without a fallback to MNP4
!
! \n0 - disable v.42 autoreliable
! %c0 - turn off MNP5 compression
!
@LABEL 5
pause 5
matchstr 1 6 "OK\13\10"
write "AT\\n0%c0\13"
```

As the above example shows, the \n0 command disables negotiation of these features for this modem.

4. Edit the "AT" command to permit negotiation of the V.42 LAPM/V.42bis features. You will need to consult the documentation supplied with your modem to determine the exact command. For example, if modem documentation indicates that the command \n5 performs this function, you would change the AT command from:

```
write "AT\\n0%c0\13"
```

to

```
write "AT\\n5%c0\13"
```

5. Configure the access server to use the modified CCL script.

Example Xyplex CCL Extensions

```
------------------------------------------------------------------------
----
! Xooba - Xyplex Out Of Band Authentication; Copyright (C) 1993 Xyplex,
Inc.
!
! $RCSfile: xooba.ccl,v $ $Revision: 1.5 $ $Date: 1994/02/15 14:51:57 $
!
! Uses @LABELs 100-120 and MATCHSTRs 1-16.
!
@LABEL 100
ifanswer 116
note "Authenticating..." 2
!
! OOB Authentication occurs only on originated connections.
!
@LABEL 120
matchclr
matchstr  1 101 "Enter username> "
matchstr  2 102 "Enter user password>"
matchstr  3 103 "Access Denied"
matchstr  4 104 "Enter PASSCODE: "
matchstr  5 105 "PASSCODE Accepted"
matchstr  6 106 "Please Enter the Next Code from
Your Card:"
matchstr  7 107 "Enter    Your new PIN, containing"
matchstr  8 108 "Press  <Return> to generate a new
PIN"
matchstr  9 109 "PIN:          "
matchstr 10 110 "Wait for the code on your card to change,"
matchstr 11 111 "PIN rejected. Please try again."
matchstr 12 112 "Please re-enter new PIN:"
```

If you must change these label numbers, change them below also.

The numbers 101 through 116 in these match strings are also label numbers. If any of them must be changed, change them throughout the file (see the labels below).

```
matchstr 13 113 "PINs do not match. Please try again."
matchstr 14 114 "reserved1 for future use"
matchstr 15 115 "reserved2 for future use"
matchstr 16 116 "Xooba Done"
!
! Give the server 60 seconds to generate each message.
!
matchread 600
note "Server out of band authentication timed out." 3
exit -6002 "Server out of band authentication timed out!"
!
! Ask for the username and write it on the serial port.
!
@LABEL 101
ask 0 "Enter Username:"
write "^*\13"
jump 120
!
! Ask for the password and write it on the serial port.
!
@LABEL 102
ask 1 "Enter Password:"
write "^*\13"
jump 120
!
! Out Of Band Authentication failed.
!
note "Access denied." 3
jump 120
!
! Ask for the PASSCODE and write it on the serial port.
!
@LABEL 104
ask 1 "Enter PASSCODE:"
write "^*\13"
jump 120
!
! Record PASSCODE acceptance.
!
@LABEL 105
note "PASSCODE accepted." 3
jump 120
!
! Ask for the next card code.
!
@LABEL 106
ask 1 "Please Enter the Next Code from Your Card:"
write "^*\13"
jump 120
!
! Ask for a new PIN.
!
@LABEL 107
ask 1 "Enter new PIN (Ctrl-D cancels):"
write "^*\13"
jump 120
!
! Card requires a new PIN; leave it in New PIN mode.
```

The numbers 101 through 116 in these labels must match changes made above.

```
!
@LABEL 108
write "\04"
note "Card requires new PIN.  Contact system administrator." 3
jump 120
!
! Server attempting to supply new PIN anyway.
! This is beyond the capability of a CCL to handle.
!
@LABEL 109
note "Unable to assign new PIN.  Contact system administrator." 3
write "\04\13"
jump 120
!
! Wait for code to change; log in with new PIN.
!
@LABEL 110
ask 0 "Wait for card code to change."
jump 120
!
! PIN rejected; try again.
!
@LABEL 111
ask 0 "PIN rejected.  Type Return."
jump 120
!
! Re-enter new PIN.
!
@LABEL 112
ask 1 "Please re-enter new PIN:"
write "^*\13"
jump 120
!
! PINs do not match.
!
@LABEL 113
ask 0 "PINs do not match.  Type Return."
jump 120
!
! reserved1 for future use
!
@LABEL 114
jump 120
!
! reserved2 for future use
!
@LABEL 115
jump 120
!
! Xooba Done
!
@LABEL 116
exit 0
```

Labels continue to the end of the file.

Example of a Typical CCL Script

Global Village Teleport without Xyplex authentication mods:

```
! Xyplex CCL   $RCSfile: GV_TP_Serial_High_Speed,v $
!             $Revision: 1.1 $
!                 $Date: 1993/10/26 19:49:32 $
!
! Xyplex      load: y       Mac    secure: n
!           answer: n             answer: n
!        originate: n          originate: n
!
! 08/02/92 TelePort Gold draft
! 10/23/92 CTC Switched to new script written by RBH
! 10/26/92 CTC Fix for International calling(S7=60)
! 11/11/92 CTC Rewrote the Hang up section of the script to use AT\Y
! 12/07/92 CTC Fix for Shiva LanRover/L image 1.0 problem
!
@ORIGINATE
@ANSWER
!
! Talk to the modem at 19,200 bps.
!
serreset 19200, 0, 8, 1
!
! &f    - recall factory settings
! &d0   - Ignore DTR
! &k3   - Enable Hardware flow control
! w2    - Connect result code reports modem speed
! \q3   - Use RTS/CTS flow control in full-duplex mode
! \k0   - Enter command state but do not send break
! \j0   - Disable port rate adjust
! s7=60 - To allow for an international call
!
HSReset 0 1 0 0 0 0
settries 0
matchclr
@LABEL 1
matchstr 1 4 "OK\13\10"
write "AT&f&d0&k3\\k0\\j0w2\\q3s7=60\13"
matchread 30
inctries
iftries 2 59
! Modem is not responding, reset and send a break
SBreak
jump 1
!
! Next, Set up the configuration: Turn off auto answer and command echo.
!
! S0=0 - Don't answer calls
! E0   - Turn command echo off
!
@LABEL 4
matchclr
pause 5
matchstr 1 5 "OK\13\10"
write "ATS0=0E0\13"
```

```
matchread 30
jump 59
!
! Note: When the "\Nn" commands of the TP Serial are like those of the PP
!       we will need to change the \Nn setting to allow a V.42 connection
!       without a fallback to MNP4
!
!\n0 - disable v.42 autoreliable
! %c0 - turn off MNP5 compression
!
@LABEL 5
pause 5
matchstr 1 6 "OK\13\10"
write "AT\\n0%c0\13"
matchread 30
jump 59
!
! If speaker on flag is true, jump to label 8.  Else turn off the speaker
!
@LABEL 6
ifstr 2 8 "1"
pause 5
matchstr 1 8 "OK\13\10"
write "ATM0\13"
matchread 30
jump 59
!
! The modem is ready so enable answering, or originate a call
!
@LABEL 8
pause 5
ifANSWER 30
note "Dialing ^1" 3
write "ATDT^1\13"
!
@LABEL 9
matchstr 1  11 "CONNECT 1200\13\10"
matchstr 2  12 "CONNECT 2400\13\10"
matchstr 3  13 "CONNECT 4800\13\10"
matchstr 4  19 "CONNECT 7200\13\10"
matchstr 5  14 "CONNECT 9600\13\10"
matchstr 6  20 "CONNECT 12000\13\10"
matchstr 7  18 "CONNECT 14400\13\10"
matchstr 8  50 "NO CARRIER\13\10"
matchstr 9  50 "ERROR\13\10"
matchstr 10 52 "NO DIAL TONE\13\10"
matchstr 11 53 "BUSY\13\10"
matchstr 12 54 "NO ANSWER\13\10"
matchread 700
jump 59
! Notice that all we do for different connect speeds is issue a
! "CommunicatingAt" command.  Remember, we locked the interface speed
! to 19,200 bps so we don't want to reset the serial speed after we
connect.
! CommunicatingAt tells ARA what the actual line speed is so that it
! can set it's timers appropriately.  I guess your performance would be
! sub-optimal if you don't set this...
@LABEL 11
```

Start of area dealing with V.42 LAPM/V.42bis

"AT" command for V.42 LAPM/V.42bis

```
note "Communicating at 1200 bps." 2
CommunicatingAt 1200
jump 15                                  These "jump 15" commands point to the
!                                        "success" label.
!
@LABEL 12
note "Communicating at 2400 bps." 2
CommunicatingAt 2400
jump 15
!
@LABEL 13
note "Communicating at 4800 bps." 2
CommunicatingAt 4800
jump 15
!
@LABEL 19
note "Communicating at 7200 bps." 2
CommunicatingAt 7200
jump 15
!
@LABEL 14
note "Communicating at 9600 bps." 2
CommunicatingAt 9600
jump 15
!
@LABEL 20
note "Communicating at 12000 bps." 2
CommunicatingAt 12000
jump 15
!
@LABEL 18
note "Communicating at 14400 bps." 2
CommunicatingAt 14400
jump 15
!
!
@LABEL 15                                The "success" label.
HSReset 0 1 0 0 0 0
ifANSWER 16
pause 30                                 Comment out this line.  Add "jump 100"
@LABEL 16                                immediately after it.
exit 0
!
! @ANSWER
! Set up the modem to answer
!
@LABEL 30
write "ATS0=1\13"
matchstr 1 31 "OK\13\10"
matchread 30
jump 59
!
@LABEL 31
matchstr 1  32 "RING\13\10"
matchstr 2  11 "CONNECT 1200\13\10"
matchstr 3  12 "CONNECT 2400\13\10"
matchstr 4  13 "CONNECT 4800\13\10"
matchstr 5  19 "CONNECT 7200\13\10"
```

```
matchstr 6  14 "CONNECT 9600\13\10"
matchstr 7  20 "CONNECT 12000\13\10"
matchstr 8  18 "CONNECT 14400\13\10"
matchstr 9  50 "NO CARRIER\13\10"
matchstr 10 50 "ERROR\13\10"
matchstr 11 52 "NO DIAL TONE\13\10"
matchstr 12 53 "BUSY\13\10"
matchstr 13 54 "NO ANSWER\13\10"
matchread 700
jump 31
!
@LABEL 32
userhook 1
note "Answering phone..." 2
jump 31
!
! 50: error messages
!
@LABEL 50
exit -6021
!
@LABEL 52
exit -6020
!
@LABEL 53
exit -6022
!
@LABEL 54
exit -6023
!
@LABEL 59
exit -6019
!
! Hang up the modem
!
@HANGUP
@LABEL 60
settries 0
serreset 19200, 0, 8, 1
HSReset 0 1 0 0 0 0
@LABEL 61
!
! In order to hang up quickly, we go into command state and attempt a
! reliable connection. While we are attempting to re-establish
! the connection we hit return and cancel the attempt causing the modem to
! hang up.
!
SBreak
pause 20            ! allow time for modem to return from break
Flush               ! prevent disconnect garbage
write "\13"         ! start on a clean line
@LABEL 96
matchclr
matchstr 1 97 "OK\13\10"
write "AT\13"
matchread 30
Pause 30
inctries
```

```
iftries 3 59
jump 96
@LABEL 97
settries 0
matchclr
matchstr 1 62 "NO CARRIER\13\10"
Flush                ! prevent disconnect garbage
write "AT\\Y\13"     ! Attempt to re-establish connection
!pause 2             ! Causes problems with LanRover/L 1.0
@LABEL 98
write "ho\13ho\13"   ! Dreaded Christmas Abort re-connect
inctries
iftries 50 125
Jump 98
@LABEL 125
matchread 150      ! to hang up the modem
@LABEL 62
settries 0
pause 100
!
! recall the factory settings. (see note at top of script)
!
@LABEL 63
matchclr
matchstr 1 92 "OK\13\10"
write "ATZ\13"
matchread 30
inctries
iftries 3 59
jump 63
@LABEL 92
Settries 0
@LABEL 93
matchclr
matchstr 1 64 "OK\13\10"
write "AT&f&d0&k3\\k0\\j0w2\\q3s7=60\13"
matchread 30
inctries
iftries 3 59
jump 93
! Turn off auto answer.
! S0=0 - Don't try to answer the phone
!
@LABEL 64
pause 5
matchstr 1 65 "OK\13\10"
write "ATS0=0\13"
matchread 20
!
@LABEL 65
exit 0
```

Notice that there is a label 125 in this CCL script, and a label 98 just above (five lines up), but labels 100 to 116 are available for use. No need to edit Xyplex extensions in this case.

Add Xyplex extensions immediately after this line.

Modem and Flow Control

This section describes how modem control and flow control operate. You can use this information for trouble shooting.

This section, includes information about the following topics:

- Modem Control Signal Interaction
- Flow Control

The port characteristics that you must set to achieve the desired interaction between hardware and software depends on your modem control application. The following sections describe the interaction of hardware and software for the following modem control applications:

Dial-in modems which support RNG
Dial-in modems which do not support RNG

Dial in to remote access ports which do not support RNG (automated data collection applications)

Dial in/dial out modems which support RNG
Dial in/dial out modems which do not support RNG

Dial In Modems Which Support RNG

This is the standard configuration for dial in lines that support the RNG modem control signal. The server asserts the DTR modem control signal in response to the assertion of the RNG modem control signal. Figure 23 shows the modem states that the port enters as it observes or asserts various modem signals. In the diagram, circled items indicate port states and arrows indicate activity. This application has the following port settings:

PORT Characteristic	Setting
DSRLOGOUT	DISABLED
DTRWAIT	FORRING
MODEM CONTROL	ENABLED
ACCESS	LOCAL

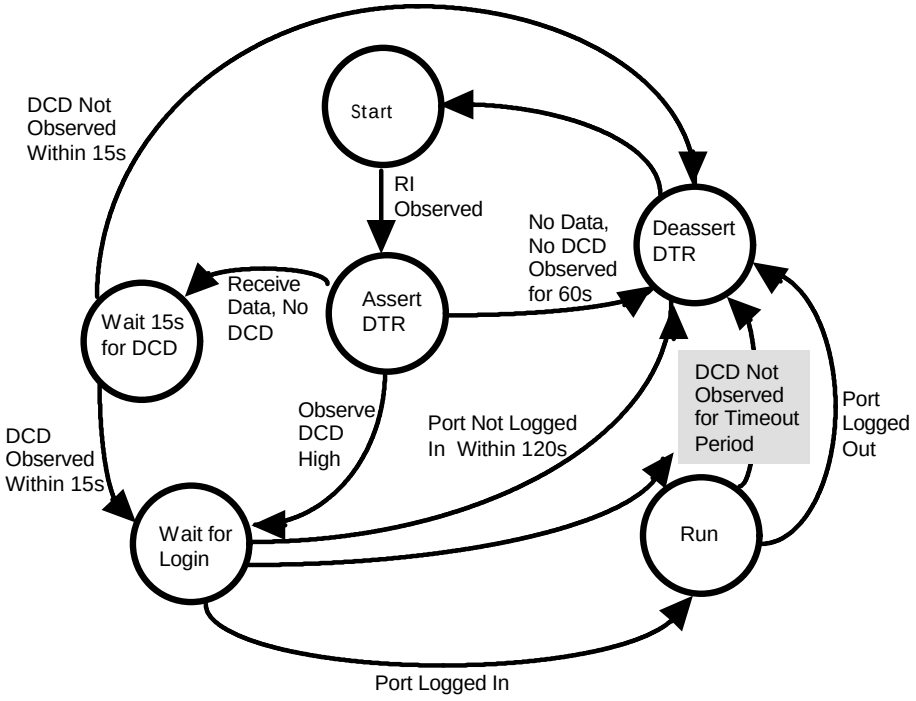


Figure 23. State Diagram for Dial In Modems Which Support RNG

The following list describes the sequence of signals for this application:

- Server asserts DTR upon seeing RNG.
- Server deasserts DTR if DCD is not asserted within 60 seconds of the assertion of DTR.
- Server permits data flow without waiting for the modem connected to the server port to assert the DCD signal (you can autobaud the port by pressing the RETURN key to select the port speed, when the PORT AUTOBAUD characteristic is set to ENABLED). However, the DCD signal must be asserted within 15 seconds after you start autobauding the port or logging in, or the session will be disconnected.
- Server deasserts DTR if user does not login within 120 seconds after the modem has asserted DCD.
- Server performs the disconnect sequence if DCD is deasserted for more than the period of time specified by the DCD TIMEOUT characteristic (default is 2 seconds).
- Server deasserts DTR if the user logs out of the port.

Dial In Modems Which Do Not Support RNG

This is the standard configuration for dial in lines that do not support the RNG modem control signal. The port asserts the DTR modem control signal by default. Figure 24 shows the modem states that the port enters as it observes or asserts various modem signals (in the diagram, circled items indicate port states, arrows indicate activity). This application has the following port settings:

Port Characteristic	**Setting**
DSRLOGOUT	DISABLED
DTRWAIT	DISABLED
MODEM CONTROL	ENABLED
ACCESS	LOCAL

The following list describes the sequence of signals for this application:

- Server normally asserts DTR
- Server permits data flow without waiting for the modem connected to the server port to assert the DCD signal (you can autobaud the port by pressing the RETURN key to select the port speed, when the PORT AUTOBAUD characteristic is set to ENABLED). However, the DCD signal must be asserted within 15 seconds after you start autobauding the port or logging in, or the session will be disconnected.
- Server deasserts DTR if user does not login within 120 seconds after the modem has asserted DCD.
- Server performs the disconnect sequence if DCD is deasserted for more than the period of time specified by the DCD TIMEOUT characteristic (default is 2 seconds).
- Server deasserts DTR if the user logs out of the port.

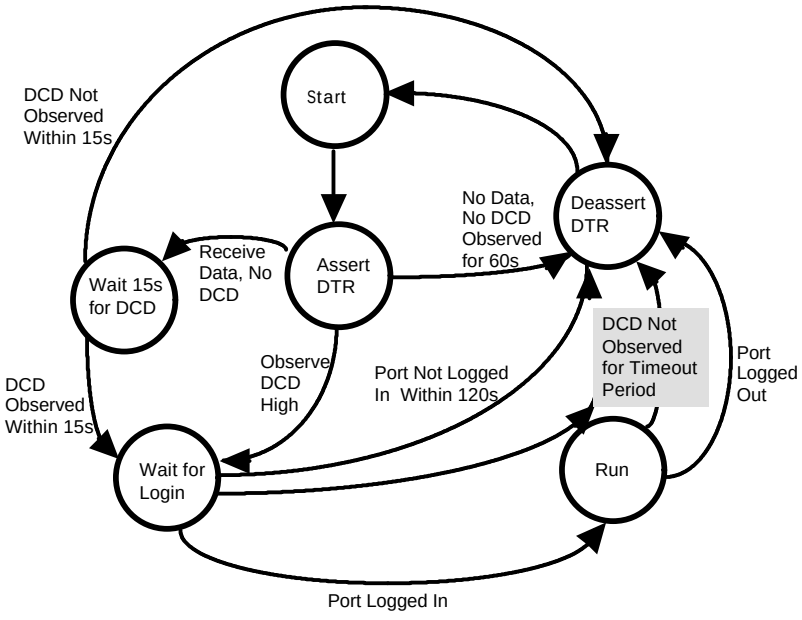


Figure 24. State Diagram for Dial In Modems Which Do Not Support RNG

Dial In to Remote Access Ports Which Do Not Support RNG

This is a non-standard configuration for dial in lines that do not support the RNG modem control signal. The server asserts the DTR modem control signal in response to the assertion of the DCD modem control signal. Dial-in to a port configured as REMOTE ACCESS is useful for automated data collection. This application has the following port settings:

Port Characteristic	Setting
DSRLOGOUT	DISABLED
DTRWAIT	ENABLED
MODEM CONTROL	ENABLED
ACCESS	REMOTE

The following list describes the sequence of signals for this application:

- Server asserts DTR.
- Server deasserts DTR if DCD is not asserted within 60 seconds of the assertion of DTR.
- Server permits data flow after the assertion of DCD.
- Server performs the disconnect sequence if DCD is deasserted for more than the period of time specified by the DCD TIMEOUT characteristic (default is 2 seconds).
- Server performs the disconnect sequence if the session is disrupted.

Dial Out Modems

This is the standard configuration for dial out lines. The server asserts the DTR modem control signal in response to the formation of a session connection to the service defined at the port. Figure 25 shows the modem states that the port enters as it observes or asserts various modem signals (in the diagram, circled items indicate port states, arrows indicate activity). This application has the following port settings:

Port Characteristic	**Setting**
DSRLOGOUT	DISABLED
DTRWAIT	ENABLED or FORCONNECTION
MODEM CONTROL	ENABLED
ACCESS	REMOTE

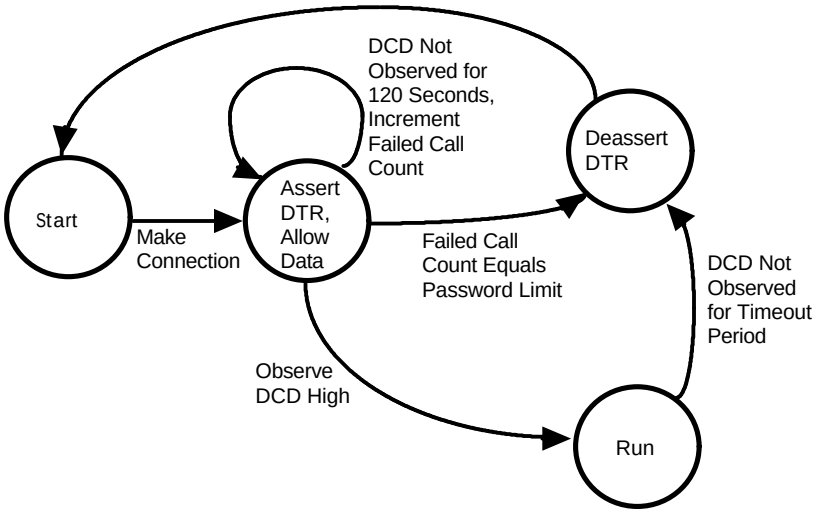


Figure 25. State Diagram for Dial Out Modems

The following list describes the sequence of signals for this application:

- Server asserts DTR when the remote connection is formed.
- Server permits data flow after the assertion of DTR.
- User may now dial out.
- Server deasserts DTR if DCD is not asserted within 120 seconds of the formation of the server connection. The server increments a failed call count. If the failed call count reaches the value set for the server PASSWORD LIMIT characteristic, the server logs out the port and deasserts DTR.
- Server performs the disconnect sequence if DCD is deasserted for more than the period of time specified by the DCD TIMEOUT characteristic (default is 2 seconds).
- Server performs the disconnect sequence if the session is disrupted.

Dial In/Dial Out Modems Which Support RNG

This is the standard configuration for dial in/out lines that support the RNG modem control signal. The server asserts the DTR signal in response to the assertion of the RNG modem control signal or the formation of a remote connection. Figure 26 shows the modem states that the port enters as it observes or asserts various modem signals (in the diagram, circled items indicate port states, arrows indicate activity). This application has the following port settings:

PORT Characteristic	Setting
DSRLOGOUT	DISABLED
DTRWAIT	ENABLED
MODEM CONTROL	ENABLED
ACCESS	DYNAMIC

The following list describes the sequencing of signals for this application:

- Server asserts DTR upon seeing RNG or the formation of a remote connection. If DTR was asserted in response to the formation of a remote connection, the line is assumed to be functioning as a dial out. If DTR was asserted in response to RNG, the line is assumed to be functioning as a dial in line.

- Server deasserts DTR if DCD is not asserted within 60 seconds of asserting DTR on a port functioning as a dial in line.

- Server permits data flow after the assertion of DTR for a dial out line. For a dial in line, the server permits data flow without waiting for the modem connected to the server port to assert the DCD signal (you can autobaud the port by pressing the RETURN key to select the port speed, when the PORT AUTOBAUD characteristic is set to ENABLED). However, the DCD signal must be asserted within 15 seconds after you start autobauding the port or logging in, or the session will be disconnected.

- Server deasserts DTR if user does not login within 120 seconds on a port functioning as a dial in line.

- Server deasserts DTR if DCD is not asserted within 120 seconds of the formation of the remote server connection. The server increments a failed call count. If the failed call count reaches the value set for the server PASSWORD LIMIT characteristic, the server performs the disconnect sequence.

- Server performs the disconnect sequence if DCD is deasserted for more than the period of time specified by the DCD TIMEOUT characteristic (default is 2 seconds).

- Server performs the disconnect sequence if the session is disrupted and the port is functioning as a dial out line.

- Server deasserts DTR if the user logs out of the port.

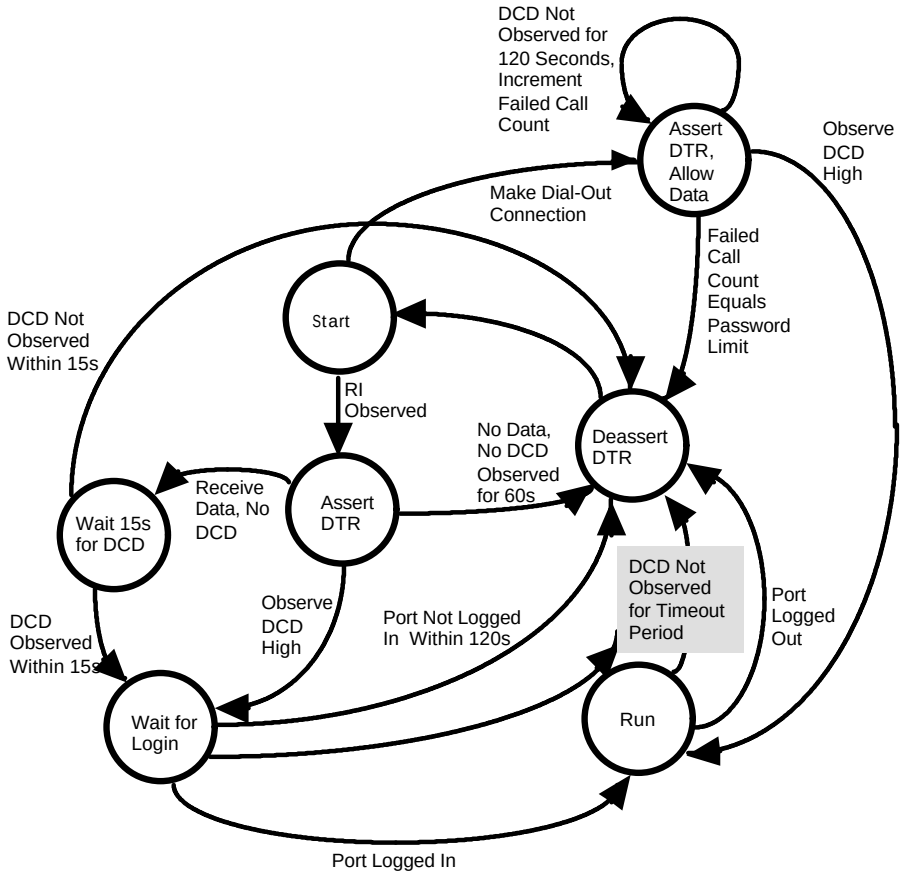


Figure 26. State Diagram for Dial In/Out Modems Which Support RNG

Dial In/Out Modems Which Do Not Support RNG

This is the standard configuration for dial in/out lines that do not support the RNG modem control signal. The port asserts the DTR modem control signal by default. Figure 27 shows the modem states that the port enters as it observes or asserts various modem signals . This application has the following port settings:

Port Characteristic	Setting
DSRLOGOUT	DISABLED
DTRWAIT	DISABLED
MODEM CONTROL	ENABLED
ACCESS	DYNAMIC

The following list describes the sequencing of signals for this application:

- Server normally asserts DTR.
- If a remote connection is formed to the server, the server assumes the port is functioning as a dial out line, else the port is assumed to be functioning as a dial in line.
- Server permits data flow after the formation of a remote connection for a dial out line. For a dial in line, the server permits data flow without waiting for the modem connected to the server port to assert the DCD signal (you can autobaud the port by pressing the RETURN key to select the port speed, when the PORT AUTOBAUD characteristic is set to ENABLED). However, the DCD signal must be asserted within 15 seconds after you start autobauding the port or logging in, or the session will be disconnected.
- Server deasserts DTR if user does not login within 120 seconds of the assertion of DCD.

- Server deasserts DTR if DCD is not asserted within 120 seconds of the formation of the remote server connection. The server increments a failed call count. If the failed call count reaches the value set for the server PASSWORD LIMIT characteristic, the server performs the disconnect sequence.
- Server performs the disconnect sequence if DCD is deasserted for more than the period of time specified by the DCD TIMEOUT characteristic (default is 2 seconds).
- Server performs the disconnect sequence if the session is disrupted and the port is functioning as a dial out line.
- Server deasserts DTR if the user logs out of the port.

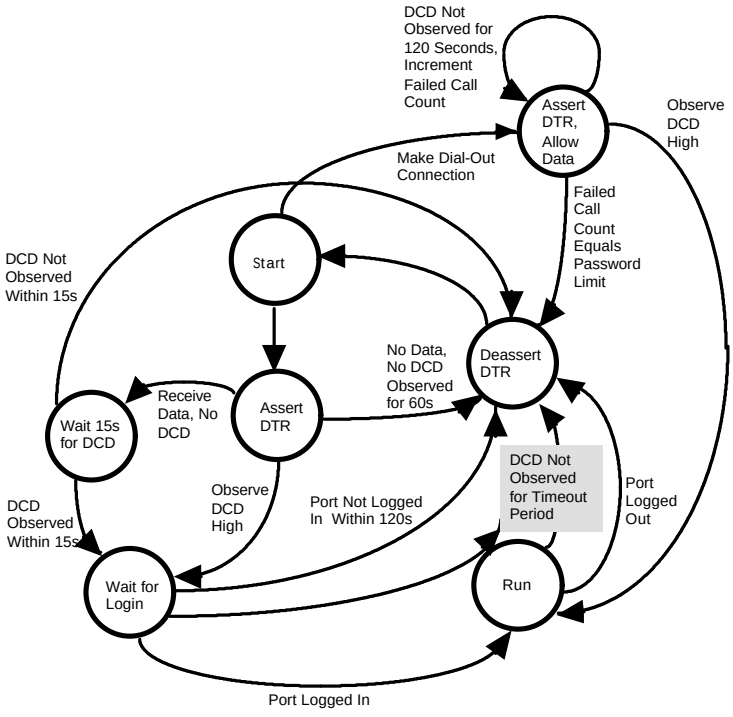


Figure 27. State Diagram for Dial In/Out Modems Which Do Not Support RNG

Flow Control

Xyplex server ports provide a flow control capability. This capability allows the port to inform the asynchronous device, such as a terminal, PC, or modem, to which it is attached to stop or start transmitting as required to prevent data loss. Similarly, a port will stop or start transmitting on request of the device to which it is attached. Xyplex servers support two modes of flow control: software flow control and hardware flow control.

Software Flow Control

Software flow control is implemented using ASCII XON and XOFF characters to start and stop transmission, respectively. The server port can both assert and observe these flow control characters. Using XON/XOFF flow control, the receiver that wants to stop a transmitter sends the transmitter an XOFF character. To start the transmitter, the receiver sends an XON character.

The use of XON/XOFF flow control can cause a problem with some data transfers, such as binary files, where the XON or XOFF characters may be data that needs to be passed to the connected partner on a session. In this situation, you can disable flow control, use hardware flow control, or set the session to Passall.

Hardware Flow Control

Hardware flow control is implemented in 6- and 7-wire interfaces with the DTR and DCD signal lines of a serial port. Using the DTR and DCD signal lines for hardware flow control, however, precludes their use as modem control signal lines.

Units with 8-wire interfaces also support hardware flow control using the RTS and CTS signal lines. These units support the concurrent use of hardware flow control and modem control.

When using hardware flow control, XON and XOFF characters can be treated as normal data, and hardware flow control is useful whenever XON or XOFF characters need to be interpreted as data. Hardware flow control is not implemented in all devices which can be attached to the port.

Set Up

The following PORT characteristics affect the flow control capability:

FLOW CONTROL
INPUT FLOW CONTROL
OUTPUT FLOW CONTROL
SESSION
TYPEAHEAD SIZE

The DEFINE/SET PORT FLOW CONTROL characteristic defines the default flow control mode of operation for the server port. The possible modes of operation are:

CTS	Hardware flow control
DISABLED	No flow control
DSR	Hardware flow control
ENABLED	XON/XOFF flow control
XON	XON/XOFF flow control

Although there are five choices, for 6- and 7-wire interfaces this characteristic effectively takes on one of three values:

1. No flow control
2. Software flow control using XON/XOFF
3. Hardware flow control (the PORT FLOW CONTROL keywords CTS and DSR are synonymous)

For 8-wire interfaces, there are separate CTS/RTS and DSR/DCD hardware flow control options, but only one of them can be used at a time.

Nonprivileged users and users at privileged ports can select among these flow control modes using a SET PORT or DEFINE PORT command.

While the port FLOW CONTROL characteristic defines the type of flow control that is used at the port, the assertion or observation of flow control can be disabled or enabled separately. The INPUT FLOW CONTROL characteristic specifies whether or not the port will be able to assert flow control when data is being transmitted by the device. The OUTPUT FLOW CONTROL characteristic specifies whether or not the port will observe flow control that is asserted by the attached device when the port is transmitting. Valid choices for both characteristics are ENABLED or DISABLED.

The user interface allows you to specify a distinct flow control mode for each session as well as for the command processor. The current flow control mode for each session is stored by the user interface, so that as you switch among sessions or the command processor, each session will resume using the correct flow control mode for that session. The mode is determined by the setting for the FLOW CONTROL characteristic.

Within each session, flow control is determined by a session mode which you specify using the SET SESSION command. Valid settings for the SESSION characteristic are INTERACTIVE, PASSALL or PASTHRU. The SET SESSION command can disable input and output software flow control on a per session basis. (The SET SESSION command does not disable flow control on ports using hardware flow control.) By setting the SESSION characteristic, you can disable the recognition of special characters such as the XON and XOFF characters, as well as other characters such as the forward switch character.

NOTE: Within a session, the recognition of special characters is dependant on the session mode, (INTERACTIVE, PASSALL or PASTHRU), not the FLOW CONTROL characteristic.

Flow Control Operation

The following operation description applies when either flow control mode is used, and the PORT INPUT FLOW CONTROL characteristic is set to ENABLED.

Each port has a buffer, called the type ahead buffer, which stores characters. As the port receives data, the port stores the data in the type ahead buffer. While the port is a source of data going into the type ahead buffer, the operation of the session takes data out of the type ahead buffer. When the type ahead buffer gets too full, the port tells the attached device to stop transmitting. The size of the type-ahead buffer can be set for each terminal port using the SET/DEFINE PORT TYPEAHEAD SIZE commands.

After a device has been told to stop transmitting, when the type ahead buffer later becomes empty, the device will be told to start transmitting again. This happens when the server empties the type ahead buffer (due to operation of the session). The frequency at which the server empties the type ahead buffer is determined by the setting of the DEFINE/SET SERVER CIRCUIT TIMER characteristic. The effect of the operation described above is that the port tells the device to stop transmitting when the type ahead buffer is nearly full, or to start up again when the buffer is empty.

The server determines when the type ahead buffer is "too full" by determining the size of the unfilled portion of the type ahead buffer. This size is compared to two thresholds. The first threshold is reached when there is space for 32 characters or the buffer is 80 percent full, whichever is greater. When a character is placed in the type ahead buffer after this threshold is reached, the port asserts flow control to inform the device to stop transmitting. The second threshold is reached when there is space for 16 characters or the buffer is 90 percent full, whichever is greater. When a character is placed in the type ahead buffer when this second threshold is reached, the port asserts flow control for each additional character received. Recognition of XON/XOFF flow control assertion or deassertion to the server can take up to 9 character times.

The effect of these thresholds is most notable when the port is using XON/XOFF flow control. In this case, an XOFF character is sent after the port receives a character when the first threshold is reached. If the buffer keeps filling, then each character received after the second threshold is reached will cause the port to send an XOFF character to the device for each character it receives. No XOFF is sent for characters that are received between the time when the first threshold is reached until the second threshold is reached.

The operation of flow control when the port is transmitting (when the PORT OUTPUT FLOW CONTROL characteristic is set to ENABLED) is considerably simpler than when the port is receiving. Regardless of the flow control method, whenever the attached device asserts flow control to tell the port to stop transmitting, the port stops. Whenever the attached device tells the port to start transmitting again, the port starts.

Finally, you can disable flow control. However, without an active flow control mechanism, data can be lost. This can occur with data flow either into or out of the port. Without flow control, the only mechanism that prevents data loss at the server port is the type ahead buffer, and the only mechanism that prevents data loss at the device is some similar type of buffer or the speed at which the device is able to absorb data.

Hardware Flow Control Operation Using The Modem Control Signal Lines

When the FLOW CONTROL characteristic is set to CTS or DSR (hardware flow control is selected), modem signals are used in place of XON and XOFF to control the flow of data. Only one type of hardware flow control can be used at a given port.

When using hardware flow control, the server observes the DCD or CTS modem control signal line and manipulates the DTR or RTS modem control signal lines. To stop data flow from the device to the port, the port deasserts the DTR or RTS signal. To start data flow, the port asserts DTR or RTS. Similarly, the port monitors DCD or DSR. When the port observes that the device has deasserted DCD or DSR, the port will stop transmitting. When the port detects that the device has asserted DCD or DSR, the port will start transmitting again.

INDEX