

Advanced Features Guide

For Xyplex Access Servers

Xyplex, Incorporated

295 Foster Street

Littleton, MA 01460

1-800-435-7997 (U.S.)

508-264-9903 (International)

support@xyplex.com (Internet)

www.xyplex.com (WWW)

420-0558F

The information in this document is subject to change without notice and should not be construed as a commitment by Xyplex. Xyplex reserves the right to revise this publication without obligation to provide notification of such revisions. While reasonable precautions have been taken, Xyplex assumes no responsibility for errors that may appear in this document.

No part of this publication may be copied or reproduced in any form or by any means without prior written consent of Xyplex, Inc.

AppleTalk is a registered trademark of Apple Computer, Inc.

DECnet and VAX are a registered trademarks of Digital Equipment Corporation.

Ethernet is a trademark of the Xerox corporation.

IPX is a registered trademark of Novell Corporation.

UNIX is a trademark of UNIX Systems, Inc.

Xyplex is a registered trademark of Xyplex, Inc.

Network 9000 and MAXserver are trademarks of Xyplex, Inc.

Copyright © 1996 by Xyplex, Inc.

Contents

Access Server Documentation Map	xi
---------------------------------------	----

Chapter 1 - Setting Up the Access Server

Getting Started	1
Configuring Specific Features	2
The Server's Parameter Databases	3
Define and Set Commands	3
Server Change Setting	3
Selecting Protocols and Features	5
Enabling UNIX Daemons	9

Chapter 2 - Managing Server Resources

How the Access Server Allocates Memory.....	12
Text Pool Area	12
Memory Management Guidelines	13
Select Only Features and Protocols Necessary At Your Site.....	13
Optimize Settings for the Enabled Features/Protocols.....	13
Upgrading Memory	14
Parameters that Directly Affect Memory Allocation.....	15
Local Services	16
LAT Services.....	16
Domain Names	17
Identifying Memory Problems	17
Error Messages.....	18
Server Displays.....	18
Adjusting Parameters	19
Server Node Limit	20
Server Queue Limit	21
Server Session Limit.....	22
Server Textpool Size	22
Server Packet Buffers.....	23
Parameter Server Limit	24
Port Typeahead Size	24
Port Internet TCP Window Size.....	26
Helpful Displays	27

Show/Monitor Server Counters	27
Show/Monitor Server Status	27
Show/Monitor Server Alternate Status	29

Chapter 3 - Using TCP/IP Features

Defining Basic Internet Settings	32
IP Address and Subnet Mask	32
Domain Name Server Support	33
Internet Broadcast Address.....	34
Internet Primary and Secondary Gateways.....	34
Show/List/Monitor Server Internet Characteristics.....	35
Configuring IP Routes	36
Host and Network Routes.....	37
Dynamic Routing	38
Static Routing	38
Define/Set Server Internet Route	39
Clear/Purge Server Internet Route	40
Show/List/Monitor Server Internet Route.....	40
IP Traffic Filtering.....	41
Enabling IP Filtering.....	42
IP Traffic Filter Criteria.....	42
Traffic Filter Commands	44
IPX Traffic Filters.....	49
Traffic Filter Criteria.....	49
Determining the Most Specific Filter	50
Traffic Filter Commands	51
IPX-RIP Import/Export Filters	54
When There Are Multiple Matching Filters.....	54
Defining RIP Filters	54
SAP Import/Export Filters.....	57
SAP Filter Criteria.....	57
When There Are Multiple Matching Filters.....	58
Defining SAP Filters.....	58
Configuring Rotary Connections	62
Configuring the Rotary.....	64
Domain Name Storage.....	66
Clear/Purge Server Internet Rotary.....	66
Show/List/Monitor Server Internet Rotary	67

Show/List Domain	67
Clear/Purge Domain.....	67
Configuring RLOGIN Support	68
Considerations	69
Associated Commands	69
Network Management	70
Using SNMP	70
Obtaining/Importing the Supported MIBs.....	73
Defining a Trap Client.....	74
Assigning SNMP Security Information (Optional)	74
Miscellaneous SNMP Settings	77
Using Telnet to Access the Console Port	78
Telnet Console Command.....	79
Define/Set Server Console Logout.....	80
Loading through Internet Protocols	81
Loading Images and Parameter Files	83
Configuring Load Protocols	84
Directed TFTP	84
Eliminating TFTP Broadcasts.....	85
Saving Parameters in Permanent Database.....	85
Dump Transmission	86
Define/Set Parameter Server	86
Define/Set Server Parameter Server Check.....	86
Show Server Status Display.....	87
Using the Server as a Domain Name Server	87
Domain Name Resolution.....	87
Obtaining/Storing Domain Names	88
Domain Name Time-to-Live TTL	89
Define/Set Domain.....	89
Show/List Domain	90
Clear/Purge Domain.....	90
Using IP Reassembly	91
Using TCP Resequencing	92
Translation Tables.....	94

Chapter 4 - Setting Up TN3270 Terminals

Enabling the Tn3720 Protocol	94
Enabling Extended Attributes.....	95
Defining TN3270 Devices	96
Creating a New Device Type	97
Using the TN3270 Command	97
Defining a TN3270 TERMINALTYPE	98
Defining the TN3278TYPE.....	99
Modifying the Keymap	99
IBM 3270 Display Station Functions for TN3270 Keymap.....	101
Modifying the Screenmap.....	103
Assigning a TN3270 Device to a Port.....	108
Other Commands that Affect TN3270 Devices	109
Defining TN3270 Translation Tables	110
Creating a New Translation Table	110
Entering New Values Into the Table.....	111
Assigning a Translation Table to a Port.....	114
Other Commands that Affect Translation Tables	115
Using Alternate Keymaps.....	116
Defining the Alternate Keymaps.....	117
Error Codes.....	119
Using Line 25 as a Status Line.....	122
Change the Terminal Display.....	122
Select the 25th Line.....	122
Activate the 25th Line	123
Status Line Information	124
Local Printer Support	125

Chapter 5 - Setting Up UNIX Daemons

Enabling Daemons	129
Using the Finger Daemon (fingerd).....	129
Using the Route Daemon (routed)	133
Using the rwho Daemon (rwhod).....	135
Memory Requirements	137

Chapter 6 - Using Nested Menus

About the Nested Menu Feature	138
How the Server Obtains the Menu File	141
How a Port Obtains the Menus	141
Setting Up the Script Servers.....	142
Creating the Nested Menu File	144
%menu_file	145
%menu_start <i>n title</i>	145
%menu_entry <i>n entry-description command-string</i>	145
%menu <i>n</i>	146
%menu_wait	147
%menu_noprompt.....	147
%menu_end.....	147
%menu_prompt <i>prompt-text</i>	147
%menu_continue <i>prompt-text</i>	147
%menu_top <i>x text-string</i>	148
%menu_up <i>x text-string</i>	148
%menu_exit <i>x text-string</i>	148
%menu_logout <i>x text-string</i>	148
%menu_repaint <i>x text-string</i>	149
Using Comment Lines in the Menu File	149
General Guidelines	150
Debugging the Menu File	150
Configuring the Server to Support Nested Menus.....	151
Server Settings	151
Port Settings.....	154
Sample Nested Menu Files	156
Sample File 1	156
Sample File 2	160

Chapter 7 - Using Scripts

How the Script Feature Works.....	174
Setting Up the Script Server	175
Create the Script File	178
Writing the Script.....	179
The # Character.....	181
Directory Requirements	183

Setting Up the Access Server to Use Scripts	185
Define/Set Port Script Echo.....	186
Script, Define/Set Port Script	187
Clear/Purge Server Script Server	187
Show/List/Monitor Server Script Server	188
Script File Execution and Processing	189
Sample Scripts	192

Chapter 8 - Using the Accounting Feature

Enabling the Accounting Feature.....	196
Enabling the syslogd Daemon.....	197
Information In the Account Log	198
The Default Account Log	198
Memory Considerations.....	199
Sample Default Account Log.....	200
Enabling the Verbose Account Log.....	200
Verbose Priority Number and Log File Location	201
Clearing the Account Log	203
Associated Displays.....	204
Show/Monitor Server Accounting.....	204
Show/List/Monitor Server Characteristics.....	204
Show/List Unit and Show/List/Monitor Server Alternate Status	204

Chapter 9 - Maintaining Boot Records

Viewing Initialization Parameters.....	206
Status of MX-1620/1640 Initialization Records	207
Enabling and Disabling Protocols.....	208
Changing Protocols through Commands	209
CARD, NVS, XMOP, and MOP Protocols.....	210
CARD Protocol	210
NVS Protocol.....	210
XMOP and MOP Protocols	210
Changing the Software Filename	211
BOOTP and RARP Protocols	211
DTFTP Protocol	211
Resetting Parameters to Defaults.....	214

Chapter 10 - How the Server Obtains the Current Time

Obtaining the Time.....	217
-------------------------	-----

Appendix A - Compatibility Issues

Appendix B - DEC Software Install and Management Tools

Index

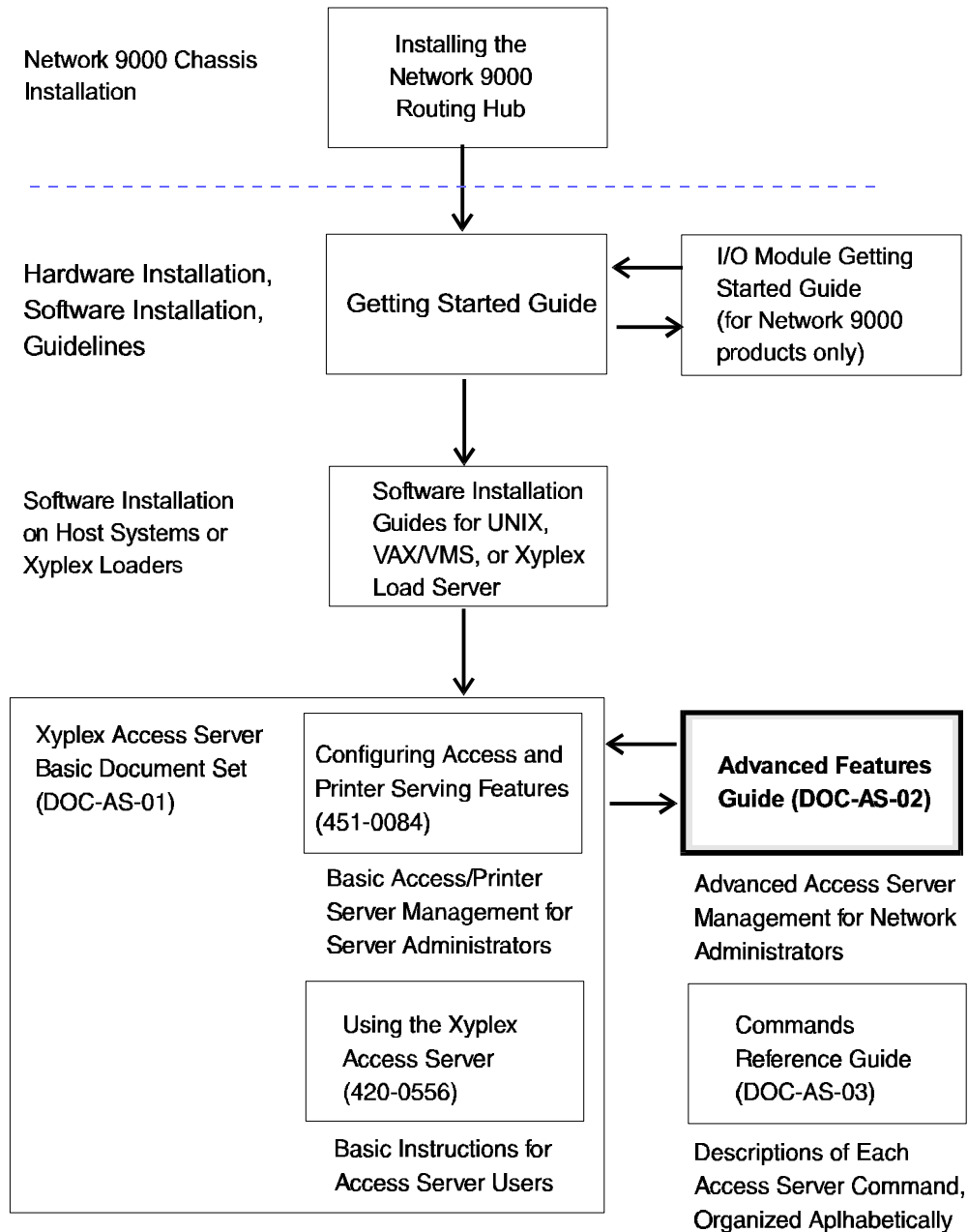
Figures

Figure 1 - Server Internet Characteristics Display	35
Figure 2 - IP Route	36
Figure 3 - Server Internet Routes Display.....	40
Figure 4 - IP Traffic Filter Example	46
Figure 5 - IPX Traffic Filter Example.....	52
Figure 6 - Rotary Connections.....	63
Figure 7 - Server Internet Rotary Display.....	67
Figure 8 - Connecting to Host via RLOGIN.....	68
Figure 9 - Network Management through SNMP	72
Figure 10 - Telnet Connections to Console Ports.....	78
Figure 11 - Loading through Internet Protocols.....	82
Figure 12 - Show Domain Display.....	90
Figure 13 - Server TN3270 Display.....	116
Figure 14 - Server TN3270 Translationtable Display.....	116
Figure 15 - Three-Level Menu Structure.....	139
Figure 16 - Sample Menu	140
Figure 17 - Sample Script Server Directory Structure	143
Figure 18 - Sample Script Server Directory Structure	177
Figure 19 - Server Script Server Display	188
Figure 20 - Sample Accounting Display.....	200
Figure 21 - Verbose Account Log.....	201
Figure 22 - Server Loaddump Primary Characteristics - MX-1620.....	206
Figure 24 - Show Server Display.....	218

Tables

Table 1 - Memory Usage For Features and Protocols	7
Table 2 - UNIX Daemons.....	9
Table 3 - Memory Usage for Various Session Types	14
Table 4 - Server Settings that Affect Memory Usage.....	15
Table 5 - Server Session Limit, Server Text Pool.....	15
Table 6 - Port Settings that Affect Memory Usage	16
Table 7 - Server Status Display Fields	27
Table 8 - Server Alternate Status Fields.....	29
Table 9 - Get/Set Client and Community Settings.....	75
Table 10 - IBM Display Station Functions.....	101
Table 11 - Screenmap Actions.....	104
Table 12 - Special Values for Escape Sequences	105
Table 13 - EBCDICTOASCII, USEENGLISH Translation Table.....	112
Table 14 - ASCII TO EBCDIC, USEENGLISH Translation Table.....	113
Table 15 - The %menu Commands	144
Table 16 - Priority Numbers for Messages from UNIX Daemons.....	202
Table 17 - Protocols for Loading and Dumping.....	208

Access Server Documentation Map



Preface

This manual describes the setup and management of advanced Access Server features. Its intended audience includes network, server, and UNIX and VAX host system managers.

Conventions

This manual uses the following conventions:

- Keys that you press are represented using left and right angle brackets (< and >). For example, the notation <CTRL> means that you press the CTRL key; <A> means that you press the A key; and <RETURN> means that you press the RETURN key.
- Unless otherwise specified, commands are executed when you press <RETURN>.
- The manual uses the following typographical conventions:

Monospace Typeface	indicates text displayed at a terminal (displays, messages, system responses, etc).
--------------------	---

<i>italics</i>	indicates variables in commands and procedures.
----------------	---

- The command prompt for non-privileged and secure users is:

`Xyplex>`

The command prompt for privileged users is:

`Xyplex>>`

This is the default user interface prompt; a server manager can specify a different prompt, so the prompt in use at your site might be different.

- The following typeface indicates user input in response to system prompts:

```
Xyplex> connect
```

- The following default user prompts are used (different prompts might be in use at your site):

VMS	\$
UNIX/ULTRIX	%
UNIX/Ultrix Superuser	#
DOS	C:\

If you have questions about this product...

At your convenience, please forward these to Xyplex at the following addresses:

Internet Mail: support@xyplex.com

United States Mail: Xyplex, Inc.
295 Foster Street
Littleton, MA 01460

Attn: Manager, Customer Support

If you have comments about this guide...

To help us in our effort to improve the quality, usefulness, and technical accuracy of the product documentation you receive, Xyplex is interested in any comments or suggestions that you have about this guide, or any technical corrections that you believe should be made. At your convenience, please forward these to Xyplex at the following addresses:

Internet Mail: documentation@xyplex.com

United States Mail: Xyplex, Inc.
295 Foster Street
Littleton, MA 01460

Attn: Manager, Technical Documentation

Chapter 1

Setting Up the Access Server

This chapter describes steps that you must take before setting up access server features. It covers the following topics:

- Getting started
- The server's parameter databases
- Selecting protocols and features
- Enabling UNIX daemons

Getting Started

Some basic server set-up steps are not covered in this guide; these include:

1. **Install server hardware and connect devices to the ports.** The *Hardware Installation* or *Getting Started Guide* supplied with your access server describes these tasks.
2. **Install software.** If the server will not be loading its software from a local memory card or diskette, you can install the software on a DEC VAX or Alpha, UNIX, or Xyplex load server. (A load server is a host or other device that stores the access server's software and sends it to the access server when requested.) The *Software Installation Guides* supplied with your software kit describes these tasks.

3. **Boot the unit.** The unit boots when you connect power to it. When the unit boots, it loads software from a local source, such as a memory card or diskette, or from a load server. You can reboot the unit anytime by entering the Initialize command (e.g., `INIT DELAY 0`), or by using the Reset switch. The *Hardware Installation* or *Getting Started Guide* supplied with your access server explains how to use the Reset switch.

Configuring Specific Features

Once you have completed the Getting Started steps, you can set up the access server for your specific needs. Some of the related tasks are covered in the rest of this chapter (Steps 4 and 5, which follow). Others are covered in later chapters or in other documents (*Configuring Printer Serving Features* or *Configuring Access Serving Features*).

4. **Specify which protocols, features, and daemons are to be used.** Note that some servers do not have enough memory to support all of the available features concurrently. Also, some features are disabled by default; you must enable them if you plan to use them.
5. **Reboot the access server after enabling features and protocols.** This allows all of the changes that you have made to take effect. Use the Initialize command to reboot the server (e.g., `INIT DELAY 0`).

The following start-up activities are covered in other chapters:

6. **Assign basic IP settings, if applicable.** (This step does not apply to LAT-only units.) The IP settings that you need to specify include:
 - The server's IP address and subnet mask

- Domain name server settings
- Other settings that enable the access server to communicate through IP gateways, if needed.

Refer to Chapter 4 ([page 32](#)) for more information about these settings.

7. Set up specific access server features. Refer to Chapters 4 through 13, as well as the documents *Configuring Printer Serving Features* or *Configuring Access Serving Features*, for instructions.

The Server's Parameter Databases

Access servers keep a *permanent database* of settings, which includes port and server settings, a list of services offered at the server, and in some cases destinations to which users can connect. When you reboot the server, the contents of the permanent database are copied into memory to serve as an “operational database.”

Define and Set Commands

Changes to the operational database are discarded when you reboot the server or a user logs off from a port. Use a Define command to change settings in the parameter file permanently. Use a Set command to change parameters temporarily.

Server Change Setting

Changes to settings that you make through Define commands take effect when you reboot the server. However, you can set the Server Change setting to Enabled, in which case Define commands take effect immediately.

Retaining Parameters when Loading New Software

To maintain your current parameters when you install a new software diskette or memory card, follow these steps:

1. Insert the new diskette or memory card into the access server.
2. Issue a Define command, such as `DEFINE PORT 1 TYPE ANSI`. This forces the server to update all parameters on the diskette or memory card.
3. Wait a minute, then issue the `SHOW PARAMETER SERVER` or `MONITOR PARAMETER SERVER` command. Check that the display shows “Storage State: Idle” and “Status: Current.” For a Network 9000 720 module, make sure that the CARD light is off.
4. Reboot the access server to load the new software version.

Selecting Protocols and Features

Xyplex Access Servers offer many features and network protocols — more than most sites require. The protocols and features that you use depends on your network environment and the amount of memory installed in the server. Chapter 3, [Managing Access Server Resources](#) (page 12), explains how to select the appropriate features and protocols for your site.

[Table 1](#), on page 7, lists various protocols and features and the amount of memory that each requires. In general, if you do not require a particular protocol or a feature, you should disable it to free up memory.

Use this command to enable or disable a protocol:

```
DEFINE SERVER PROTOCOL protocol-name [ENABLED]
                                     [DISABLED]
```

Valid *protocol* values are listed in the first column of [Table 1](#). If the server requires a password to enable a protocol, it will prompt you for it. Contact your Xyplex Sales Representative or Distributor for information about obtaining the password(s).

Use this command to enable or disable a feature:

```
DEFINE SERVER feature [ENABLED]
                      [DISABLED]
```

Valid *features* are listed in the first column of [Table 1](#).

Reboot the server after you have made all feature changes. When you enable a feature, the server sets all related server or port settings to their default values. When you disable a feature, the server changes all related server or port settings to reflect the fact that the feature is disabled.

Setting Up the Access Server

(For example, if you disable the Menu feature, the Port Menu setting is also set to Disabled for each port.)

When you enable or disable a feature, the server displays a message indicating approximately how much memory remains available:

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

IMPORTANT

Xyplex *strongly* recommends that you leave a minimum of 180 KB of memory available after enabling all needed features. If the server does not have enough memory to support a feature, it will display a message indicating approximately how much memory you must free up to enable the feature:

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

Tip - Save Current Parameters Before Enabling Features/Protocols

Because free memory might be fragmented (divided into pieces), the server might have enough total memory to load a feature — but not enough contiguous memory to run it. When this happens, the server appears to have enough memory — until you attempt to reboot it and the boot process fails.

At this point, you cannot reboot the server until you load a copy of an old parameter file that you have saved, or load the default parameter file. Rebooting with a saved parameter file saves much effort because you do not have to reset all previously configured settings.

Xyplex recommends that you save a copy of your original parameter file before you create a new one. If the new parameter file requires too much memory, the server will not boot.

Table 1 - Memory Usage For Features and Protocols

Feature Name	Memory Used in Kilobytes	Type	Default	Comments
LAT	55	Protocol	Enabled	
TN3270	55	Protocol	Disabled	
SNMP	80	Protocol	Disabled/ Enabled	Disabled by default for 1 MB load images; Enabled by default for multi-MB load images.
KERBEROS 4 KERBEROS 5	30 50	Feature Feature	Disabled Disabled	These features are mutually exclusive.
ACCOUNTING	0.5 to 90	Feature	0.5	Memory used depends on number of accounting entries.
MENU	7	Feature	Disabled	
MULTISESSIONS	12	Feature	Disabled	
INTERNET SECURITY	12	Feature	Disabled	
HELP	88 (2+ MB units) 30 (1 MB units)	Feature	Disabled	With V6.0 and later, full Help is disabled in multi-MB images.
XREMOTE	22	Protocol	Disabled	Requires more memory for each open session. Requires Multi-MB load image.
Manager Load	375	Feature	See Comment	Enabled by default on Terminal Server 720 and MAXserver 1620/1640 Terminal Servers if a memory card is present at initialization. Disabled on 1600/1450.

Setting Up the Access Server

Xprinter	25 plus 2 additional per port	Protocol	Disabled	Requires Multi-MB load image
PPP (Point-to- Point Protocol)	20 plus 6 additional per port		Disabled	Requires Multi-MB load image
ULI (UNIX [®] Operating System-Like Interface)	32	Feature	Enabled	Additional space required for command aliases (up to 512 bytes per port).
Nested Menus	40	Feature	Disabled	Additional memory required for the menu file.
ARAP	160 plus 43 additional per port	Protocol	Disabled	Requires Multi-MB image
IPX	84	Protocol	Disabled	Requires Multi-MB image
IP Filtering	10	Feature	Disabled	Requires Multi-MB image
IPX Filtering	25	Feature	Disabled	Requires Multi-MB image
APD	5	Feature	Disabled	Requires Multi-MB image
SecurID	15	Feature	Disabled	Requires Multi-MB image
RADIUS	11	Feature	Disabled	Requires Multi-MB image; at least 4 MB installed memory
lpd	8	Daemon	Disabled	Requires Multi-MB image

Enabling UNIX Daemons

Xyplex Access Servers support several UNIX daemons, which are system processes that run in background mode at network hosts. Daemons are commonly used in UNIX environments and TCP/IP networks to exchange information about network activity, or to manage resources such as printers and peripherals.

Use this command to enable UNIX daemons:

```
DEFINE SERVER DAEMON daemon-name [ENABLED]  
[DISABLED]
```

Valid *daemon-names* include fingerd, routed, rwhod, lpd, and syslogd. They are disabled by default. Table 2 lists the memory requirements for each daemon except syslogd. The memory allocation for this daemon is not dynamic. The syslogd daemon maintains an accounting log on the UNIX host (see “Accounting” in Table 1 on [page 7](#)).

Table 2 - UNIX Daemons

Daemon	Memory Used in Kilobytes
fingerd	12
routed	8
rwhod	6
lpd	16

Chapter 2

Managing Server Resources

An access server allocates (sets aside portions of) its memory to support activities such as:

- Storing the software load image and parameters
- Supporting enabled features and protocols
- Storing information about sessions, network destinations, and the connection queue
- Providing session resources, such as the type-ahead buffer, for users

Since each site's needs are different, the access server enables you to specify how the memory is allocated. This chapter explains how to do so, and covers these topics:

- How the access server allocates memory
- Strategies for managing memory
- Parameters that directly affect memory allocation
- Helpful displays

How the Access Server Allocates Memory

The server allocates memory when it boots. The server first loads its software image, then the parameter file. It then checks the list of configurable features and protocols in the parameter file, to determine which ones are enabled. The server then frees up memory that is not being used by the disabled features/protocols.

Next, the server allocates portions of memory for specific purposes, or to store specific types of data. The server is then ready to run.

In general, further memory allocations occur on an as-needed basis. For example, the server can allocate memory to store information about a LAT service, or to provide resources such as the typeahead buffer when a user wants to establish a new session. Similarly, the server frees up memory when it is no longer needed, or when instructed by the server manager.

Text Pool Area

The text pool area is a permanently allocated portion of memory, the size of which is fixed when the server boots. The server stores identification strings for nodes, LAT services, and domain names in this area. When the server boots, it immediately stores in the text pool space the identification strings for the local services and domain names that are currently stored in the permanent database.

The server fills the remaining text pool space as needed. (All other information about nodes and services is stored in a non-text pool portion of memory and is allocated when needed.)

Memory Management Guidelines

The goal of memory management is to balance the relationship among features and protocols, performance, and cost-effectiveness. A proper balance ensures that the server has enough resources for all users when needed.

Select Only Features and Protocols Necessary At Your Site

[Table 1](#) (page 7) and [Table 2](#) (page 9) list protocols, features, and UNIX daemons that you can enable, and the amount of memory that each requires. Chapter 2 explains how to enable/disable features, protocols, and daemons.

To free up memory, you should disable the protocols and features that you do not need. Some protocols or features require that another protocol also be enabled. For example, SNMP, TN3270, and Kerberos require that Telnet be enabled.

Help does not rely on other features or protocols.

Optimize Settings for the Enabled Features/Protocols

Server, port, and per-session settings can affect the amount of memory that a feature uses, and the performance of sessions that make use of the feature.

Typically, you do not need to change these settings after they are initially defined. Often, the default values are adequate. However, you might need to monitor and adjust some settings for better performance, and to ensure that there is enough memory to support all users' needs.

For example, Table 3 lists the amount of memory that various types of sessions use when the related Server and Port settings are set to the default values:

Table 3 - Memory Usage for Various Session Types

Session Type (Using default PORT values)	Memory Used Per Session (Bytes)
LAT session	1568, plus 1200 per virtual circuit
Telnet session	2320
TN3270 Model 2 session: no Extended Attributes	5104
TN3270 Model 2 session: with Extended Attributes	7008
TN3270 Model 5 session: no Extended Attributes	6720
TN3270 Model 5 session: with Extended Attributes	10352
Xremote Session	78300 for the initial X connection and XDM login window, plus 27000 for each additional window

Upgrading Memory

If you cannot enable every feature that you need and still have enough memory available for users' sessions, you might need to add memory to the server or upgrade to a model that supports more memory. The Network 9000 Access Server 720 and MAXserver 800, 1600, 1620, and 1640 servers all offer memory upgrade options.

Parameters that Directly Affect Memory Allocation

The values you set for the Server settings in Table 4 and Table 5, and the Port settings in Table 6, control the maximum amount of memory that the server software can allocate. For example, the software can allocate a larger portion of memory when the Server Node Limit setting is 300 than when it is 200. Except for the TextPool Size, the server does not actually allocate memory for these settings until it is needed.

Table 4 - Server Settings that Affect Memory Usage

Characteristic	Default	Minimum	Maximum
Node Limit	100	1	1000
Parameter Server Limit	4	1	8
Queue Limit	24	0	100
Identification Size	63	0	63

Table 5 - Server Session Limit, Server Text Pool

Product	Session Limit Default / Min / Max	Textpool Size Default / Min / Max
MX 1820	64 / 0 / 255	16384 / 8192 / 131070
MX 2120/2220	64 / 0 / 255	8192 / 8192 / 131070
MX 1620/1640/1608A	64 / 0 / 255	16384 / 8192 / 131070
MX 800	64 / 0 / 64	16384 / 8192 / 131070
Network 9000 AS/720	64 / 0 / 255	16384 / 8192 / 131070

Table 6 - Port Settings that Affect Memory Usage

Product	Setting	Minimum	Maximum	Default
MX 2120/2220 server cards, MX 1608A/1620/1640 Standalone Access Servers	Session Limit	0	16	4
	Typeahead Size	80	16384	128
	Command Size	80	16384	80
	Internet TCP Window Size	64	8192	256
Network 9000 Access Server 720	Session Limit	0	16	4
	Typeahead Size	80	16384	128
	Command Size	80	16384	80
	Internet TCP Window Size	64	8192	256
The Typeahead Buffer Size is used during sessions. The Command Buffer Size is used when the command prompt (e.g., Xyplex>) is present. Xyplex recommends that you leave the Command Buffer Size at its default setting.				

Local Services

The server software allocates a fixed amount of memory for the local services that the server makes available. The actual number of available local services depends on the amount of text pool space that is available. (Refer to “[Text Pool Area](#)” on page 12.)

LAT Services

The server stores information about LAT services to which users can make connections. The Server Service Groups setting controls the number of LAT services that are available. You specify a *group-list* for this setting, which has no maximum value.

Domain Names

For domain names, information other than identification strings is stored in non-text pool memory. The server software allocates this memory whenever it stores a domain name. The amount of non-text pool space used to store domain name information depends on the number of domain names being stored, and text pool limitations.

Table 5 lists the default, minimum, and maximum values for the session limit and text pool size for various server models. This table assumes that the server has the basic amount of installed memory. The Show Server Characteristics display shows the current values for these settings.

Identifying Memory Problems

If there is *not enough memory* allocated for a setting, users might not be able to connect to nodes or services because the memory limit has been reached. If *too much memory* is allocated, the server might not have enough memory for other purposes. For example, if you allocate too much memory for the Node Limit setting, the server might not be able to allocate an adequate Typeahead Buffer.

The section “[Adjusting Parameters](#)” on page 19 describes how to adjust various server and port parameters to meet your needs.

Error Messages

The server software generates error and status messages when too much or too little memory is allocated for a setting. For example, when a user attempts to establish a new session or connect to a node or service that is normally available, the following messages might appear:

```
-710- Node node-name not known
-711- Service service-name not known
-719- Insufficient resources to complete operation
-772- Queued access failed, error or no response from
      service
```

The first two messages might indicate that the server does not have enough memory to store the unknown node or service name. The third message might indicate that the server has prevented an operation because too much memory had been allocated already. The last message might indicate that the connection queue is full.

Server Displays

Certain server displays also indicate memory allocation problems. For example, the following displays indicate when users are unable to locate a node, service, or a domain name — possibly indicating that you need to adjust the memory allocation:

```
SHOW/MONITOR DESTINATIONS
SHOW/MONITOR DOMAIN
SHOW/MONITOR NODE
SHOW/MONITOR SERVICE
```

The section “[Helpful Displays](#)” on page 27 describes the useful information in these displays.

Adjusting Parameters

This section describes the settings that affect memory allocation and explains how to change settings to free up memory.

In addition to adjusting these settings, you can use the following procedures to free up memory or text pool space, or to change a server's memory setup.

% Memory Used — Check the % Memory Used field of the Show Server Status display whenever the server uses the maximum amount of memory for a given resource (e.g., the Typeahead Buffer). If the maximum % Memory Used is lower than 80 - 90 percent, you can usually correct the problem by increasing the value for the setting that controls that resource. If the maximum % Memory Used is already very high, you should probably consider freeing up memory elsewhere as well.

Server Identification Size — Decrease the storage requirements for identification strings for local services, LAT services offered on the network, nodes, and domain names. This releases text pool space so that the server can store information about additional services, nodes, and domain names. To do this, you can change the value of the Server Identification Size setting or use shorter service, node, and domain names.

LAT Service Groups — Restrict the number of LAT service groups that the server uses; only use groups that users actually need. This limits the number of LAT services and nodes that offer services. This change is most helpful on a busy server that is part of a large network. Restricting the number of LAT groups results in more text pool and non-text pool space, and more efficient use of LAT virtual circuits.

Typeahead Buffers — Check that the typeahead buffers are not unnecessarily large. This releases memory for other needs.

Command Buffers — Xyplex recommends that you leave the Command Buffer Size at its default setting (80). You typically do not need to increase it — *even if you need to increase the Typeahead Size*.

Local Services — Move a local service to a less heavily loaded server. This releases small amounts of both text pool and non-text pool memory.

Packet Buffers — Some features and network protocols, such as IP reassembly, Xremote, SLIP, and PPP, require a large number of packet buffers for ideal performance. If you are running one of these protocols and experience poor performance, you should increase the packet buffer count. PPP requires five packet buffers for port speeds less than 57.6 Kbps and 10 buffers for speeds of 57.6 Kbps or greater. SLIP always requires 10 packet buffers. (Refer to [page 23](#).)

TCP Window Size — The Port Internet TCP Window Size setting specifies the size of the TCP window used during a session. If you define a window size that is too large, the server might not have enough memory to support the usual number of sessions. (Refer to [page 26](#) for more information.)

Server Node Limit

If users are unable to connect to a node that *should* be available, or cannot locate a node or service in a Show Node, Show Service, or Show Destinations display — or if the Show Server Status display indicates Discarded Nodes or Resource Errors — you should check the Show Server Alternate Status display and the Reachable Nodes and Connected Nodes fields of the Server Status display.

If the “highest” load is the same as the “maximum” load in the Reachable Nodes field, the Server Node Limit setting might be set too low. If the highest load is less than the maximum load, then the Server Textpool Size might be set too low.

The Server Alternate Status display also indicates whether there is a problem with text pool memory or non-text pool memory.

The Connected Nodes field in the Server Status display shows the number of service nodes to which the server has established a LAT virtual circuit. If the “current” value is the same as the “maximum” value, the server will not permit a user to connect to another node.

If this is causing connection failures, you do not have a memory allocation problem. To eliminate this problem, decrease the number of virtual circuits that the server needs to establish, by moving users who usually connect to a given node to a different server.

Server Queue Limit

When the Solicitations Rejected and Solicitations Accepted fields of the Show Server Counters display indicate a large number of rejected queued connection requests — relative to the number of accepted requests — check the Queue Limit field of the Server Status display. If the “high” number of queue entries in this field is the same as the “maximum” number, you might need to increase the Server Queue Limit setting.

If the number is lower, connection requests might be failing for other reasons. Otherwise, if you have noticed problems with other resources, you might need to lower the value for the Server Queue Limit setting and raise the value for the setting with which you are having the problem.

Server Session Limit

The Session Limit setting determines the maximum number of active sessions the server can support concurrently. If users are unable to make additional connections because the server has reached this limit — or because of a resource limitation — the Session Limit might be set too low. Check the Connected Sessions and Resource Error fields of the Show Server Status display.

The Session Limit might be too low if the “highest” load equals the “maximum” allowed. If the highest load is less than the maximum allowed, and there are resource errors, you must free up additional memory to allow for additional sessions.

Server Textpool Size

You control the size of the text pool area through the Server Textpool Size setting. Since the text pool area is permanently allocated, you reduce the amount of memory available for non-text pool memory when you increase the size of this area. Also, lengthy identification strings can fill up the text pool space and limit the number of nodes, services, and domain names available to users.

If users cannot connect to a node or service that *should* be available, or are unable to locate a node or service in a Show Node or Show Service display — or if the Show Server Status display indicates that “Discarded Nodes” or “Resource Errors” have occurred, check the Show Server Alternate Status display. If this display indicates that there have been “Free Text Pool” errors, you should increase the Textpool Size.

If the Show Server Alternate Status display indicates that there have been “Free Memory” failures, you should decrease the Textpool Size. If this is not possible, consider reducing the size of the node and service identification strings that must be stored in the text pool area. You can do this for node names, domain names, local services, and LAT services offered by other hosts and servers on the network.

Server Packet Count

Some features and network protocols, such as IP reassembly, Xremote, and PPP, require a large number of packet buffers for ideal performance. If you are running one of these protocols, and have experienced poor performance, you should increase the packet buffer count through the following command:

```
DEFINE SERVER PACKET COUNT packet-buffers
```

You can specify a *packet-buffers* value from 80 to 1088 for units with 2+ MB of memory, and 80 to 160 for units with less than 2 MB of memory. The default is 80.

Increasing the number of packet buffers can improve performance and response time, but doing so will decrease the amount of available free memory. The Show/Monitor Server Alternate Status display lists the number of packet buffers being used and the maximum number available.

Parameter Server Limit

A server does not require much memory to store information about parameter servers. If a heavily used server is unable to store information about all eligible parameter servers, its Server Parameter Server Limit setting is set too low. You can adjust the setting to match the number of available parameter servers in your network, through the following command. The maximum value for the setting is eight; the default is four.

```
DEFINE SERVER PARAMETER SERVER LIMIT number
```

If there are more than eight parameter servers in your network, you can use the Set Parameter Server command to define specific parameter servers, and the Clear Parameter Server command to remove other parameter servers.

Port Typeahead Size

The Port Typeahead Buffer can contain between 80 and 16384 characters. The default buffer size, 128 bytes, is adequate for most situations. You only need to increase this value if an application requires it (e.g., certain VAX/VMS applications).

VAX/VMS Typeahead Buffers

VAX/VMS terminal sessions use a separate typeahead buffer or alternate typeahead buffer, which is specified through VMS SYSGEN settings. If the size of the server typeahead buffer is larger than the VAX typeahead buffer, the VAX typeahead buffer can overrun during data transfers.

During asynchronous port use, this overrun might cause the VAX session to issue ASCII BELL characters to the server, rather than XOFF characters. This is most noticeable when a user continuously presses an arrow key while using an editor, or when using an asynchronous file transfer protocol, such as XMODEM.

If this problem occurs, the VAX system manager can specify a typeahead buffer size, or alternate typeahead buffer size, that is large enough to support packets sent by the server and the asynchronous file transfer protocol.

For example, the XMODEM software package properly transfers binary files from a host to a PC when using a typeahead buffer size of 512 characters. On a system with VMS 5.0, the VMS system manager would run the SYSGEN program, enter the following commands from the SYSGEN> prompt, and then reboot the system:

```
SYSGEN> SET TTY_TYPAHDSZ 512  
SYSGEN> WRITECURRENT
```

When you use an asynchronous file transfer protocol, you can increase the server typeahead buffer to match the VMS typeahead buffer size through the Set Port Typeahead Size command. However, since the larger VMS typeahead buffer applies to all terminals, increasing its size will use more of the computer's memory.

Port Command Buffer Size

The Port Typeahead Buffer can contain between 80 and 16384 characters. The default buffer size, 80 bytes, is enough for most uses. Xyplex recommends that you leave the default setting intact — *even if you need to increase the Typeahead Buffer Size*.

This setting controls the number of characters that can be buffered (temporarily stored) at the command prompt (e.g., Xyplex>) before overruns occur. It differs from the port typeahead setting, which is used during sessions.

Port Internet TCP Window Size

The Port Internet TCP Window Size setting specifies the size of the TCP window used during a session. Valid values are whole numbers between 64 and 8192. The default value is 256. If you define a window size greater than 256, the server might not have enough memory for the usual number of sessions. A typical TCP/IP session requires about:

$$[1600 + (3 * \text{TCP_window_size})] \text{ bytes}$$

The window size that is in effect when a session begins remains in effect for the entire session.

Helpful Displays

This section describes displays that provide useful information when you are monitoring memory usage.

Show/Monitor Server Counters

This display shows statistics about server activity, and indicates when errors have occurred. Important fields include:

- Solicitations Accepted
- Solicitations Rejected

Refer to “[Server Queue Limit](#)” on page 21 for descriptions of these fields.

Show/Monitor Server Status

This display indicates how well the server is operating under the current load and can be helpful in identifying network or port problems. Table 7 describes the important fields:

Table 7 - Server Status Display Fields

Field	Description
Queue Entries	The number of connection requests in the server's connection queue. Refer to “ Server Queue Limit ” on page 21 for more information.
Available Services	The number of LAT services for which the server is storing information in its memory, and which are therefore available to users. Refer to “ LAT Services ” on page 16 and “ Server Textpool Size ” on page 22 for more information.
Local Services	The number of LAT services offered at the server. If you have a problem creating a new local service, refer to “ LAT Services ” on page 16.

Managing Server Resources

Reachable Nodes	The number of LAT nodes, including computers and other servers, that offer services and are reachable for service connections. Refer to “Server Node Limit” on page 20 and Server Textpool Size on page 22 for more information.
Connected Nodes	The number of service nodes to which the server has connected a LAT virtual circuit. Refer to “Server Node Limit” on page 20 for more information.
Connected Sessions	The total number of sessions that the server has connected. Refer to “Server Session Limit” on page 22 for more information.
% CPU Used	The percentage of processing time that the server has used — calculated every second. This indicates of how busy the server is or has been.
% Memory Used	The percentage of the server memory pool that is being used to store information for the node and service database, queued requests, and sessions.
Discarded Nodes	The number of nodes that the server cannot include in its node database because the Server Node Limit has been reached or because there is not enough memory. Refer to “Server Node Limit” on page 20 and “Server Textpool Size” on page 22 for more information.
Resource Errors	The number of times that the server could not create an internal data structure because it did not have enough memory. Refer to “Server Textpool Size” on page 22 for more information.

Show/Monitor Server Alternate Status

This display shows how well the server's memory is operating under the current load. For each resource listed, the display shows the Current, High, and Maximum values since the server was last booted. It also shows the number of failures (insufficient amount) associated with each resource, and when the last failure occurred.

Table 8 describes the important fields:

Table 8 - Server Alternate Status Fields

Field	Description
Free Text Pool	The amount of text pool space used; the number of times an operation was attempted, but for which there was not enough text pool space; and when the last failure occurred.
Free Memory	The amount of non-text pool space used; the number of times an operation was attempted, but for which there was not enough non-text pool space; and when the last failure occurred.
Packet Buffers	The number of incoming and outgoing packets that are being buffered (stored temporarily) in memory.

Chapter 3

Using TCP/IP Features

Xyplex Access Servers support connections to TCP/IP nodes. They also support several Internet protocols and features. This chapter describes how to configure a server's Internet-related settings and enable Telnet features. It covers these topics:

- Defining Basic Internet Settings
- Configuring Internet Routes
- IP and IPX Traffic Filtering¹
- IPX RIP and SAP Import/Export Filters
- Configuring Rotary Connections
- Configuring RLOGIN support
- Remote Management Support
- Loading through Internet Protocols
- Configuring the Access Server as a Domain Name Server
- Using IP Reassembly
- Using TCP Resequencing

¹ Although IPX RIP and SAP filtering is not related to TCP/IP, it is included in this chapter so that all routing protocol filters can be covered in a single chapter.

Defining Basic Internet Settings

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

IP Address and Subnet Mask

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 internet primary domain address
        128.3.0.200

Xyplex>> define server internet secondary domain address
        128.3.0.250
Xyplex>> set server internet 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.

Internet Broadcast Address

This setting specifies the server's IP address that is used in Internet 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 internet broadcast address
          172.19.255.255
Xyplex>> set server internet broadcast address
          172.19.255.255
```

Internet Primary and Secondary Gateways

An access server can use an Internet 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 internet primary gateway 172.19.1.2
Xyplex>> set server internet primary gateway 172.19.1.2

Xyplex>> define server internet secondary gateway
          172.19.1.3
Xyplex>> set server internet secondary gateway 172.19.1.3
```

Show/List/Monitor Server Internet Characteristics

Use this command to view the current settings for Internet-related settings. Figure 1 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 Terminal 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 1 - Server Internet Characteristics Display

Configuring IP Routes

Large networks with many hosts and servers are often divided into smaller, separate networks. These subnetworks, or “subnets,” can exist in the same or separate locations. Sites with a small number of devices, which are connected through routers to larger IP networks (e.g., the Internet), can also be divided into subnets.

The Internet protocol supports communication between devices on separate networks and subnets through gateways (or routers). An IP route specifies the preferred router for routing data traffic to a remote network or subnet. The server maintains a table of IP routes, called the “IP route table,” in both its operational and permanent databases.

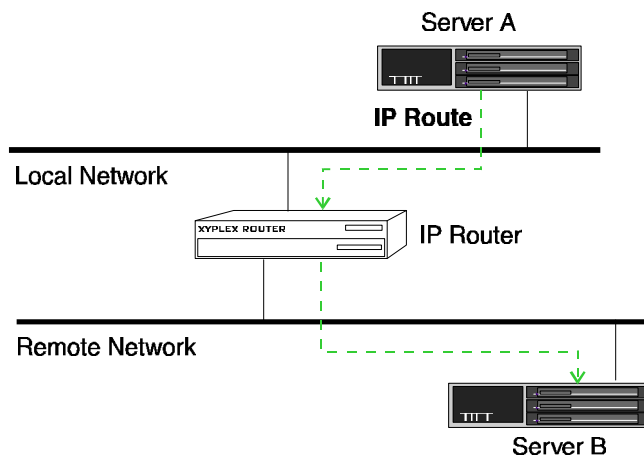


Figure 2 - IP Route

Host and Network Routes

There are two types of IP route table entry: host entries and network entries. Each specifies a gateway to use. When the server attempts to communicate with an Internet node, it compares the node's IP address and subnet mask with its own address and mask, to determine whether the node resides on the same network. If it does, the server sends the data traffic directly to the node.

If the node does not reside on the same network, the server searches its IP route table to find a matching host entry. If it finds a match, the server sends the data traffic to the gateway indicated in the table. The gateway forwards the traffic to the destination. If the server does not find a matching host entry, it looks for a matching network entry.

If it still does not find a match, the server sends the traffic to the primary or secondary gateway. (Refer to [page 34](#).)

Dynamic Routing

IP routers communicate among themselves, forwarding network traffic to each other and between networks, using routing protocols such as the Routing Information Protocol (RIP). The routers also select the most efficient path to remote networks; this is called “dynamic routing.”

As conditions change, the path to a remote network might change. When this happens, the access server is informed through routing messages that the packets it has sent to a particular router were forwarded to a different router on the same network. This ensures that the packets eventually reach the destination. IP routes that the server obtains in this way are called “learned” routes.

NOTE: Xyplex Access Servers “listen” to routing protocol messages when the `routd` daemon is enabled; however, they do not actively participate in the protocols.

Static Routing

For some networks with multiple routers, dynamic routing might not be enabled or available. At these sites, the access server must select a specific router to insure that packets are forwarded to the right destination. To do this, the host or server manager specifies database entries that map specific destination networks or hosts to specific routers. This is called “static routing.” Static IP routes are also called “locally specified IP routes.”

The access server can store both dynamic and static IP routes. The software prevents duplicate entries (entries where the destination address and subnet mask match an existing entry).

Define/Set Server Internet Route

Use the following commands to define a locally specified (static) IP routes. You can define up to 64 routes. (You can increase the maximum number of routes by adjusting the Server Internet Routing Table Size.)

```
DEFINE/SET SERVER INTERNET ROUTE ip-address [route-spec]
```

Valid *route-specs* follow:

```
GATEWAY gateway-ip-address  
GATEWAY gateway-ip-address FIXED/VARIABLE  
GATEWAY gateway-ip-address MASK subnet-mask  
GATEWAY gateway-ip-address MASK subnet-mask FIXED/VARIABLE  
GATEWAY gateway-ip-address HOST  
GATEWAY gateway-ip-address HOST FIXED/VARIABLE
```

If you do not specify the HOST keyword, the server assumes that the entry is a network route. The FIXED keyword specifies that the server must not modify the route table entry based on routing messages that it receives. (This is the default setting.) The VARIABLE keyword specifies that it may modify the entry.

Examples:

```
Xyplex>> define server internet route 192.168.12.1 gateway  
192.168.10.1  
Xyplex>> set server internet route 192.168.12.1 gateway  
192.168.10.1  
  
Xyplex>> define server internet route 192.168.12.101  
gateway 192.168.10.1 host  
Xyplex>> set server internet route 192.168.12.101  
gateway 192.168.10.1 host  
  
Xyplex>> define server internet route 192.168.12.67  
gateway 192.168.12.33 mask 255.255.255.224  
Xyplex>> set server internet route 192.168.12.67  
gateway 192.168.12.33 mask 255.255.255.224
```

Clear/Purge Server Internet Route

Use these commands to remove a static IP route from the server's databases:

```
CLEAR/PURGE SERVER INTERNET ROUTE entry  
CLEAR/PURGE SERVER INTERNET ROUTE ALL
```

An *entry* refers to the entry number that corresponds to the route in the List/Show/Monitor Server Internet Routes display. (See Figure 3.)

Show/List/Monitor Server Internet Route

Use these commands to view all currently available IP routes. Figure 3 shows a sample display:

Xyplex> SHOW SERVER INTERNET ROUTE ALL						
	Address	Gateway	Mask		Last Modified	
1	192.168.10.101	192.168.11.1	255.255.255.0	NET/FIXED	21 Mar 1996	09:22
2	192.168.11.101	192.168.11.2	0.0.0.0	HOST/VAR	21 Mar 1996	09:22

Figure 3 - Server Internet Routes Display

IP Traffic Filtering

IP traffic filters determine which IP sources and destinations can communicate with each other through the access server's ports. You can set up these filters on the LAN interface or on individual ports. You define filters on a per-port basis when users/clients gain access to the server through a network protocol, such as SLIP or PPP.

The access server applies the filters to IP packets as it *receives* them from the attached LAN or specified port(s).

A traffic filter specifically allows or restricts traffic between two points; for example, between a dial-in client and a network host. (See Figure 4 on [page 46](#).) Traffic filters also determine which IP protocols the access server may forward, and can allow or restrict communication through specific TCP and UDP ports.

Additionally, IP traffic filters can allow or restrict the forwarding of TCP packets when the packet's SYN bit is set to ON and the ACK bit is set to OFF. This bit pattern indicates that the sender is trying to open a new session with a destination port. By discarding packets with this bit pattern, you prevent remote users from opening sessions with hosts on the local network.

Enabling IP Filtering

By default, IP traffic filtering is disabled. Use this command to enable it, or to disable it later:

```
DEFINE SERVER IP FILTERING [ENABLED]  
[DISABLED]
```

IP Traffic Filter Criteria

You can specify the following criteria in an IP traffic filter:

- IP protocol type: specific protocol ID number, TCP, UDP, or ALL
- Destination IP address and subnet mask
- Destination port number or range of numbers
- Source port number or range of numbers
- Source IP address and subnet mask
- Whether a TCP packet has its SYN bit ON and its ACK bit OFF

Determining the Most Specific Filter

When an inbound IP packet is neither TCP nor UDP, the server ignores any filters that specify an individual TCP or UDP port, or range of ports. *If it receives a packet that has multiple matching filters, the server applies the most specific filter.* The server uses the following process to determine the most specific filter:

1. The server first looks for matching filters that specify *an IP protocol type* (including type ALL). If one or more filters meet this condition, the server ignores filters that do not specify an IP protocol type. If only one matching filter specifies an IP protocol type, the server applies that filter.
2. If more than one matching filter specifies an IP protocol type, the server looks for filters that match the *range of destination TCP or UDP ports* (i.e., the range that the TCP or UDP port falls within). If multiple filters specify an equal range of port numbers, the server looks for filters with the lowest beginning port number. If only one matching filter specifies the narrowest range of destination TCP or UDP ports, the server applies that filter.
3. If more than one of the remaining filters specify the narrowest destination TCP/UDP port range, the server looks for filters that specify a *TCP SYN value* (ON or OFF). If only one filter specifies a TCP SYN value, the server applies that filter.
4. If more than one of the remaining filters specify a TCP SYN value, the server looks for filters that specify the *range of source TCP or UDP ports*. When multiple filters specify an equal range of port numbers, the server looks for filters with the lowest beginning port number. If only one matching filter specifies the narrowest range of source TCP/UDP ports, the server applies that filter.

5. If more than one of the remaining filters specify the narrowest range of source TCP or UDP ports, the server looks for matching filters that specify the *destination IP subnet*. If only one filter specifies the largest destination IP subnet mask, the server applies that filter.
6. If more than one of the remaining filters specify the largest destination IP subnet mask, the server looks for matching filters that specify the *source IP subnet*. At this point only one filter should remain; the server applies that filter.

Traffic Filter Commands

Use the following commands to define IP traffic filters. Note that the Define/Set Server commands affect packets that the server receives from the attached LAN. Define/Set Port commands affect packets received through individual ports.

DEFINE/SET SERVER IP FILTER *criteria instructions*

DEFINE/SET PORT *port-list* IP FILTER *criteria instructions*

Valid *criteria* follow:

PROTOCOL [*protocol-id*] (0-to-255)

[TCP]

[UDP]

[**ALL**]

DESTINATION PORT [*port-number*] (0-to-65535)

[*port-number* - *port-number*] (0-to-65535 - 0-to-65535)

[**ALL**]

SYN [ON]

[**ALL**]

SOURCE PORT [*port-number*] (0-to-65535)
 [*port-number* - *port-number*] (0-to-65535 - 0-to-65535)
 [**ALL**]

DESTINATION [*ip-address* [MASK *subnet-mask*]]
 [**ALL**]

SOURCE [*ip-address* [MASK *subnet-mask*]]
 [**ALL**]

Valid *filter-instructions* follow:

DISCARD
FORWARD

Protocol

The TCP or UDP Protocol ID number.

SYN ON/ALL

SYN ON means that the SYN (synchronization) bit is set to ON and the ACK (acknowledge) bit is set to OFF in the TCP header. SYN ALL represents any value for the SYN and ACK bits.

Example 1

A network manager wants to allow Nodes 105 (192.168.22.**105**), 106 (192.168.22.**106**), and 107 (192.168.22.**107**) to access hosts on the corporate LAN through the server, using a PPP. No other nodes connected to the access server may access these nodes.

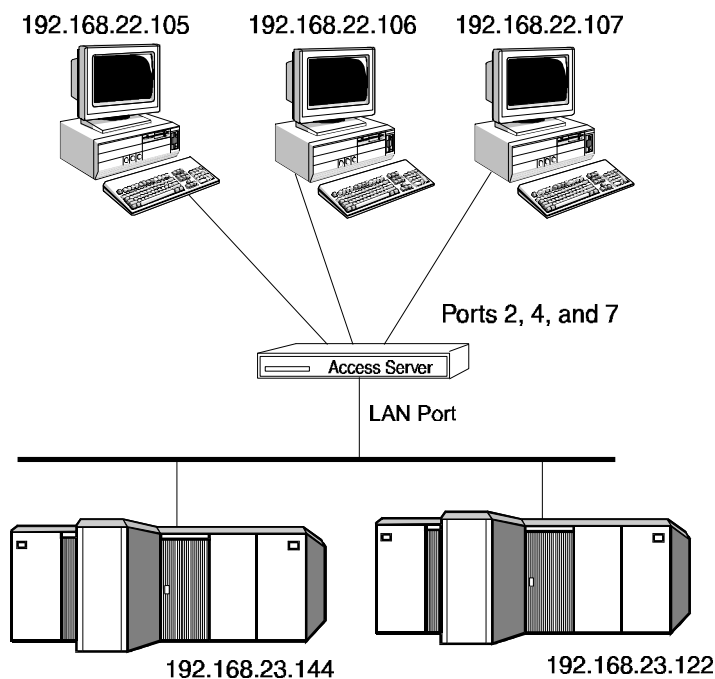


Figure 4 - IP Traffic Filter Example

The network manager defines these IP traffic filters:

- 1) Xyplex>> define port all ip filter destination
192.168.23.144 discard
- 2) Xyplex>> define port all ip filter destination
192.168.23.122 discard
- 3) Xyplex>> define port 2 ip filter source 192.168.22.105
destination 192.168.23.144 forward

- 4) Xyplex>> define port 2 ip filter source 192.168.22.105
destination 192.168.23.122 forward
- 5) Xyplex>> define port 4 ip filter source 192.168.22.106
destination 192.168.23.144 forward
- 6) Xyplex>> define port 4 ip filter source 192.168.22.106
destination 192.168.23.122 forward
- 7) Xyplex>> define port 7 ip filter source 192.168.22.107
destination 192.168.23.144 forward
- 8) Xyplex>> define port 7 ip filter source 192.168.22.107
destination 192.168.23.122 forward

Because Filters 3 - 8 specify source IP addresses, they are more specific than Filters 1 and 2, which only specify destinations. Therefore, Filters 3 - 8 override Filters 1 and 2.

Example 2

This example builds upon the scenario described in the previous example. Now, the network manager wants to deny Telnet access to the hosts on the corporate LAN. The TCP port number for Telnet is 23. The network manager defines these filters:

```
Xyplex>> define port 2,4,7 ip filter dest port 23 dest  
192.168.23.122 discard  
  
Xyplex>> define port 2,4,7 ip filter dest port 23 dest  
192.168.23.144 discard
```

Because these filters specify a TCP port number, they are more specific than the source/destination filters defined in Example 1. Therefore, they override the action of the filter defined in Example 1. As a result, the nodes connected to Ports 2, 4, and 7 cannot access the hosts through Telnet, but can still access them through other IP protocols.

Example 3

This example builds upon the scenarios described in the previous examples. Now, the network manager wants to prevent the nodes connected to Ports 2, 4, and 7 from sharing files with the hosts on the corporate LAN through UDP Network File Sharing (NFS). To do this, the network manager prevents UDP broadcasts with this filter:

```
Xyplex>> define port 2,4,7 ip filter protocol udp discard
```

Because this filter specifies a protocol, it is more specific than the filters defined in the previous examples. Therefore, it overrides the filters defined in the previous examples.

IPX Traffic Filters

NOTE: Although IPX Traffic Filtering is not a TCP/IP feature, it is described here so that all routing protocol filters can be covered in a single chapter.

IPX traffic filters determine which IPX sources and destinations may communicate with each other through the access server's ports. A traffic filter specifically allows or restricts traffic between two points; for example, between two networks or between a NetWare client and server. (See [Figure 5](#) on page 52.)

You configure traffic filters on a server and/or individual port basis. The server applies them to packets as it receives them.

Traffic Filter Criteria

You can specify the following criteria in a traffic filter:

- Destination IPX network and/or Ethernet address
- Source IPX network and/or Ethernet address

Determining the Most Specific Filter

When the server receives an IPX packet that meets the criteria of more than one traffic filter, it applies the *most specific filter*. The server uses the following process to determine which traffic filter is most specific:

1. The server looks for filters that specify a matching **destination IPX network**. If one or more matching filters meet this condition, the server ignores any filters that do not. If only one of the remaining filters specifies a matching destination network, the server applies that filter.
2. If more than one filter specifies a matching destination network, the server looks for filters that specify a matching **destination Ethernet address**. If one or more matching filters meet this condition, the server ignores any filters that do not. If only one of the remaining filters specifies a matching destination address, the server applies that filter.
3. If more than one of the remaining filters specify a matching destination address — or if no filters do — the server looks for filters that specify a matching **source IPX network**. If one or more matching filters meet this condition, the server ignores any filters that do not. If only one of the remaining filters specifies a matching source network, the server applies that filter.
4. If more than one of the remaining filters specify a matching source network — or if no filters do — the server looks for a filter that specifies a matching **source Ethernet address**. At this point only one filter should remain; the server applies that filter.

Traffic Filter Commands

By default, IPX traffic filtering is **disabled**. Use this command to enable it, or to disable it later:

```
DEFINE SERVER IPX FILTERING [ENABLED]
                             [DISABLED]
```

Use these commands to define IPX traffic filters:

```
DEFINE SERVER IPX FILTER destination-criteria [FORWARD]
                                                             [DISCARD]

DEFINE SERVER IPX FILTER source-criteria [FORWARD]
                                                             [DISCARD]

DEFINE SERVER IPX FILTER dest-criteria source-criteria [FORWARD]
                                                             [DISCARD]

DEFINE PORT port-list IPX FILTER destination-criteria [FORWARD]
                                                             [DISCARD]

DEFINE PORT port-list IPX FILTER source-criteria [FORWARD]
                                                             [DISCARD]

DEFINE PORT port-list IPX FILTER dest-criteria source-criteria
[FORWARD]
                                                             [DISCARD]
```

The *destination-criteria* can include:

```
DESTINATION NETWORK [ipx-network]
                    [ALL]

DESTINATION NODE [node-address] (e.g., 08008712AB34)
                    [ALL]
```

Using TCP/IP Features

The *source-criteria* can include:

```
SOURCE NETWORK [ipx-network]  
                [ALL]
```

```
SOURCE NODE [node-address] (e.g., 08008712ABCD)  
            [ALL]
```

NOTE: The Show Server IPX RIP Status and Show Server IPX SAP Status displays list IPX *node-addresses*.

Example

A network manager wants to allow users on Ports 2, 4, and 7 to access two file servers on the corporate LAN (Network BBBB). The server users *may not* access any other nodes on the corporate LAN.

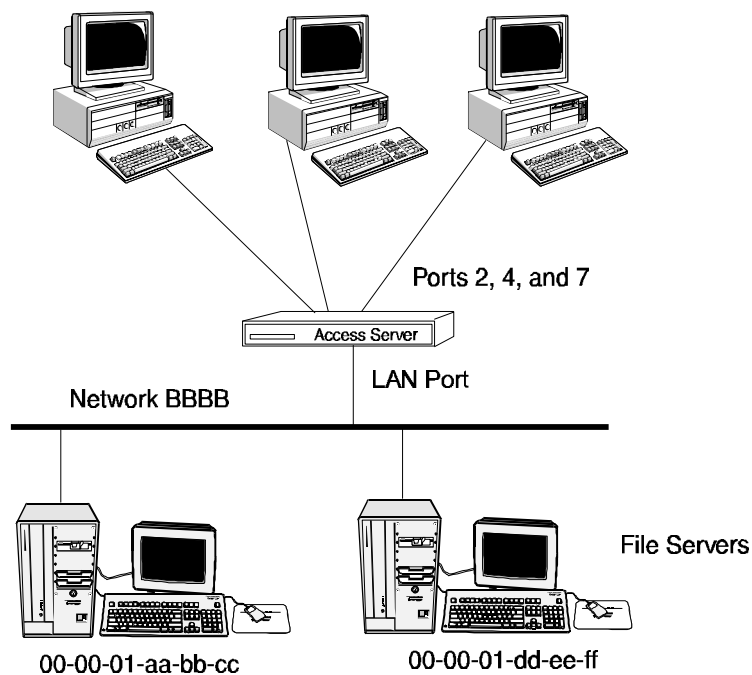


Figure 5 - IPX Traffic Filter Example

First, the network manager defines a traffic filter that discards all traffic to the corporate LAN by default:

```
Xyplex>> define port all ipx filter dest network bbbb  
          discard
```

Next, the network manager defines filters that allow users at Ports 2, 4, and 7 to forward traffic to the two file servers on Network BBBB:

```
Xyplex>> define port 2,4,7 ipx filter dest node  
          000001aabbcc forward
```

```
Xyplex>> define port 2,4,7 ipx filter dest node  
          000001ddeeff forward
```

Because these filters are more specific than the previously defined filter, the server applies them, and consequently allows traffic to be forwarded to the two destination addresses.

IPX-RIP Import/Export Filters

By default, when the IPX protocol is enabled, a Xyplex Access Server adds all routes that it learns through RIP to its IPX route table. This process is called *importing*. The server also advertises all routes in its IPX route table to other IPX routers, by default. This process is called *exporting*.

You can define filters that prevent the server from importing or exporting routes. You can define these filters on a server-wide basis or on an individual port basis. Import filter rules specify whether routes learned through RIP are “accepted” or “discarded.” Export filter rules specify whether routes in the route table are “advertised” or “hidden.” You can apply filters to specific networks or to ranges of networks.

NOTE: If an IPX service advertisement (SAP table entry) is associated with a filtered RIP packet, the filter rule is also applied to the SAP table entry.

When There Are Multiple Matching Filters

When checking its RIP filter database, the server might find more than one matching filter. When this happens, *the server applies the most specific filter*. A filter that specifies an actual network is considered more specific than one that specifies Network ALL.

Defining RIP Filters

By default, IPX filtering is disabled. Use this command to enable it, or to disable it later:

```
DEFINE SERVER IPX FILTERING [ENABLED]  
                             [DISABLED]
```

IPX-RIP Export Filters

Use the following commands to define IPX-RIP export filters. Note that using the SERVER keyword affects routes that the server learns through the attached LAN. The PORT keyword affects routes that the server learns through a specified port.

```
DEFINE/SET SERVER IPX RIP EXPORT NETWORK [network] [ADVERTISE]
                                                [ALL]      [HIDE]

DEFINE/SET PORT port-list IPX RIP EXPORT NETWORK [network] [ADVERTISE]
                                                [ALL]      [HIDE]
```

A *network* is a hexadecimal value from 1 to FFFFFFFF. Use these commands to remove RIP export filters:

```
CLEAR/PURGE SERVER IPX RIP EXPORT [ALL]
                                   [NETWORK [network]]
                                   [ALL]

CLEAR/PURGE PORT port-list IPX RIP EXPORT [ALL]
                                           [NETWORK [network]]
                                           [ALL]
```

Examples

```
Xyplex>> define server ipx rip export network all hide
Xyplex>> define port all ipx rip export network all hide
Xyplex>> define server ipx rip export network 1234
          advertise
Xyplex>> define port all ipx rip export network 1234
          advertise

Xyplex>> purge server ipx rip export all
Xyplex>> purge server ipx rip export network network 1234
Xyplex>> purge port all ipx rip export network 1234
```

RIP Import Filters

Use the following commands to define RIP import filters. Note that using the SERVER keyword affects routes that the server learns through the attached LAN. The PORT keyword affects routes that the server learns through a specified port.

```
DEFINE SERVER IPX RIP IMPORT NETWORK [network] [ACCEPT]
                                         [ALL]      [DISCARD]

DEFINE PORT port-list IPX RIP IMPORT NETWORK [network] [ACCEPT]
                                         [ALL]      [DISCARD]
```

A *network* is a hexadecimal value from 1 to FFFFFFFE. Use these commands to remove RIP import filters:

```
CLEAR/PURGE SERVER IPX RIP IMPORT [ALL]
                                         [NETWORK [network]]
                                         [ALL]

CLEAR/PURGE PORT port-list IPX RIP IMPORT [ALL]
                                         [NETWORK [network]]
                                         [ALL]
```

Examples

```
Xyplex>> define server ipx rip import network all discard
Xyplex>> define port all ipx rip import network all discard
Xyplex>> define server ipx rip import network 1234 accept
Xyplex>> define port all ipx rip import network 1234 accept

Xyplex>> purge server ipx rip import network all
Xyplex>> purge port all ipx rip import all
Xyplex>> purge server ipx rip import network 1234
Xyplex>> purge port all ipx rip import network 1234
```

SAP Import/Export Filters

By default, when the IPX protocol is enabled, a Xyplex Access Server adds all service “Names” and “Types” that it learns through the Service Advertisement Protocol (SAP) to its IPX SAP table. This process is called *importing*. The server also advertises all service Names and Types in its SAP table to other IPX routers, by default. This process is called *exporting*.

You can define filters that prevent the server from importing or exporting service Names and Types. Import filter rules specify whether service Names and Types learned through SAP are “accepted” or “discarded.” Export filter rules specify whether service Names and Types are “advertised” or “hidden.” You can apply SAP filters to specific networks or to ranges of networks.

NOTE: If a service advertisement (SAP) is associated with a filtered RIP packet, the server applies the filter rule to the SAP also. (RIP import/export filters are described on [page 54](#).)

SAP Filter Criteria

SAP filters identify services through one or more of these criteria:

- Network — the source or destination network
- Type — The NetWare service type; e.g., File Server, Printer

The `SHOW IPX SAP Display` shows the Network and Type information associated with each SAP in the server’s database.

When There Are Multiple Matching Filters

When checking its SAP filter database, the server might find multiple matching filters. When this happens, the server uses the following rules to determine which filter to apply:

1. Only consider filters that indicate a specific network.
2. If multiple matching filters still exist, only consider the filter(s) that indicate a specific Type.
3. At this point only one filter remains. The server applies its rule.

Defining SAP Filters

By default, IPX filtering is disabled. Use this command to enable it, or to disable it later:

```
DEFINE SERVER IPX FILTERING [ENABLED]
                             [DISABLED]
```

SAP Export Filters

Use the following commands to define SAP export filters. Note that using the SERVER keyword affects services that the server learns through the attached LAN. The PORT keyword affects services that the server learns through a specified port.

```
DEFINE/SET SERVER IPX SAP EXPORT sap-characteristics [ADVERTISE]
                                                         [HIDE]
```

```
DEFINE/SET PORT port-list IPX SAP EXPORT sap-characteristics
[ADVERTISE]
                                                         [HIDE]
```

Use these commands to remove SAP export filters:

```
PURGE/CLEAR SERVER IPX SAP EXPORT [sap-characteristics]
[ALL]
```

```
PURGE/CLEAR PORT port-list IPX SAP EXPORT [sap-characteristics]
[ALL]
```

Valid *sap-characteristics* can include:

```
NETWORK [network]
[ALL]
```

```
TYPE [type-value]
[ALL]
```

<i>type-value</i>	Description
0	Unknown
1	User
2	User Group
3	Print Queue
4 or 278	File Server
5	Job Server
6	Gateway
7	Print Server
8	Archive Queue
9	Archive Server
A	Job Queue
B	Administration
24	Remote Bridge Server
47	Advertising Printer Server
107	Server (internal)

A network is a hexadecimal value from 0 to FFFFFFFF.

Examples

```
Xyplex>> define server ipx sap export network all hide
Xyplex>> define port all ipx sap export network all hide
Xyplex>> define server ipx sap export network 1234 type 6
          advertise
Xyplex>> define port all ipx sap export network 1234 type 6
          advertise
Xyplex>> purge server ipx sap export network all
```

SAP Import Filters

Use the following commands to define SAP import filters. Note that using the SERVER keyword affects services that the server learns through the attached LAN. The PORT keyword affects services that the server learns through a specified port.

```
DEFINE SERVER IPX SAP IMPORT sap-characteristics [ACCEPT]
                                                [DISCARD]

DEFINE PORT port-list IPX SAP IMPORT sap-characteristics [ACCEPT]
                                                [DISCARD]
```

Use these commands to remove SAP import filters:

```
PURGE/CLEAR SERVER SAP IMPORT [sap-characteristics]
                               [ALL]

PURGE/CLEAR PORT port-list SAP IMPORT [sap-characteristics]
                               [ALL]
```

The *sap-characteristics* are described on [page 59](#).

Examples

```
Xyplex>> define server ipx sap import network all discard
Xyplex>> define port all ipx sap import network all discard
Xyplex>> define server ipx sap import network 1234 type 4
          accept <allow file servers only>
Xyplex>> define port all ipx sap import network 1234 type 4
          accept <allow file servers only>
Xyplex>> purge server ipx sap import network all
Xyplex>> purge port all ipx sap import network all
```

Configuring Rotary Connections

The term “rotary” refers to the assignment of an IP address or domain name to multiple destinations that offer the same type of service. When a user attempts to connect to the IP address or domain name, the server connects him to an available port that has been assigned that address or domain name. (See [Figure 6](#) on page 63.)

The access server supports two rotary connection types:

- Assigning an IP address to a group of ports on a server
- Assigning a single domain name to several ports or groups of ports, which can be on multiple servers

You can use these methods together or separately. For example, while a given IP address for a group of ports can only be used at a single server, you can assign many IP addresses to the same domain name. Generally, this requires that you have a domain name server that returns multiple IP addresses for a single name. Figure 6 shows both of these applications.

The rotary is transparent to users. A user simply requests a connection to a destination domain name or IP address, and the server sets up the connection with one of the available ports in the rotary group.

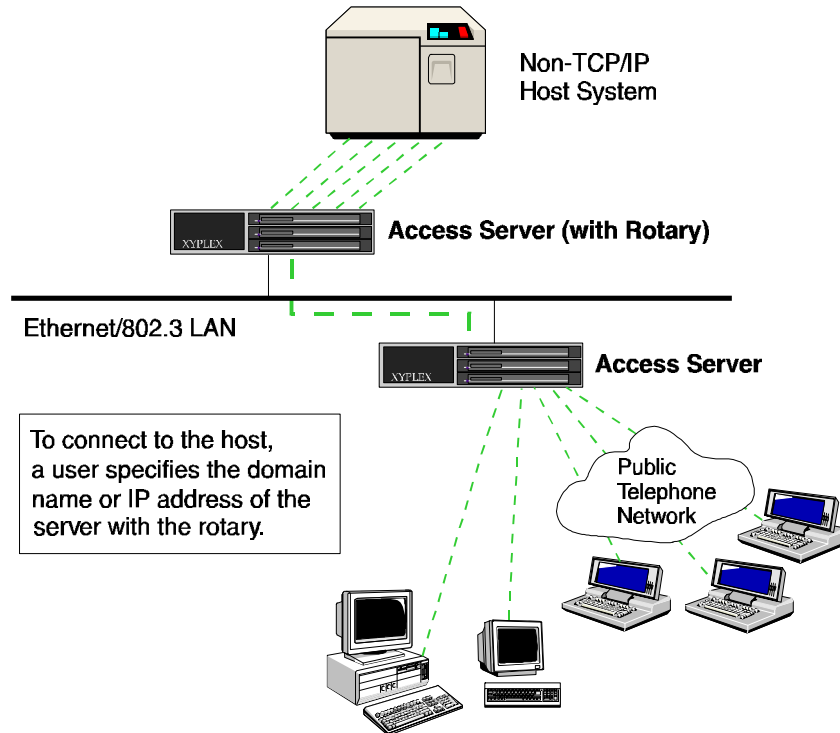


Figure 6 - Rotary Connections

When a user tries to connect to a rotary that is configured on one or more other Xyplex servers, the servers set up the connection on the least recently used server in the rotary group. When connecting to a rotary through a domain name, the server tries to connect to the least recently tried IP address for that domain.

If the user has Autoconnect enabled, the server tries each available IP address until it makes a connection. Otherwise, the server only attempts to connect to the next IP address associated with the domain name entry.

When a user on a non-Xyplex unit, such as a UNIX host, tries to make a connection to the rotary through a domain name, the unit makes the connection based on its own rules. UNIX implementations vary. Some hosts only attempt to connect to the first IP address associated with the domain name. Others go down the list in order.

Configuring the Rotary

Use the following commands to set up a rotary. By default, the server's IP address is mapped to all ports (Port 0 - Port *n*).

```
DEFINE/SET SERVER INTERNET ROTARY ip-address port-list  
DEFINE/SET SERVER INTERNET ROTARY domain-name port-list
```

The following sections explain how to set up rotaries for sample applications.

Simple Rotary on a Single Server

Users can connect to this type of rotary through either an IP address or domain name. To connect eight server ports to a host that is not connected to the Ethernet network, use this command:

```
Xyplex>> define server internet rotary 192.168.11.10 1-8  
Xyplex>> define port 1-8 telnet remote 23
```

The server's address, 192.168.11.10, can only be used at a single server, even if ports at another server physically connect to the same device or offer the same type of service.

Connections to the Rotary through a Domain Name

To enable users to connect to the rotary through a domain name, you must map the domain name to the appropriate IP address in the host name tables at your domain name server (e.g., `\etc\hosts`).

If you are using a Xyplex server to resolve domain names (map them to IP addresses), add a locally defined domain name to that server's domain name database. For example, to assign the domain name "host.xyplex.com" to a rotary that has the IP address 192.168.11.10, you would use these commands:

```
Xyplex>> define domain host.xyplex.com 192.168.11.10
Xyplex>> set domain host.xyplex.com 192.168.11.10
```

Rotary on Multiple Servers

Configuring a rotary that is located on two or more servers is similar to configuring a single-server rotary. You assign an IP address, which is unique among all devices on the network, to the ports on each server in the rotary. You then define a single domain name that maps to *all* of these IP addresses. (When using an access server to resolve domain names, the domain name can map to a maximum of 16 IP addresses.)

Domain Name Storage

To use the Rotary Connection feature, you must store domain names in a database. When the server performs a domain name query (requests information from the domain name server), it stores all answers for a given domain name in its operational database. The software can store multiple addresses for a domain name, in these ways:

- Multiple addresses in a single response from a domain name server.
- Multiple addresses from different domain name servers (responses to a broadcast domain name query) — the server saves all answers.
- Multiple addresses can be defined in the server's databases through the Define/Set Domain commands.

The server can store up to 100 domain names. Each domain name is limited to 16 IP address entries.

Clear/Purge Server Internet Rotary

Use the following command to remove an existing rotary:

```
CLEAR/PURGE SERVER INTERNET ROTARY [entry]  
                                     [ALL]
```

An *entry* corresponds to an entry number that appears in the Show/List/Monitor Internet Rotary display. (See Figure 7.)

Show/List/Monitor Server Internet Rotary

Use these commands to view all currently available port IP address assignments (but not domain names). Figure 7 shows a sample display:

Xyplex> SHOW SERVER INTERNET ROTARY	
Internet Address	Ports
172.19.245.110	1-8

Figure 7 - Server Internet Rotary Display

Show/List Domain

This display includes an Entry field, and shows multiple entries for a single domain name, when applicable. (See [Figure 12](#) on page 90.)

Clear/Purge Domain

Use these commands to delete one or all entries for a single domain name. (Refer to [page 90](#).)

Configuring RLOGIN Support

The RLOGIN feature enables a user to log onto a host system through a server port, as shown in Figure 8:

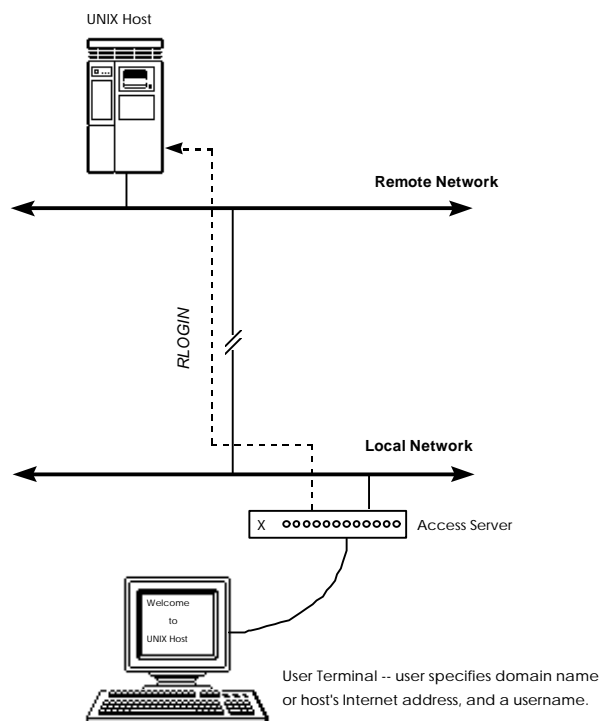


Figure 8 - Connecting to Host through RLOGIN

The user enters the domain name or IP address of the host system, and a username that the host recognizes. The server passes its IP address to the host, along with the username of the port and the username entered on the RLOGIN command line.

If the user did not enter a username on the RLOGIN command line, the server uses the username of the port. Depending on the RLOGIN implementation at the UNIX host, this might be enough to allow the user to bypass the host's login routine.

Considerations

To set up the RLOGIN feature on the host, you modify certain files. For example, on some UNIX hosts, you include an entry in the `/etc/hosts.equiv` file and, optionally, each user's `.rhosts` file. Then, when a user attempts to login to an account — using RLOGIN and a username that matches an entry in the `etc/hosts.equiv` file — that user is automatically logged on to the host. The user is not prompted for a username and password.

The RLOGIN feature is enabled by default on the access server. You might not want to use the RLOGIN feature with sensitive accounts, however, since anyone who knows the right username can log on to the account. You can disable RLOGIN through this command:

```
DEFINE SERVER RLOGIN DISABLED
```

Associated Commands

The access server supports RLOGIN through these commands:

```
DEFINE SERVER RLOGIN ENABLED/DISABLED
```

This command specifies whether users can make connections through RLOGIN. The default is ENABLED.

```
RLOGIN
```

Log on to a host by specifying the host system and a username.

```
SHOW/MONITOR SESSION
```

View information about an RLOGIN session.

Using TCP/IP Features

SHOW PORT x STATUS

Indicates “RLOGIN” in the Current Service field and “Port 513” in the Current Port field when a port is being used for RLOGIN.

Network Management

This section provides an overview of the access server's network management features. The software supports network management in these ways:

- Simple Network Management Protocol (SNMP) support
- Telnet access to a server's console port

Using SNMP

This section explains how to use SNMP to manage Xyplex Access Servers. It assumes that you plan to use an SNMP-based network management software application, such as Xyplex ControlPoint™, to manage the server(s). Hereafter, this section refers to your network management application as a Network Operations Center (NOC).

SNMP is an Internet protocol, defined by RFC (Request for Comment) 1157, that specifies how network management information is carried through a network. The access server software implements an SNMP *agent*. The server stores Management Information Base (MIB) data, and makes it available when requested through SNMP.

The software supports standard and Xyplex-proprietary MIB objects. The software agent accepts the GetRequest, GetNextRequest, and the SetRequest functions and supports the following SNMP traps. These are notices that can be sent to an SNMP client, indicating that a specific event has occurred or the condition of a unit has changed.

Xyplex Acces Servers can generate the following SNMP traps:

coldStart	Signifies that the server is booting in a way that indicates that its configuration might have been changed. The access server generates this trap immediately after loading parameters.
authenticationFailure	Signifies that the server is the addressee of an improperly authenticated network protocol message. SNMP community name and client authentication failures cause the server to generate this trap.

One trap is generated for each occurrence of its defined event. Note that no traps are transmitted if the server has not been assigned an IP address.

Figure 9 shows an example of SNMP in an extended network:

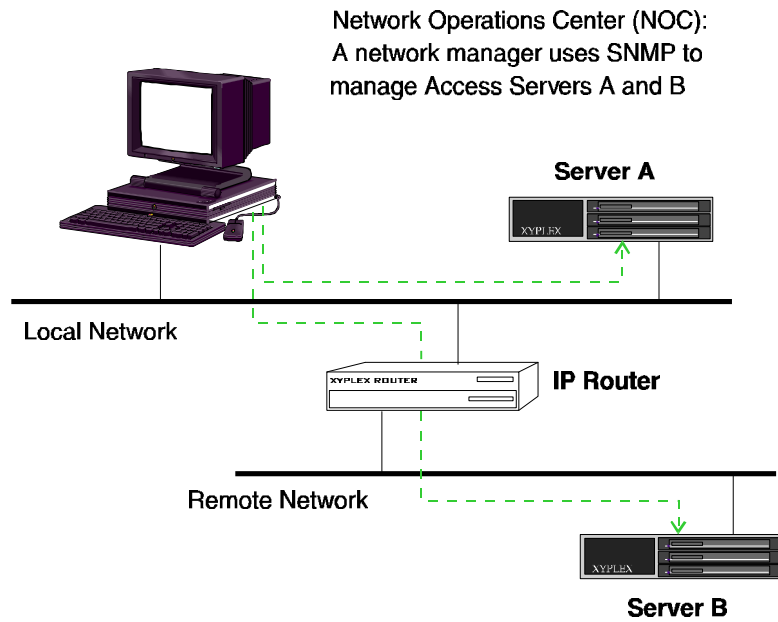


Figure 9 - Network Management through SNMP

The server stores the following groups of MIB data:

System — Describes the system as a whole.

Interfaces — The network interfaces over which the system can send or receive IP datagrams (messages).

Address Translation — The Address Resolution Protocol (ARP) table, which the server uses to match IP addresses to Ethernet addresses.

IP — The operation of the Internet Protocol.

ICMP — The operation of the Internet Control Message Protocol.

TCP — The operation of the Transmission Control Protocol.

UDP — The operation of the IP User Datagram Protocol.

Obtaining/Importing the Supported MIBs

If you are using the Xyplex ControlPoint software, you do not need to import any MIBs. The ControlPoint software package includes all of the MIBs that you need.

If you are not using ControlPoint, you can use SNMP to manage all of the access server's settings — provided that your NOC is capable of compiling proprietary MIBs. The Xyplex-proprietary MIBs are available through the Xyplex World Wide Web (WWW) server: <http://www.xyplex.com>.

In addition, a MIB kit is available from Xyplex, which contains all standard and Xyplex-proprietary MIBs that Xyplex products support. The kit also includes a text file that lists the MIB groups supported by the access server's SNMP agent, and provides useful information about how the agent handles MIB objects.

To obtain the MIB kit, contact your Xyplex sales representative or distributor.

Defining a Trap Client

An access server will not generate any SNMP Trap messages until you define a Trap Client. A Trap Client is a specific NOC to which the access server sends Trap messages. You can define one or more Trap Clients through this command:

```
DEFINE/SET SERVER INTERNET SNMP TRAP CLIENT [index] ip-address
```

An *index* value is a number from 1 to 4. The *ip-address* identifies the NOC that should receive the Trap messages.

Example

```
Xyplex>> define server internet snmp trap client 1  
172.18.12.3
```

Assigning SNMP Security Information (Optional)

By default, a Xyplex Bridge/Router accepts SNMP Get, Get_Next, and Set requests from any NOC. You can restrict SNMP access to the bridge/router by defining SNMP Clients and Communities. A Community refers to one or more NOCs that specify the same Community string in their SNMP messages. A Client is a specific NOC, which you identify through an IP or Ethernet address.

Table 9 shows how Get/Set Client and Community values determine which NOCs can manage an access server.

Table 9 - Get/Set Client and Community Settings

Get/Set Client	Get/Set Community	NOCs that Can Issue Get/Set Requests
None	None	Any
None	Defined	Any NOC that knows the Community value
Defined	None	NOCs defined as Clients
Defined	Defined	NOCs defined as Clients, which know the Community value

Defining Get and Set SNMP Clients

A Get Client is a specific NOC that is allowed to manage the access server through Get and Get_Next requests. A Set Client is a NOC that may issue Set requests to the access server. You can use the following commands to define up to four of each of these client types:

```
DEFINE SERVER INTERNET SNMP GET CLIENT [index] ip-address
```

```
DEFINE SERVER INTERNET SNMP SET CLIENT [index] ip-address
```

An *index* value is a number from 1 to 4. Use the keyword NONE to delete a previously defined Get or Set Client.

NOTE: Be sure to define Get and Set Client entries for your NOC before you define any other Get or Set Clients.

Example

```
Xyplex>> define server internet snmp set client 1
172.18.121.3
```

Defining SNMP Communities

Get and Set Communities provide an additional level of security. If you do not define any Get Clients, the access server will accept Get and Get_Next requests from any NOC whose Get requests include a Community name that matches the server's Get Community. If you do not define a Get Community, the access server will accept Get and Get_Next requests from any NOC.

Similarly, if you do not define any Set Clients, the access server will accept Set requests from any NOC whose requests include a Community name that matches the server's Set Community. If you do not define a Set Community, the access server will accept Set requests from any NOC.

If you define a Trap Community, the access server will include the Trap Community name in the Trap messages that it generates.

Use the following commands to define Get, Set, and Trap Community names:

```
DEFINE SERVER INTERNET SNMP GET COMMUNITY [ "community-name" ]  
[ NONE ]
```

```
DEFINE SERVER INTERNET SNMP SET COMMUNITY [ "community-name" ]  
[ NONE ]
```

```
DEFINE SNMP TRAP COMMUNITY "community-name"
```

A *community-name* can include up to 23 characters. Do not include spaces. Use the value NONE to delete a previously defined Get or Set Community.

Examples

```
Xyplex>> define server internet snmp get community none
Xyplex>> define server internet snmp set community "xyplex"
```

Miscellaneous SNMP Settings

This section explains how to define SNMP Contact and Location strings.

Contact

An SNMP contact identifies a person to contact when the access server needs attention. Use this command to define a contact:

```
DEFINE SERVER INTERNET SNMP CONTACT "contact-string"
```

The "*contact-string*" can include up to 60 characters; do not include spaces.

Example

```
Xyplex>> define server internet snmp contact "bobby_jones"
```

Location

An SNMP Location specifies where the access server is located. Use this command to specify a location:

```
DEFINE SERVER INTERNET SNMP LOCATION "location-string"
```

The "*location-string*" can include up to 60 characters; do not include spaces.

Example

```
Xyplex>> define server internet snmp location "closet_1"
```

Using Telnet to Access the Console Port

You can connect to the console port (Port 0) of an access server through Telnet. Once connected to Port 0, you have access to the server's command interface, and can issue server commands as though you were directly logged on to the server. Figure 10 shows two Telnet connections:

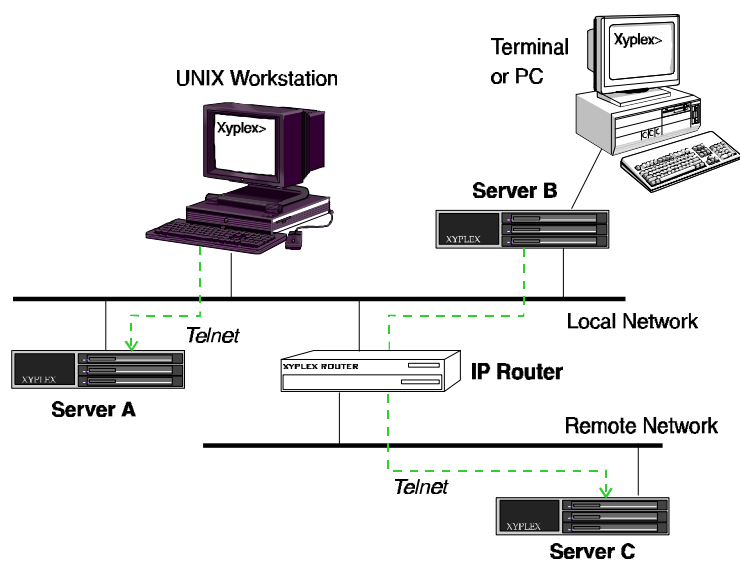


Figure 10 - Telnet Connections to Console Ports

Once you have connected to the console port of a server, you can use Telnet to connect to another server's console port. Using Figure 10 as an example, you can connect from the console port of Server A to the console port of Server B or C.

You can suspend a Telnet session with the console port of one server by hitting the <Break> key, or typing the local switch character, and then open a session with the console port of another server. You can then suspend that Telnet session and resume the original session, or open a session with the console port of a different server.

Telnet Console Command

Use these commands to access the console port (Port 0) of an access server through Telnet:

```
TELNET CONSOLE ip-address
TELNET CONSOLE ip-address:telnet-port-number
TELNET CONSOLE domain-name
TELNET CONSOLE domain-name:telnet-port-number
```

Use the following command to access the console port from a UNIX host, through Telnet. Note that you do not need to specify 2000 if you have previously issued the command `DEFINE PORT 0 TELNET REMOTE 23`.

```
TELNET [ip-address] 2000
        [domain-name]
```

The following port settings are predefined for the access server's console port. You cannot change the values for these settings:

Characteristic	Setting
ACCESS	LOCAL
AUTOBAUD	DISABLED
BREAK	DISABLED
CHARACTER SIZE	8
DEDICATED SERVICE	NONE
DIALUP	DISABLED
DSRLOGOUT	DISABLED
DTRWAIT	DISABLED
FLOW CONTROL	XON
INPUT FLOW CONTROL	ENABLED
INPUT SPEED	9600
MODEM CONTROL	DISABLED
OUTPUT FLOW CONTROL	ENABLED
OUTPUT SPEED	9600
PARITY	NONE
SPEED	9600

Examples:

```
Xyplex> telnet console 192.168.2.30
```

Establish a session between this port and the console port of the Telnet destination whose IP address is 192.168.2.30.

```
Xyplex> telnet console access1.xyplex.com
```

Establish a session between this port and the console port of the Telnet destination whose domain name is access1.xyplex.com.

Define/Set Server Console Logout

Use the following command to specify whether the server should immediately disconnect a console port session when a user logs out from the port. This setting is enabled by default.

```
DEFINE/SET SERVER CONSOLE LOGOUT [ENABLED]  
                                   [DISABLED]
```


Loading through Internet Protocols

The access server supports three Internet protocols for loading software and parameters:

- Bootstrap Protocol (BOOTP)
- Trivial File Transfer Protocol (TFTP) and Directed TFTP (DTFTP)
- Reverse Address Resolution Protocol (RARP)

These network protocols enable the server to obtain load images (software files) and parameter files from UNIX hosts and to use the UNIX hosts as dump servers. Support for these protocols also enables servers to receive load images and parameters, and to dump memory contents, through an IP router.

These capabilities are illustrated in Figure 11.

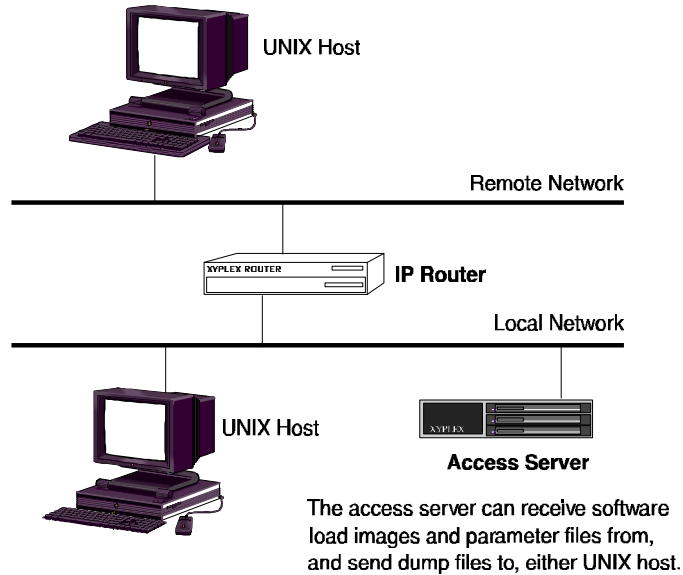


Figure 11 - Loading through Internet Protocols

The server uses BOOTP to find:

- The server's IP address
- The IP address of a node that can serve as a load, dump, or parameter server
- The name of the file to load
- Potential parameter servers

The server uses TFTP to:

- Copy load images and parameter files to the server
- Write parameters to parameter servers
- Dump memory

Loading Images and Parameter Files

Network 9000 Access Server 720 modules and MAXserver 1620 and 1640 Access Servers have three *initialization records*. These records define how and where the server can obtain a load image and parameter file. The ROMs in these servers use Initialization Record 1 to search for a load image and parameter file. If this attempt fails, they use Initialization Record 2. If the second attempt fails they try Initialization Record 3.

By default, these servers look for a load image on the flash card if a card is present in the card slot. Access Server 720 modules can also obtain a parameter file from a flash card. MAXserver 1620/1640 servers attempt to obtain parameters from NonVolatile Storage (NVS), using the NVS protocol by default. If these default protocols are not enabled, or a flash card is not present in the card drive, the ROMs will search the network for a load image and parameter file.

Refer to the manual *Managing Network 9000 Modules and Power Supplies* for more information about initialization records on Access Server 720 modules. Refer to the *Getting Started Guide* for MAXserver 1620 and 1640 Access Servers for more information about the initialization records on these units.

Configuring Load Protocols

Use the Initialization Configuration Menu, which is described in the hardware documentation supplied with your access server, to select one or more of the following load protocols:

- CARD protocol (Access Server 720 modules for load image and parameter file; MAXserver 1620 and 1640 for load image only)
- NVS protocol (MAXserver 1620/1640 Access Servers for parameter file only)
- Xyplex protocol (XMOP)
- Maintenance Operations Protocol (MOP)
- Bootstrap Protocol (BOOTP)
- Reverse Address Resolution Protocol (RARP)
- Directed TFTP (DTFTP)

Directed TFTP

You can designate a specific TFTP load server by using directed TFTP. You configure the related settings through the Initialization Configuration Menu, or through the commands described in “[DTFTP Protocol](#)” on page 211. Refer to the hardware documentation supplied with your server for a description of the Initialization Configuration Menu.

Eliminating TFTP Broadcasts

Servers that are configured to store parameters locally do not broadcast requests for parameter servers through TFTP or any other protocol. Only servers that are configured for remote parameter storage do so.

You can eliminate TFTP broadcasts that occur while the server is running by disabling TFTP, or by disabling all parameter server checks through the Server Parameter Server Check setting. (Refer to [page 86](#).)

Alternately, rather than broadcasting a TFTP message and waiting for responses, you can create a static list of parameter servers through the Define/Set Parameter Server command. Each parameter server in the list will be queried, through TFTP, to determine whether a parameter file exists, what version it is, and other related information.

To eliminate TFTP broadcasts during loading, you can configure the server to load through the Xyplex proprietary protocol (XMOP), MOP, or directed TFTP.

Saving Parameters in the Permanent Database

The access server supports the XMOP, MOP, TFTP and Xyplex protocols for parameter saving. The server always attempts to save parameters through XMOP first. If the Server Parameter Server Check setting is enabled (refer to [page 86](#)), the server also attempts to save parameters through TFTP.

If a parameter server uses the TFTP protocol, the parameters are written into a `.prm` file. If this operation is successful, the parameters are also written to a `.bck` (backup) file.

Dump Transmission

When the access server dumps its memory contents, it can use the same protocols as it does for loading, and the process for selecting a protocol is the same.

Define/Set Parameter Server

Use this command to define a parameter server, using either an Ethernet address or an IP address:

```
DEFINE/SET PARAMETER SERVER name ADDRESS ethernet-address  
DEFINE/SET PARAMETER SERVER name INTERNET ADDRESS ip-address
```

Example

```
Xyplex>> define parameter server unix-host internet address  
172.18.1.1
```

Define/Set Server Parameter Server Check

Use these commands to enable/disable the Parameter Server Check setting, or to specify how the access server locates and updates eligible parameter servers. *This setting is enabled by default.*

```
DEFINE/SET SERVER PARAMETER SERVER CHECK [ENABLED]  
[DISABLED]  
[PROPRIETARY ENABLED]  
[TFTP ENABLED]  
[TIMER timer-value]
```

Use the TFTP ENABLED option to instruct the server to use TFTP *only* for parameter serving. The PROPRIETARY ENABLED keyword instructs the server to use Xyplex-proprietary protocols only.

The **TIMER** setting specifies how often the access server attempts to locate additional usable parameter servers. You can specify a value from 1 - 120 (minutes). The default is 30 (minutes).

Show Server Status Display

This display shows the IP Load and Dump addresses when the server uses the BOOTP/TFTP protocol for loading and dumping.

Using the Server as a Domain Name Server

The access server provides limited domain name server support. Though the server cannot function as a general Domain Name Server for your network, it can resolve domain names (map them to IP addresses) when it is directly asked to do so. To use this feature, you define the access server as a primary or secondary name server for a host or for another server.

Typical applications include:

- Using a server as the primary or secondary domain name server for other access servers in the network.
- Using a server to resolve domain names on a small network that does not use a host as a domain name server.
- Using an access server to resolve domain names through the rotary feature.

Domain Name Resolution

Upon receiving a request to resolve a domain name, the server checks the domain name table in its operational database. The server sends a response (indicating the IP address[es] associated with the domain) for any domain names that appear in its database.

Obtaining/Storing Domain Names

Domain names in the permanent database are entered into the operational database whenever the server boots. You can use the Purge Domain command to remove a domain name from the permanent database.

In addition to the locally defined domain names, which you specify through Define/Set Domain commands, the server can use domain names that it learns from domain name servers in the network. The server enters these domain names *into the operational database only*.

The operational database can hold up to 100 learned and local domain name/IP address combinations. You can assign each domain name up to 16 IP addresses, to form a rotary group. However, if you are using a domain name server, you should not specify more than 99 domain name/IP address combinations, or the server will not be able to learn any domain names.

The server keeps a learned domain name in its operational database until one of the following occurs:

- You remove it with a Clear Domain command.
- The time-to-live period assigned by the domain name server expires. The software limits the time-to-live period to the value specified by the Server Internet Domain TTL setting.
- The operational database contains the maximum number of domain names, a user adds a new domain name with a Set Domain command, or the server learns a new domain name from a domain name server. In this case, the server replaces the oldest learned domain name in its operational database with the new name.

Domain Name Time-to-Live (TTL)

Each IP address for a domain name has its own “time-to-live.” The first time a user attempts a connection (or a ping) to a domain name, the server requests a translation from the domain name server(s). Once it receives an answer, the server does not request another translation until the time-to-live expires for all entries associated with that domain name. You specify the time-to-live value of domain name responses through this command:

```
DEFINE/SET SERVER INTERNET DOMAIN TTL time-to-live
```

You can define a *time-to-live* from 0 - 168 hours.

Example

```
Xyplex>> define server internet domain ttl 24
```

Define/Set Domain

Use this command to define a domain name entry:

```
DEFINE/SET DOMAIN domain-name ip-address
```

Example

```
Xyplex>> define domain john.xyplex.com 192.168.1.1
```

Show/List Domain

Use this command to view the domain names currently stored in the server database. Figure 12 shows a sample display. The server resolves requests *only* for the domain names listed.

Internet		Domain		10 May 1993 14:09:51	
Entry	Address	TTL	SRC	Name	
2	140.179.139.254	47	Pri	FINANCE.SUN.COM	
1	140.179.20.1	49	Pri	MINX.XYPLEX.COM	
3	140.179.20.1	49	Pri	XEBRA.XYPLEX.COM	

Figure 12 - Show Domain Display

Clear/Purge Domain

Use these commands to delete one or all locally defined domain names:

```
CLEAR/PURGE DOMAIN [domain-name]
                    [ENTRY entry-number]
                    [ALL]
```

An *entry-number* refers to an entry number listed in the Show Domain display (see Figure 12).

Using IP Reassembly

Sometimes data packets that are forwarded by a gateway or a router become fragmented (broken into pieces). When this happens, the server can either attempt to reassemble the packet fragments, or it can simply discard them. Packet fragmentation can cause problems, particularly when using network protocols that do not include a re-send mechanism (for example, UDP, which does not guarantee that data is received successfully).

If packet fragmentation is a problem, you can enable the Server Internet IP Reassembly setting, thereby allowing the server to attempt to reassemble fragmented packets. (There is no guarantee that this operation will be successful, however). Use this command:

```
DEFINE/SET SERVER INTERNET IP REASSEMBLY  [ENABLED]  
                                              [DISABLED]
```

IP reassembly uses additional memory, because the server must store all fragments until it can reassemble the complete packet. Therefore, you might need to increase the Server Packet Count setting through this command:

```
DEFINE SERVER PACKET COUNT packet-buffers
```

You can define a *packet-buffers* value from 80 to 1088. (*Default = 80.*) If the server has limited memory, or if packet fragmentation is not a frequent problem, you should leave the Server IP Reassembly setting disabled (as it is by default).

Using TCP Resequencing

Sometimes, when a host has a lot of data to send to the access server, it divides the data into many smaller packets. Each packet is transmitted in sequence, along with a sequence number. Occasionally, a packet is delayed during transmission, usually by a node between the host and the access server. This can cause packets to arrive out of sequence.

When a server receives packets out of sequence, it can either discard the data and not acknowledge receiving it, or it can collect the packets and wait until the out-of-sequence packets arrive. It then passes on the data in the proper sequence. This process is called TCP resequencing.

If the server does not acknowledge the data, the host retransmits all the information again.

Use the following command to enable TCP resequencing. *It is disabled by default.*

```
DEFINE SERVER INTERNET TCP RESEQUENCING [ENABLED]  
[DISABLED]
```

TCP resequencing uses additional memory, because the server must collect all packets until the out-of-sequence packets arrive. Therefore, you might need to increase the Server Packet Count setting through this command:

```
DEFINE SERVER PACKET COUNT packet-buffers
```

You can define a *packet-buffers* value from 80 to 1088. (The default is 80.) If the server has limited memory (e.g., less than 180 KB after all features are enabled), you should leave the Server Internet TCP Resequencing setting disabled (as it is by default).

Chapter 4

Setting Up TN3270 Terminals

This chapter describes the following tasks:

- Enabling the TN3270 Protocol
- Defining a TN3270 Device
- Defining a TN3270 Translation Table
- Alternate Keymaps
- Using Line 25 as a Status Line
- Local Printer Support

Xyplex Access Servers support TN3270, which enables server users to communicate with an IBM host over a LAN, using Telnet. While connected to the IBM host, users can access databases or run host-based applications. The server translates key input from the terminal into an IBM 3270 data stream. By doing so, the server enables the users' terminals to emulate an IBM display station.

The access server software can emulate the IBM 3278 Model 2 and Model 5 display stations. The server sends the 3270 data stream over the LAN to a Telnet Server at the IBM host site. Similarly, the server translates the 3270 data stream from the Telnet Server into data that the users' terminals can understand.

An access server user can have other, non-TN3270 sessions active while a TN3270 session is active.

Translation Tables

The access server supports several device types and provides an U.S. English translation table. However, your site might require different devices and tables. To create a new device or translation table, you make a copy of the information in an existing device or table, rename it, and then modify the information.

When you create a new device or translation table, you use Define commands only. The server saves devices and tables in its permanent database only; not in the operational database.

You must assign a TN3270 device to each port to ensure proper terminal emulation during a TN3270 session. If you do not want the default translation table (USEENGLSH), you must also define the translation table.

Enabling the TN3270 Protocol

The TN3270 protocol is disabled by default. To enable it, you must enter a software password or “key.” Check with your Xyplex sales representative or distributor if you do not know the TN3270 password. Use this command to enable TN3270:

```
Xyplex>> define server protocol TN3270 enabled
```

When you enter this command, the server prompts you for a password. Enter the password (key) to enable the protocol:

```
TN3270 Password> xxxxxxxx
```

The password does not appear on the screen when you type it. After entering the password, reboot the server to activate the protocol.

After enabling the protocol, you must assign TN3270 device types and translation tables to server ports. These steps are described in the following sections.

NOTE: Some TN3270 features, such as color support, require that the server run the enhanced software for multi-MB servers. Servers with less than 2 MB of memory do not support these features. The access server software *Release Notes* explain how to obtain the enhanced software.

Enabling Extended Attributes

Certain TN3270 features, including the screen attributes blinking, reverse video, and underscore, and extended color support, require that you enable the extended attributes feature on individual ports. Use this command to do so:

```
DEFINE PORT port-list TELNET TN3270 XDATTRS [ENABLED]  
[DISABLED]
```

Defining TN3270 Devices

TN3270 device tables contain the following categories of information, which the server uses to emulate IBM 3270 display stations:

TN3278TYPE — The model of TN3270 device that the server emulates during a TN3270 session.

Local Terminal Type — The default terminal types included with the server software are ANSI, VT100, VT220-7, and VT220-8.

Keymap — Contains the escape sequences that the server uses to translate users' keyboard entries into 3270 display station functions.

Screenmap — Contains the escape sequences that the server sends to the users' terminals to control screen functions such as clear the screen, move the cursor, or set the bold attribute. Optionally, you can assign screenmap color modes. The mode you use, if any, depends on the colors that your terminal supports as well as the colors that the IBM host application supports.

Use the commands described in this section to create a new device using information from an existing device, and to then modify the information for the new device. All commands begin with the keywords `DEFINE SERVER TN3270 DEVICE`.

Xyplex has defined terminal keymaps and screenmaps for the Wsye 50, IBM 3162, IBM 3164, QVT 82, Televideo (905, 910, 925), and VT 330. You can obtain them by contacting Xyplex Customer Support.

Creating a New Device Type

Use the following command to create a new device type, based on an existing device type. You can specify an existing device type, or specify the number of a port where the device has been set up. The server can use that port's device type and keymap to create the new one. The server can maintain up to eight different TN3270 device types.

```
DEFINE SERVER TN3270 DEVICE new-device [CREATE existing-device]  
[PORT port-number]
```

Valid *existing-device* values are ANSI, VT100, VT220-7, and VT220-8

Examples

```
Xyplex>> define server TN3270 device tv925 create vt100  
Xyplex>> define server TN3270 device tv925 port 6
```

Using the TN3270 Command

To log on to a TN3270 host, a user can enter the TN3270 command, followed by a host name or IP address. The access server then displays a list of the currently defined terminal (device) types. The user selects a device type by entering the corresponding number, and the server attempts to connect to the host using that device type. An example follows:

```
Xyplex> tn3270 172.19.2.101  
  
Select TN3270 Device Type  
  
1.  ANSI  
2.  VT100  
3.  VT220-7  
4.  VT220-8
```

Setting Up TN3270 Terminals

NOTE: Whether you use the TN3270 command or the Telnet command to initiate the TN3270 session, three Telnet options must be negotiated successfully between the client and server: TerminalType, Binary, and EOR.

Defining a TN3270 TERMINALTYPE

Use the following command to specify the local terminal type that the server uses during TN3270 emulation:

```
DEFINE SERVER TN3270 DEVICE device-name TERMINALTYPE "termtype"
```

The *device-name* can be a device that you have defined, or a device supplied by the server. The “*term-type*” describes the device; it can comprise up to 21 characters.

Example

```
Xyplex>> define server TN3270 device tv925 terminaltype  
          "Televideo-925"
```

NOTE: During Telnet negotiations, the “termtype” string is sent from the client to the server as part of the Terminal Type sub-negotiations.

Also, in a separate Terminal Type sub-negotiation, the TN3278 model is transmitted and appears as “IBM-3278-2” or “IBM-3278-5” in the TCP/IP packet. (Refer to [Defining the TN3278TYPE](#) on page 99.)

Defining the TN3278TYPE

Use this command to specify the type of IBM display station that the server emulates on the local terminal during a TN3270 session:

```
DEFINE SERVER TN3270 DEVICE device-name TN3278TYPE model
```

The server uses the *device-name* that you specify to emulate the display station *model* that you designate: either MODEL2 (24 rows x 80 columns) or MODEL5 (27 rows x 132 columns).

Example

```
Xyplex>> define server tn3270 device tv925 tn3278type  
model2
```

Modifying the Keymap

Use the following command to modify entries in the Keymap portion of the TN3270 device information:

```
DEFINE SERVER TN3270 DEVICE device-name KEYMAP key  
"escape-seq" "description"
```

The *device-name* specifies the device to which the new keymap escape sequence is to be applied.

The *key* specifies the IBM 3270 display station function to be executed when a user enters the key sequence that you specify through the “*escape-seq*” variable. [Table 10](#) on page 101 lists possible values for the *key* variable.

Setting Up TN3270 Terminals

The “*escape-seq*” is a byte sequence to be mapped to the IBM display station function in the *key* variable. You can specify the characters in two ways: (1) by entering their hexadecimal values, which you can obtain from the Programmer's Reference manual for the terminal, or (2) by manually pressing the keys on the terminal. This variable can include from 0-8 hexadecimal values. Enclose it in quotes.

The “*description*” is a 1 to 5 character string that identifies the escape sequence in various keymap displays, such as the Show Port Keymap and Show Server TN3270 displays. The description also appears in the display that appears when a user presses the SHOWKEYS status key during TN3270 terminal emulation.

Example

```
Xyplex>> define server TN3270 device tv925 keymap PF1  
          "01 40 13" "F1"
```

NOTE: If the Port Telnet Newline Filtering setting is “None,” causing a CR/NULL to terminate lines, you must define the keymap for Enter as “0D 00 ”. The trailing space after 00 is *required*.

IBM 3270 Display Station Functions for TN3270 Keymap

Keymaps contain the escape sequences from users' terminals that emulate IBM display station functions. Table 10 lists the IBM display station functions that a local key sequence emulates during a TN3270 session:

Table 10 - IBM Display Station Functions

AO	Erase Input	PF4	PF19
AYT	Fast Left	PF5	PF20
Back Tab	Fast Right	PF6	PF21
Break	Field Mark	PF7	PF22
CentSign	Flush Input	PF8	PF23
Cursor Down	Home	PF9	PF24
Cursor Left	Insert Mode	PF10	Print
Cursor Right	IP	PF11	Refresh
Cursor Up	New Line	PF12	Reset
Clear	PA1	PF13	Scroll
Cursor Sel	PA2	PF14	ShowKeys
Delete	PF3	PF15	Status ON/OFF
Dup (Duplicate)	PF1	PF16	Sys-req
Enter	PF2	PF17	Tab
Erase EOF	PF3	PF18	Test
DevCncl			

NOTE: The Abort Output (AO), Are You There (AYT), Break, and Interrupt Process (IP) functions require the enhanced server software. These Telnet commands generate FF F5 (AO), FF F6 (AYT), FF F3 (Break), or FF F4 (IP) in a datastream to the Telnet server on the host. The Telnet server must be able to interpret these functions.

The Scroll Function

To display the complete 3270 model 5 screen, (27 rows x 132 columns) on terminals that can support 27 lines (or 28 for a separate status line), set the Keymap Scroll function to null, as follows:

```
DEFINE SERVER TN3270 DEVICE device-name KEYMAP SCROLL " "
```

Device Cancel

The Device Cancel (DevCncl) keymap command enables TN3270 users to cancel a queued print request. The keyboard of a user who executes a local screen print is locked up until the screen print completes. If the print request is queued because the printer is busy, the user might have to wait a long while before keyboard operation is restored. (The status line displays "PrtBusy X" when this happens.) By typing the Device Cancel key sequence, the user can cancel the print request and unlock the keyboard,

```
DEFINE SERVER TN3270 DEVICE device-name KEYMAP DEVCNCL  
"escape-sequence" "description"
```

Modifying the Screenmap

Use this command to modify entries in the Screenmap:

```
DEFINE SERVER TN3270 DEVICE device-name SCREENMAP map-action
```

Valid *map-actions* include:

```
action "escape-seq"  
MOVECURSOR "escape-seq" [BASE value]  
SGR ENABLED/DISABLED
```

The *device-name* specifies the device to which the new escape sequence is applied.

The *action* variable specifies the action associated with the command. [Table 11](#) on page 104 lists valid *actions*. (Note that when you specify MOVECURSOR as the screenmap action, you can optionally specify an offset value [BASE *value*] for the row and column positions, which differs from the default value of 1.)

The “*escape-sequence*” is a hexadecimal value that corresponds to the *action* that you specify. Refer to your terminal’s Programmer’s Reference manual to obtain the value. Enclose the “*escape-sequence*” in quotes.

The BASE *value* specifies the offset for the MOVECURSOR screenmap action. This *value* specifies the row and column position for the cursor. Valid values for this variable are 0-255. The default is 1.

SGR specifies how the server implements the bold, blink, and underscore screen attributes. If you enable this setting, the program uses the Set Graphic Rendition command to support these attributes. If you disable this setting, the program uses ON/OFF escape sequences to support the attributes.

Setting Up TN3270 Terminals

Example

```
Xyplex>> define server TN3270 device tv925 screenmap  
clearscr "1B 2A"
```

Table 11 - Screenmap Actions

Action	Description
BEEP	Produce a sound beep.
BLINKOFF	Set blink mode off.
BLINKON	Set blink mode on.
BOLDOFF	Set bold intensity off.
BOLDON	Set bold intensity on.
CHARSET	Set the character set.
CLEARSCR	Clear the screen.
COL132	Set 132 column mode.
COL80	Set 80 column mode.
COLORBLUE	Set ANSI-standard for blue
COLORGREEN	Set ANSI-standard for green
COLORPINK	Set ANSI-standard for pink
COLORRED	Set ANSI-standard for red
COLORTURQUOISE	Set ANSI-standard for turquoise
COLORWHITE	Set ANSI-standard for white
COLORYELLOW	Set ANSI-standard for yellow
ERASEEOL	Erase from cursor to the end of the line.
EXITRESET	Series of commands used to reset the terminal upon disconnecting. The command(s) execute after the Reset 1 - 4 actions. Refer to page 108 .
MOVECURSOR	Position the cursor.
RESET1	Series of commands used to reset the terminal (part 1).
RESET2	Series of commands used to reset the terminal (part 2).

RESET3	Series of commands used to reset the terminal (part 3).
RESET4	Series of commands used to reset the terminal (part 4).
NOTE: The Reset 1 - 4 commands execute during TN3270 initialization and upon termination of a session.	
REVERSEOFF	Set reverse video off.
REVERSEON	Set reverse video on.
SGR	Set graphic rendition
STATUS1	Write to the 25th line (part one).
STATUS2	Write to 25th line (part two).
UNDERSCOREOFF	Set the underscore off.
UNDERSCOREON	Set the underscore on.

The standard escape sequences can include as many as nine hexadecimal values. The STATUS line and RESET strings, however, might require more room. These can consist of several segments, each containing up to fifteen hexadecimal values.

The server provides a set of special hexadecimal values that indicate where to place row and column values and status line data within an escape sequence. Table 12 shows these characters.

Table 12 - Special Values for Escape Sequences

Hex Value	Means
FE	Binary column
FC	Binary row
FA	Character column
F8	Character row
F6	Status line data in ASCII
F4	Status line data in hexadecimal

Setting Up TN3270 Terminals

For example, the escape sequence for the MOVECURSOR action on a VT100 and its hexadecimal value — including the FA character column value — is:

escape sequence: ESC [row;column H
hexadecimal value: 1B 5B F8 3B FA 4B

The STATUSLINE escape sequence for a CIT 224 and its hexadecimal value, including the F4 status line value in hexadecimal, is:

escape sequence: esc P); 1 N Q data esc \
hexadecimal value: 1B 50 30 3B 31 51 F4 1B 5C

Defining Screenmap Color

The Xyplex TN3270 protocol supports two color modes for IBM terminal emulation: basic 4-color mode and extended 7-color mode. The basic 4-color mode consists of green, red, blue, and white. The extended 7-color mode includes the four basic colors and pink, yellow, and turquoise. The mode you use depends on the colors that your users' terminals support as well as the colors that the IBM host application supports.

You define a screenmap escape sequence for each color that you plan to use. You must also enable the [TENLET TN3270 XTDATTRS](#) setting at each port where you plan to use the extended 7-color mode. (Refer to page 95. This setting requires a multi-MB software image.)

The following command defines color for the TN3270 screenmap. The syntax descriptions show the ANSI-standard escape sequence for each color. Refer to the terminal's documentation if your users' terminals support other escape sequence types.

Setting Up TN3270 Terminals

```
DEFINE SERVER TN3270 DEVICE device-name SCREENMAP  
color-value "escape-sequence"
```

<i>color-value</i>	Description
COLORRED	"1B 5B 33 31 6D" ANSI-standard for red
COLORGREEN	"1B 5B 33 32 6D" ANSI-standard for green
COLORBLUE	"1B 5B 33 34 6D" ANSI-standard for blue
COLORYELLOW	"1B 5B 33 33 6D" ANSI-standard for yellow
COLORWHITE	"1B 5B 33 37 6D" ANSI-standard for white
COLORPINK	"1B 5B 33 35 6D" ANSI-standard for pink
COLORTURQUOISE	"1B 5B 33 36 6D" ANSI-standard for turquoise

Defining an Exit Reset String

The Exit Reset screenmap command sends a hexadecimal command string of up to nine characters to the terminal when a TN3270 session terminates. These characters are sent after any defined RESET strings have been transmitted to the terminal. This feature is useful if you want to disable a function such as Auto Scroll during a TN3270 session and then enable it after the session ends. To do this, you disable the function (using the appropriate command string) in one of the RESET screenmap commands, and then enable it through the Exit Reset screenmap.

```
DEFINE SERVER TN3270 DEVICE device-name SCREENMAP EXITRESET  
"hex-values"
```

Assigning a TN3270 Device to a Port

Use the following command to assign a TN3270 device to a port or group of ports. You can assign a device that you have created, or one of the device types that Xyplex supplies.

When you assign a device to a port, the server automatically assigns the USENGLSH language translation table to that port. If you want to assign a different translation table to the port, refer to the section [“Assigning a Translation Table to a Port,”](#) on page 114.

```
DEFINE/SET PORT port-list TELNET TN3270 DEVICE [device-name]  
[NONE]
```

The value NONE disables the terminal emulation feature at the specified port(s).

Example

```
Xyplex>> define port 4-16 telnet 3270 device tv925
```

Other Commands that Affect TN3270 Devices

The following list briefly describes commands that affect TN3270 devices, but which are described elsewhere in the access server document set:

Command	Function
PURGE SERVER TN3270 DEVICE <i>device-name</i>	Removes the device that you specify from the server database.
SHOW/LIST SERVER TN3270	Displays a list of TN3270 devices and translation tables defined in the server's permanent database.
SHOW/LIST SERVER TN3270 DEVICE <i>device-name</i>	Displays the contents of the device table that you specify.
DEFINE/SET SERVER TN3270 PORT KEYMAPS DISABLED/ENABLED	Specifies whether individual ports can maintain copies of a Keymap.
DEFINE/SET PORT TELNET XTDATTRS DISABLED/ENABLED	Specifies whether the ports that you specify support extended screen attributes during a TN3270 session.
SHOW/LIST PORT TELNET CHARACTERISTICS	Displays the TN3270 device for the ports that you specify.
SET PORT KEYMAP <i>key</i> "escape-seq" "description"/NONE	Assigns an individual copy of a keymap to a port and modifies keymap escape sequences within the keymap. The NONE keyword removes a keymap assignment from a port. The Server TN3270 Port Keymaps setting must be enabled to use this command.
SHOW PORT KEYMAP	Displays the contents of an individual copy of a keymap assigned to a port.

Defining TN3270 Translation Tables

Translation tables contain the information that the server uses to interpret the EBCDIC data from the IBM host and convert it to ASCII data for the terminal, and vice versa. This translation process occurs through sets of tables that include the hexadecimal values for each character in the tables.

When you create a new set of translation tables, you copy an existing one and then enter hexadecimal values for the unique characters in the new tables. Before you begin, make sure that you have the documentation that lists the hexadecimal ASCII and EBCDIC values for the new characters in the table. This includes the *IBM 3270 Information Display System Character Set Reference* and the equivalent reference for the ASCII terminal.

The server includes one set of tables for USENGLSH (American English). You can define up to eight other sets of tables for other languages.

Creating a New Translation Table

To create a new set of translation tables, you first make a copy of a set of existing tables. The first time you do this, the tables that you copy are the Xyplex-supplied USENGLSH tables. (They are the only ones that exist.) Once you create your own translation tables, you can use them as the basis for new tables.

```
DEFINE SERVER TN3270 TRANSLATIONTABLE new-table CREATE  
existing-table
```

The server creates a new TN3270 translation table using the *new-table* name that you specify. The name can comprise up to eight characters. Most translation table names reflect the language of the table; e.g., USENGLSH, FRENCH, SPANISH, and GERMAN.

The server bases the new translation table on the *existing-table* that you specify. If this is the first new table you have created, USEENGLSH is the value you use for this variable.

Once you copy the existing table and rename it, you can begin changing the values in the new table. To do this, you create entries for data to be sent from your terminal to the IBM host (ASCII to EBCDIC translation) and for data to be sent from the IBM host to your terminal (EBCDIC to ASCII).

Entering New Values Into the Table

```
DEFINE SERVER TN3270 TRANSLATIONTABLE trans-name table
offset value
```

The *trans-name* specifies the name of the new translation table. (You can only specify the name of a translation table that you have created. You cannot modify the USEENGLSH table.)

The two *table* values you can use are:

ASCIITOEBCDIC Apply the new value to outgoing data from the user's terminal to the IBM host.

EBCDICTOASCII Apply the new value to incoming data from the IBM host to the user's terminal.

The *offset* variable specifies the value to be modified. The values 40 through FF apply to EBCDICTOASCII, as shown in Table 13. The values 20 through FF apply to ASCIITOEBCDIC, as shown in Table 14.

Setting Up TN3270 Terminals

The *value* variable specifies the new hexadecimal translation table entry. The values 20 through FF apply to EBCDICTOASCII and the values 40 through FF apply to ASCII TO EBCDIC. (This is the reverse of the offset values.)

Table 13 - EBCDICTOASCII, USENGLSH Translation Table

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	20	20	20	20	20	26	2d	20	20	20	20	20	7b	7d	5c	30
1	20	20	20	20	20	20	2f	20	61	6a	7e	20	41	4a	20	31
2	20	20	20	20	20	20	20	20	62	6b	73	20	42	4b	53	32
3	20	20	20	20	20	20	20	20	63	6c	74	20	43	4c	54	33
4	20	20	20	20	20	20	20	20	64	6d	75	20	44	4d	55	34
5	20	20	20	20	20	20	20	20	65	6e	76	20	45	4e	56	35
6	20	20	20	20	20	20	20	20	66	6f	77	20	46	4f	57	36
7	20	20	20	20	20	20	20	20	67	70	78	20	47	50	58	37
8	20	20	20	20	20	20	20	20	68	71	79	20	48	51	59	38
9	20	20	20	20	20	20	20	60	69	72	7a	20	49	52	5a	39
a	20	20	20	20	5b	21	7c	3a	20	20	20	20	20	20	20	20
b	20	20	20	20	2e	24	2c	23	20	20	20	20	20	20	20	20
c	20	20	20	20	3c	2a	25	40	20	20	20	20	20	20	20	20
d	20	20	20	20	28	29	5f	27	20	20	20	20	20	20	20	20
e	20	20	20	3b	2b	3b	3e	3d	20	20	20	20	20	20	20	20
f	20	20	20	2a	7c	5e	3f	22	20	20	20	20	20	20	20	20

Table 14 - ASCII TO EBCDIC, USENGLSH Translation Table

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	00	00	40	f0	7c	d7	79	97	00	00	00	00	00	00	00	00
1	00	00	5a	f1	c1	d8	81	98	00	00	00	00	00	00	00	00
2	00	00	7f	f2	c2	d9	82	99	00	00	00	00	00	00	00	00
3	00	00	7b	f3	c3	e2	83	a2	00	00	00	00	00	00	00	00
4	00	00	5b	f4	c4	e3	84	a3	00	00	00	00	00	00	00	00
5	00	00	6c	f5	c5	e4	85	a4	00	00	00	00	00	00	00	00
6	00	00	50	f6	c6	e5	86	a5	00	00	00	00	00	00	00	00
7	00	00	7d	f7	c7	e6	87	a6	00	00	00	00	00	00	00	00
8	00	00	4d	f8	c8	e7	88	a7	00	00	00	00	00	00	00	00
9	00	00	5d	f9	c9	e8	89	a8	00	00	00	00	00	00	00	00
a	00	00	5c	7a	d1	e9	91	a9	00	00	00	00	00	00	00	00
b	00	00	4e	5e	d2	4a	92	c0	00	00	00	00	00	00	00	00
c	00	00	6b	4c	d3	e0	93	6a	00	00	00	00	00	00	00	00
d	00	00	60	7e	d4	5a	94	d0	00	00	00	00	00	00	00	00
e	00	00	4b	6e	d5	5f	95	a1	00	00	00	00	00	00	00	00
f	00	00	61	6f	d6	6d	96	00	00	00	00	00	00	00	00	00

Assigning a Translation Table to a Port

```
DEFINE/SET PORT port-list TELNET TN3270 TRANSLATIONTABLE  
trans_name
```

The *trans_name* specifies the translation table to be used during a TN3270 session. The value you specify is either USEENGLSH (supplied by Xyplex) or a table you have created.

Example

In this example, a network manager creates a new translation table for a Spanish language terminal, and then changes an entry in the table for both directions of the data flow:

```
Xyplex>> define server TN3270 translationtable spanish  
        create english
```

This command creates a new translation table called SPANISH based on USEENGLSH. The new table is exactly the same as USEENGLSH because this command only duplicates the existing table and renames it. You must then add or change each entry individually to create the unique characters for the SPANISH table.

The next commands add the Spanish character à for a terminal that provides multinational characters in ASCII positions 80-ff. The ASCII value for this character is e0; the EBCDIC value is 44.

```
Xyplex>> define server TN3270 translationtable spanish  
        asciitoebcdic e0 44  
  
Xyplex>> define server TN3270 translationtable spanish  
        ebcdictoascii 44 e0
```

You can change as many translation table entries as necessary for the new language. When you finish customizing the new table, you can assign it to individual ports for use during a TN3270 session, as follows:

```
Xyplex>> define ports 4-16 telnet TN3270 translationtable
spanish
```

The translation table name Spanish appears in the TN3270 Translation Table field of the Show Port Telnet Characteristics display for Ports 4-16.

Other Commands that Affect Translation Tables

Command	Function
PURGE SERVER TN3270 TRANSLATIONTABLE <i>trans-name</i>	Removes a specified translation table from the server's database.
SHOW/LIST SERVER TN3270 TRANSLATIONTABLE <i>trans-name</i> <i>table</i>	Displays the EBCDICTOASCII or ASCII TO EBCDIC portion of a specified translation table. See Figure 14 (page 116).
SHOW/LIST SERVER TN3270	Displays a list of TN3270 devices and translation tables defined in the permanent database of the server. See Figure 13 (page 35).
SHOW/LIST PORT TELNET CHARACTERISTICS	Displays the translation table for a specified port.

Setting Up TN3270 Terminals

```
Xyplex>> SHOW SERVER TN3270

TS/720 V6.0S65 Rom 470003 HW 00.02.00 Lat Protocol V5.2 Uptime: 5 02:30:27
Address:08-00-87-02-34-56   Name:X023456           Ethernet:A   Number:    0

Port Keymaps : Disabled

Devices : ANSI, VT100, VT220-7, VT220-8, IBM3164

TranslationTables : USENGLSH
```

Figure 13 - Server TN3270 Display

```
Xyplex> SHOW SERVER TN3270 TRANSL USENGLSH EBCDICTOASCII

TS/720 V6.0S65 Rom 470003 HW 00.02.00 Lat Protocol V5.2 Uptime: 5 02:45:51
Address:08-00-87-02-34-56   Name:X023456           Ethernet:A   Number:    0

TranslationTable Name: USENGLSH           Table: EBCDICTOASCII

      0x  1x  2x  3x  4x  5x  6x  7x  8x  9x  ax  bx  cx  dx  ex  fx
      -----
x0 | 20  20  20  20  20  26  2d  20  20  20  20  20  7b  7d  5c  30
x1 | 20  20  20  20  20  20  2f  20  61  6a  7e  20  41  4a  20  31
x2 | 20  20  20  20  20  20  20  20  62  6b  73  20  42  4b  53  32
x3 | 20  20  20  20  20  20  20  20  63  6c  74  20  43  4c  54  33
x4 | 20  20  20  20  20  20  20  20  64  6d  75  20  44  4d  55  34
x5 | 20  20  20  20  20  20  20  20  65  6e  76  20  45  4e  56  35
x6 | 20  20  20  20  20  20  20  20  66  6f  77  20  46  4f  57  36
x7 | 20  20  20  20  20  20  20  20  67  70  78  20  47  50  58  37
x8 | 20  20  20  20  20  20  20  20  68  71  79  20  48  51  59  38
x9 | 20  20  20  20  20  20  20  60  69  72  7a  20  49  52  5a  39
xa | 20  20  20  20  5b  21  7c  3a  20  20  20  20  20  20  20  20
xb | 20  20  20  20  2e  24  2c  23  20  20  20  20  20  20  20  20
xc | 20  20  20  20  3c  2a  25  40  20  20  20  20  20  20  20  20
xd | 20  20  20  20  28  29  5f  27  20  20  20  20  20  20  20  20
xe | 20  20  20  2a  2b  3b  3e  3d  20  20  20  20  20  20  20  20
xf | 20  20  20  3b  7c  5e  3f  22  20  20  20  20  20  20  20  20
```

Figure 14 - Server TN3270 Translationtable Display

Using Alternate Keymaps

Alternate keymaps enable you to define multiple sets of input sequences for users' terminal keyboards — all mapping to the same function. For example, a user can invoke the TN3270 PF1 function through any of these methods:

- The numeric keypad 1, or
- ESC followed by the digit 1, or
- <CTRL> <A>.

You can define up to three alternate keymaps, giving a total of four unique keyboard sequences that can be mapped to a single TN3270 function (the original sequence plus three alternates).

NOTE: The server does not check for errors when you map alternate keymaps to a device. It does so when the user attempts to establish a TN3270 session with the host.

The first two alternate sequences are displayed on the TN3270 Showkeys display, in addition to the regular primary keymap.

Defining the Alternate Keymaps

Alternate keymaps are stored as TN3270 device table entries (as are normal keymaps). The maximum number of TN3270 device table entries is 20. Use this command to create an entry to hold the alternate keymap:

```
DEFINE SERVER TN3270 DEVICE device-name CREATE EMPTY
```

When you use this command, the Show Server TN3270 Device Name command does not display nonessential information.

IMPORTANT

Only use the Create Empty option for alternate keymaps. A device table entry that you specify as a port's Telnet TN3270 device name *must not* be created EMPTY.

Use this command to map an alternate keymap to a TN3270 device table entry:

```
DEFINE SERVER TN3270 DEVICE device-name KEYMAP ALTMAP name
```

The *device-name* specifies the name of the TN3270 device table entry to which the alternate keymap is being assigned. The *name* identifies the TN3270 device table entry that contains the alternate keymap. Specifying NONE as the *name* disables any alternate keymap for the TN3270 device.

Example

The following commands modify the distributed VT100 device and add two alternate keymaps. As a result, the TN3270 function keys PA1-PA3 can also be invoked by the sequences ESC-1 through ESC-3, and Control/A-1 through Control/A-3.

```
DEFINE SERVER TN3270 DEV VT100 KEYMAP ALTMAP TV925
DEFINE SERVER TN3270 DEV TV925 CREATE EMPTY
DEFINE SERVER TN3270 DEV TV925 KEYMAP PA1 "1B 31" "ESC-1"
DEFINE SERVER TN3270 DEV TV925 KEYMAP PA2 "1B 32" "ESC-2"
DEFINE SERVER TN3270 DEV TV925 KEYMAP PA3 "1B 33" "ESC-3"
DEFINE SERVER TN3270 DEV TV925 KEYMAP ALTMAP PT250
DEFINE SERVER TN3270 DEV PT250 CREATE EMPTY
DEFINE SERVER TN3270 DEV PT250 KEYMAP PA1 "01 31" "CTRA1"
DEFINE SERVER TN3270 DEV PT250 KEYMAP PA2 "01 32" "CTRA2"
DEFINE SERVER TN3270 DEV PT250 KEYMAP PA3 "01 33" "CTRA3"
```

Associated Displays

The Show Server TN3270 Device command displays the names of alternate keymap devices. The SHOWKEYS display for a TN3270 session displays the descriptions for up to three keymaps; the primary keymap plus two alternate keymaps. Each description is separated by a comma.

Error Codes

The creation of an operational keymap occurs during the opening of a TN3270 session. A list of possible errors follows:

-1000- Cannot use reserved TN3270 device name

Explanation: The names EMPTY and NONE are reserved names and cannot be used as device names. This error is displayed during command processing, when you define the device name with a reserved name.

Action: Define a device name other than EMPTY or NONE.

-1001- TN3270 alternate keymap 'name' does not exist

Explanation: You have attempted to establish a TN3270 session in which the alternate keymap name does not exist in the list of devices.

Action: Check the List/Show Server TN3270 display for devices that have been created. Use the command `DEFINE SERVER TN3270 DEVICE name CREATE EMPTY.`

Setting Up TN3270 Terminals

-1002- TN3270 maximum number of alternate keymaps exceeded

Explanation: You have defined more than three alternate keymaps or you have defined an alternate keymap that points back to itself (as shown in the following example).

```
define server TN3270 dev vt100 keymap altmap  
vt100c  
define server TN3270 dev vt100c keymap altmap  
vt100
```

Action: View the `LIST/SHOW SERVER TN3270 DEVICE name display` and check the alternate device name. Then enter the command again using the alternate name for the *name* and check its alternate device name.

If they are the same, the alternate keymaps are pointing back to each other. If they are not the same, continue using this command until you reach the fourth alternate keymap name. Correct the problem by defining the third alternate keymap as having no alternate map.

Example: Keymap A points to altmap B /* ok */
Keymap B points to altmap C /* ok */
Keymap C points to altmap D /* ok */
Keymap D points to altmap E /* NOT ok */

To correct: `DEFINE SERVER TN3270 DEVICE D KEYMAP ALTMAP none`

-1003- TN3270 maximum keymap size exceeded

Explanation: More than 500 keys have been mapped into the device's token table. Given the default of 59 tokens plus three alternate maps with 59 keymaps defined, this is an unlikely error.

Action: Limit the amount of tokens to less than 500.

-1004- TN3270 could not allocate memory to start session

Explanation: TN3270_INIT failed allocating (reserving portions of) memory for building a keymap.

Action: A corrupted keymap might exist or there might be a shortage of memory. Contact Xyplex Customer Support.

-1005- TN3270 error building key mappings

Explanation: This error is displayed in conjunction with one of the above errors and indicates that TN3270 has failed to build a keymap for the device, and that the TN3270 session has terminated.

Action: Refer to the actions for errors 1001-1004.

Using Line 25 as a Status Line

To configure the access server to support the use of the 25th line of a VT330 or VT420 terminal as a status line, follow these steps:

1. Change the terminal display.
2. Select the 25th line.
3. Activate the 25th line.

Change the Terminal Display

Using the terminal's Display Setup menu, enable the "Host Writable Status Display" setting. The terminal's default setting is "No Status Display."

Select the 25th Line

For the TN3270 session to write to the 25th line, you must define hexadecimal values that "select" the 25th line — using any of the four RESET screenmap commands. An example follows:

```
Xyplex>> define server TN3270 device vt330 screenmap reset3  
          "1B 5B 32 24 2D"
```

This command defines the control function that enables the host to select the type of status line to be displayed on line 25. The hex value 32 used in the preceding example denotes "Host Writable Status Line."

Activate the 25th Line

To activate the 25th line, use the STATUS1 screenmap command.
An example follows:

```
Xyplex>> define server TN3270 device vt330 screenmap status1  
          "1b 5b 31 24 7d 1b 5b 3b 48 f6 1b 5b 30 24 7d"
```

The 15 hexadecimal values in the STATUS1 screenmap denote the following:

- The first five hex values select the active status display as the status line and specify that the terminal sends data to the status line only.
- The next four values position the cursor at the beginning of the status line.
- The F6 is a special value that instructs TN3270 to write the status line data in ASCII. Changing the value to F4 would instruct TN3270 to display the status line in hexadecimal.
- The last five hex values select the active status display as the main display, instructing the terminal to send data to the main (1 - 24) lines only. (This is basically a reset. If the last five values were not entered, all TN3270 data — including the status line — would be displayed on the 25th line.)

Status Line Information

Status	Column	Meaning (Action)
S	1	Subsystem-ready
A	2	MyJob value — indicating connection to the host application
X ()	8	Terminal wait. (Wait for the condition to clear.)
X <*>	8	WrongPlace — The entry you attempted is invalid for that position on the screen; for example, entering data into a protected field. (Press the RESET key; move the cursor to the correct position.)
X *NUM	8	Numeric-only field. You tried to enter a lowercase aplphabet character into a numeric field. (Press the RESET key and correct the data in the field.)
X *>	8	You tried to enter too much data into a field on the screen. (Press the RESET key and enter the correct amount of data.)
X ? +	8	Input not accepted/understood. (Press the RESET key; check the screen, and try to enter the data again.)
X -S	8	Invalid key combination; key is not mapped to a keymap. (Press the RESET key to continue.)
^	8	Insert status.
X Printing	8	Printing in progress.
X PrtBusy	8	Printer is busy.
X NoPrinter	8	Printer is not attached. (Press RESET to continue.)

Local Printer Support

You can assign two or more local printers for TN3270 screen printing. To enable support for screen printing, you assign the ACCESS PRT3270 characteristic to one or more ports. The TN3270 printer ports must have valid device names.

To assign a port to a specific printer or to any available printer, use this command:

```
DEFINE/SET PORT port-list TELNET TN3270 PRINTERPORT [port-numb]  
[ANY]
```

The *port-list* specifies one or more server ports to assign to a printer port for local TN3270 screen printing.

The ANY keyword specifies that the server may use any available port with Access PRT3270 enabled to print the screen.

A *port-number* can be any valid port number with ACCESS PRT3270 enabled and a valid TN3270 device name.

The Show/List/Monitor Ports Telnet Characteristics display includes the TN3270 printer port assignment, or ANY if you have not specified a port.

The following commands set up normal 80-column print screen handling on a server port:

```
DEFINE PORT port-list ACCESS PRT3270  
DEFINE PORT port-list TELNET TN3270 DEVICE VT100  
DEFINE PORT port-list AUTOBAUD DISABLED  
DEFINE PORT port-list SPEED baud-rate  
DEFINE PORT port-list PARITY parity  
DEFINE PORT port-list CHARACTER SIZE character-size  
LOGOUT PORT port-list
```


Chapter 5

Setting Up Daemons

Daemons are system processes that run in background mode at hosts, or at network devices such as access servers. Daemons are often used in UNIX environments and TCP/IP networks to exchange information about network activity, or to manage resources such as printers.

Access servers support these environments by supporting a number of standard daemons. Some benefits of this support include:

- Improving the UNIX “look and feel” of the server, making it easier for UNIX users to use the server.
- Enabling the server to take advantage of information that is typically available in a TCP/IP network, and to make this information available for host users, too.

This chapter describes how to set up the following daemons at the access server:

- **fingerd** — Provides a method for exchanging information among hosts about users who are logged on to a server, using the Finger User Information Protocol (RFC 1288). Refer to [page 129](#).
- **routed** — Provides a method for exchanging routing information among gateways or hosts, using the Routing Information Protocol (RFC 1058). Refer to [page 133](#).

Setting Up Daemons

- **rwhod** — Provides a method for collecting information about domain names on the network by listening to “rwho” messages and adding currently unknown domain names to the domain name table. Refer to [page 135](#).

The server supports additional daemons, which are described in other documents. These include:

- **lpd** — Provides a method for exchanging print jobs between hosts and managing jobs that are in a print queue, using the protocol defined by RFC 1179. Setup and configuration of this daemon is described in the guide *Configuring Printer Serving Features*.
- **syslogd** — Provides a central facility for logging messages about server events. These messages can be logged at the server and/or in a file at a UNIX host.

Xyplex also supplies a host-based daemon, **csportd**, that you use to manage resources at server ports. This daemon provides a general-purpose tool for making connections to a port and “piping” data to or from that port. Setup and configuration of this daemon is described in the guide *Configuring Printer Serving Features*.

Enabling Daemons

Use these commands to enable daemons:

```
Xyplex>> define server daemon fingerd enabled
Xyplex>> define server daemon routed enabled
Xyplex>> define server daemon rwhod enabled
```

In place of the `DEFINE SERVER DAEMON ROUTED ENABLED` command, you can use:

```
Xyplex>> define server rip state enabled
Xyplex>> define rip state enabled
```

The server responds with a message similar to:

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

NOTE: Chapter 2 ([page 9](#)) discusses the amount of memory used by the daemons.

Reboot the server after issuing the command for the change to take effect.

Using the Finger Daemon (fingerd)

The finger daemon (fingerd) provides a method for exchanging information between hosts about users who are logged on to a server, using the Finger User Information Protocol (RFC 1288). The daemon responds to requests made at a host for information about a user.

Once the daemon is enabled, no other user activity is required. The daemon simply makes information about ports and users available when a user at a host makes a finger request.

Setting Up Daemons

Typically, hosts allow you to specify either a domain name or an IP address when making finger requests. Some implementations support options such as “verbose.” Refer to your host’s finger daemon documentation for information about these options.

Some implementations also allow a finger request to be forwarded through a host or server to another host or server. The Xyplex finger implementation does not forward these requests.

Xyplex servers always respond with one of two display types: a standard display and a verbose format display. This is an example of the standard display:

```
% finger @172.19.197.98
[172.19.197.98]
User Name      Port   Idle   Login      Port Name     Status
Charlie        8      00:01:10 12-Jul 11:27 PORT_8      Wait Input
```

This is an example of the verbose display:

```
xap> finger -l Charlie@172.19.197.98
[172.19.197.98]
Port 8:: Charlie      Login: 12-Jul 1993 11:27 Status: Connected
+Session 1:: UNIXHOST
Session 2:: VMSHOST
```

The following table describes each field in the finger displays:

Field	Description																
Username	A user-supplied name; the name given to the port through the Port Username setting; or “(Remote)” for ports with a remote (host-initiated) connection.																
Port	The access server port number.																
Idle	The amount of time that the port has been idle.																
Login	The time when the port was logged in.																
Port Name	The name of the port. The default name is PORT_ <i>n</i> , where <i>n</i> is the physical port number. You can assign a different name through the Define/Set Port Name command.																
Session	The name of the host or server destination of any sessions running at the port. A plus sign (+) indicates that the session is currently active. If a remote connection has been formed with the port, the service is displayed as “Remote Connect.”																
Status	<p>The current status of the port, which can be:</p> <table> <tr> <td>Autobaud</td><td>Being “autobauded” (selecting a port speed).</td></tr> <tr> <td>Available</td><td>The port’s Access setting is Remote or Dynamic, and the port is not busy.</td></tr> <tr> <td>Check Modem</td><td>Verifying that modem signals are being properly asserted.</td></tr> <tr> <td>Check Connect</td><td>Determining the status (accepted or rejected) of a pending connection.</td></tr> <tr> <td>Connected</td><td>Currently connected to a LAT service or Telnet destination.</td></tr> <tr> <td>Connect Wait</td><td>Waiting to retry a connection attempt (used when Port Autoconnect is Enabled).</td></tr> <tr> <td>Connecting</td><td>Currently attempting to connect to a LAT service or Telnet destination.</td></tr> <tr> <td>Dialback Wait</td><td>Waiting for the remote modem to answer a dial-back call.</td></tr> </table>	Autobaud	Being “autobauded” (selecting a port speed).	Available	The port’s Access setting is Remote or Dynamic, and the port is not busy.	Check Modem	Verifying that modem signals are being properly asserted.	Check Connect	Determining the status (accepted or rejected) of a pending connection.	Connected	Currently connected to a LAT service or Telnet destination.	Connect Wait	Waiting to retry a connection attempt (used when Port Autoconnect is Enabled).	Connecting	Currently attempting to connect to a LAT service or Telnet destination.	Dialback Wait	Waiting for the remote modem to answer a dial-back call.
Autobaud	Being “autobauded” (selecting a port speed).																
Available	The port’s Access setting is Remote or Dynamic, and the port is not busy.																
Check Modem	Verifying that modem signals are being properly asserted.																
Check Connect	Determining the status (accepted or rejected) of a pending connection.																
Connected	Currently connected to a LAT service or Telnet destination.																
Connect Wait	Waiting to retry a connection attempt (used when Port Autoconnect is Enabled).																
Connecting	Currently attempting to connect to a LAT service or Telnet destination.																
Dialback Wait	Waiting for the remote modem to answer a dial-back call.																

Setting Up Daemons

	Disconnected	A session was disconnected (for example, because it was inactive for too long).
	Disconnecting	A session is disconnecting from a LAT service or Telnet destination.
	Executing Cmd	Executing a command from the local command mode.
	Finding Script	Searching for a script file via TFTP read requests.
	First Dialback Login	Making a first attempt to locate a dial-back script.
	Idle	Currently not in use.
	Loading Script	The port has located a script file and is receiving the file from the script server.
	Local Mode	Logged on to the server; currently in local command mode.
	Locked	Disabled through the Lock command.
	Login	Waiting for the user to enter a login or Kerberos password.
	Login Wait	Disabled (for 60 seconds) because an incorrect password has been entered, or because a dial-back attempt failed.
	Logout	Logged out.
	Password	Waiting for the password required by a password-protected LAT service.
	PPP	Used for the Point-to-Point Protocol..
	Reset	Reverting to stored configuration.
	Retry Connect	Trying to connect to a service that was previously unavailable (used when Port Autoconnect is Enabled).
	Running Script	Executing the commands in a script file.
	Second Dialback Login	Making a second attempt to locate a dial-back script (the port searches the directory path "above" the path specified for this script server).
	Slip	Used for Serial Link Internet Protocol.

	Suspended	The user has entered the local-switch character; the session is being suspended.
	Test Wait	Executing a Test Service command.
	Test Out	Outputting the results of a Test Service command.
	Wait Input	The port is at the local prompt (waiting for the user to enter a command).
	Wait Logout	Waiting for modem control signals to be deasserted.
	Wait Output	Completing display output before logging out.
	Wait Queue	A connection request from this port is in a queue.
	Wait Session	The session is being disconnected.

Using the Route Daemon (routed)

This daemon provides a method for exchanging routing information among gateways or hosts, using the Routing Information Protocol (RIP), as defined by RFC 1058. The access server uses RIP to learn about IP routes from hosts or gateways.

The server listens for routing messages and updates its routing tables, but does not transmit any routing information to the default primary or secondary gateways or hosts (The server is a “silent” or “passive” router.)

Setting Up Daemons

Once the daemon is enabled, no other user activity is required. You can use the Show/Monitor Internet Routes command to display the routes that the server has learned, as well as any “static” (user-defined) routes that have been defined. A sample display follows:

Xyplex>> MONITOR SERVER INTERNET ROUTES ALL						
	Address	Gateway	Mask		Last Modified	
1	192.168.19.255	172.16.201.7	255.255.255.0	NET/FIXED	21 Mar 1995	09:22
2	192.169.12.21	172.16.201.8	0.0.0.0	HOST/VAR	24 Mar 1995	13:58*

Refer to “Configuring IP Routes” in Chapter 4 ([page 36](#)) for more information about IP routes.

The server processes routes that it learns through RIP in the following way:

- The server enters the routes into the routing table as “variable” routes, which can be re-learned. These routes overwrite any previously learned or manually specified variable route to the same destination. “Fixed” routes are not overwritten by learned routes.
- Routes learned through RIP are not saved to the permanent database. They are lost when the server is rebooted.
- Routes learned through RIP expire after five minutes unless the server hears another RIP broadcast message for the same route.

Using the rwho Daemon (rwhod)

This daemon provides a method for collecting information about domain names on the network by listening to “rwho” messages and adding currently unknown domain names to the domain name table.

Once the daemon is enabled, no other user activity is required. You can use the Show/List/Monitor Domain command to display the domain names that the server has learned, or any manually defined domain names. A sample display follows:

Xyplex> show domain					
Internet			Domain		19 Jan 1989 08:23:58
Entry	Address	TTL	Source	Name	
1	192.168.119.100	100	Pri	UNIXHOST.COM	
2	192.168.119.200	24	Sec	FINANCESUN.COM	
3	192.168.119.240		Loc	NET5000.COM	
4	192.168.119.242		Loc	MAX1600.COM	
5	192.168.119.216	1438	Who	ENGHOST.COM	

When the rwhod daemon is enabled, any domain names that have been learned through rwho messages are indicated as having “Who” as the source in the Show/Monitor Domain display. Refer to [Using the Server as a Domain Name Server](#) in Chapter 4 ([page 87](#)) for more information about IP routes.

When the server receives an rwho message, if the domain name already exists in the domain name table — and its source is either the primary or secondary name server — the entry is overwritten with “Who” as the source and a time-to-live of 1440. If the domain name exists in the table, and its source is “local,” the entry is not overwritten.

Chapter 6

Using Nested Menus

This chapter explains how to use the Nested Menu feature. It covers the following topics:

- About the Nested Menu Feature
- Setting Up the Script Servers
- Creating the Nested Menu File
- Configuring the Access Server to Support Nested Menus
- Sample Nested Menu Files

You can enable or require nested menus at specific ports, or through a user's login script. Chapter 10 ([page 174](#)) explains how to create login scripts. Note that you cannot enable both the Nested Menu feature and the Menu feature on the same port.

Memory Requirements

The access server must have at least 2 MB of installed memory to support the Nested Menu feature. It must also be running appropriate software. The software for Network 9000 720 modules and the MAXserver 1620/1640 support this feature. Refer to the access server *Software Release Notes* for information about software for other server types.

About the Nested Menu Feature

The Nested Menu Feature enables you to create up to 255 menus, in up to 10 levels. Each menu can have up to 20 entries. To enable the Nested Menu feature on the server, you allocate (reserve portions of) memory for the menu file. The amount of memory you allocate determines the number and size of the menus.

You specify one menu as the Top Level Menu at each port where nested menus are enabled or required. The server displays this menu first. You can specify a different top level menu at different ports.

Figure 15 shows a three-level menu structure. The top level menu for Ports 2-6 is Menu 1; for Port 7 it is Menu 3; and for Port 10 it is Menu 8.

Using Nested Menus

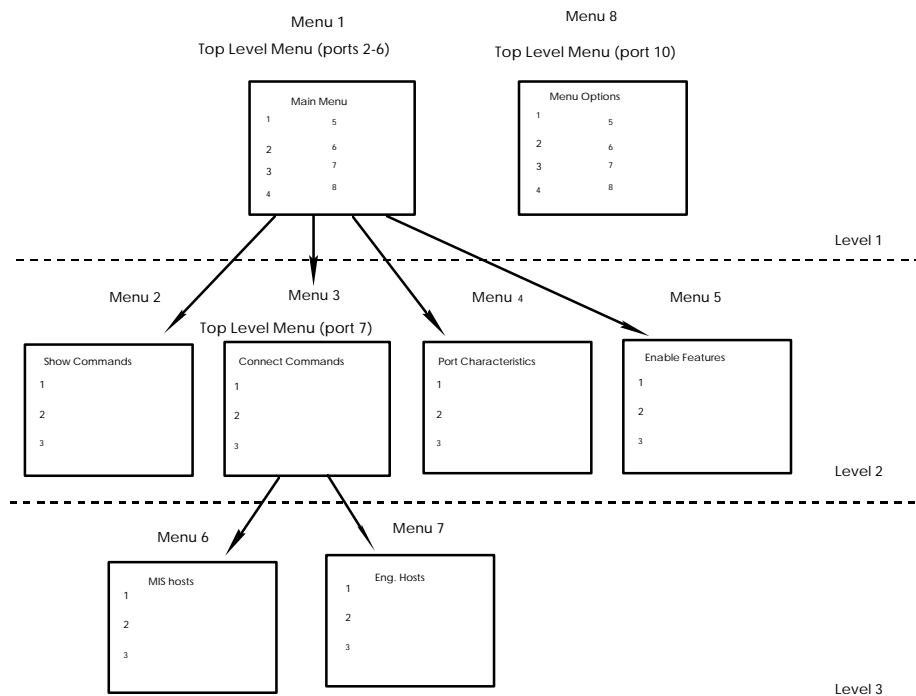


Figure 15 - Three-Level Menu Structure

Users at Ports 2-6 have access to Menus 1, 2, 3, 4, 5, 6, and 7, beginning at Menu 1. Users at Port 7 have access to Menus 3, 6, and 7, beginning at Menu 3. Users at Port 10 have access to Menu 8 only. The options on Menus 1 and 3 run Xyplex commands or open other menus. The options on menus 2, 4, 5, 6, 7, and 8 run commands only.

Using Nested Menus

Figure 16 shows what Menu 1 might look like:

Main Menu

1. Show Commands	5. Set Session Mode
2. Connect Commands	6. Resume a Session
3. Port Characteristics	7. Disconnect a Session
4. Enable Features	8. Help

Up one level: ^U Top of Menu: ^T Exit Menus: ^E Repaint: ^R Logout ^L
Enter number of selection or use arrow keys:

Figure 16 - Sample Menu

Users can type the characters shown at the bottom of the menu to move up through the menu levels without choosing options, or to exit from the menus if the port is also configured to support the command interface. You define these characters within the menu file. Users can use arrow keys (↑ and ↓) to move among the entries within a menu.

How the Server Obtains the Menu File

When the server boots, it searches for a predefined script server, which stores the menu file. When it finds the menu file, the server loads it into local memory. If the server cannot find the menu file, it continues to operate normally. It logs an error in the Accounting Log if Verbose accounting is enabled.

How a Port Obtains the Menus

When a user logs on to a port, the server searches the menu file for the top level menu, and displays that menu on the terminal. The user can then choose options from the various menus. Commands within the top level menu determine the sequence of other menus in the file.

If you have not specified a top level menu, or the port cannot find the menu file, the terminal displays an error message. If nested menus are *enabled*, the port reverts to the command interface. If nested menus are *required*, the server logs out the port.

A user can go to the command interface from within a menu if Nested Menus are not required on the port. To do this, the user enters a predefined Exit key within the menu, or chooses a menu option that disables the nested menu feature.

Setting Up Script Servers

1. Determine which hosts or access servers will act as script servers. Script servers must run TFTP. Each access server can have a maximum of four script servers. You can use two or more hosts as backup script servers; these can be a combination of script server types.
2. Determine where the TFTP process on the UNIX host starts its path search; this is the TFTP Home directory. Refer to the host's TFTP documentation or MAN page.
3. Set up a directory to contain the menu file on each script server.

Create a directory to contain the menu file. On some UNIX systems, you can create a top-level directory just for the menu file, rather than using a directory that already contains many files (e.g., `/usr`, `/bin`, `/tftpboot`, or `/etc`). Systems running with the TFTP secure option enabled might require you to place the menu file in `/tftpboot` .

Figure 17 shows how you might set up a directory to contain the menu file on a UNIX host. In the figure, the file `n.menu.file` contains the nested menu commands. The examples that follow show how to create this directory on a UNIX host and an access server.

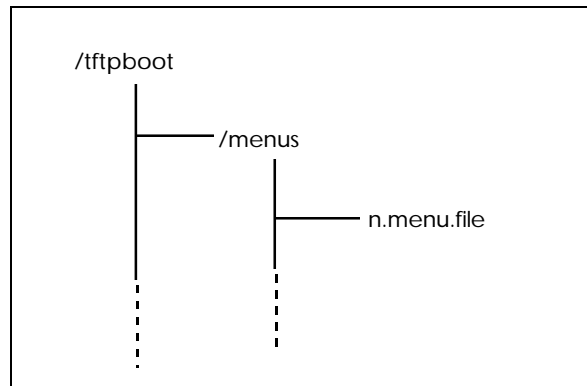


Figure 17 - Sample Script Server Directory Structure

UNIX Host:

Create the menus directory using lower-case letters:

```
% mkdir menus
```

Access Server (MAXserver 1820):

Use the MAXserver system disk in a DOS-based PC to create the directory:

```
A:> mkdir menus
```

Specify the location of the script server and the pathname of the menu file, using the Define/Set Server Script Server commands; for example:

```
Xyplex>> define server script server 172.18.12.10 " "
```

After creating the menu directory and defining the script server, use a text editor to create the menu file. The next section describes the commands that you use. The section [“Sample Nested Menu Files”](#) (page 156) includes two sample menu files.

Creating the Nested Menu File

Use the nested menu commands to:

- Define menu options
- Create text strings that prompt for input or specify menu titles
- Define keys for moving among the different menus

Table 15 lists the commands for creating nested menus:

Table 15 - The %menu Commands

Command	Purpose
%menu_file	Define the beginning of a menu file.
%menu_start	Define the beginning of a menu.
%menu_entry <i>n</i>	Define a menu entry.
%menu <i>n</i>	Open a menu from within a menu.
%menu_wait	Wait for one command to complete, then prompt the user for input before executing the next command.
%menu_noprompt	Display the menu without prompting the user after executing a command.
%menu_end	Define the end of a menu.
%menu_prompt	Define the menu entry prompt.
%menu_continue	Define the menu continue prompt.
%menu_top %menu_up %menu_exit %menu_logout %menu_repaint	Define menu control characters.

The following sections describe these commands in detail. Refer to [“Sample Nested Menu Files”](#) on page 156 for examples.

%menu_file

The menu file must begin with `%menu_file` or the server will not recognize it as a menu file.

%menu_start *n title*

This command indicates the beginning of a menu and specifies the menu number and title.

The value of *n* is the menu number. Valid values are from 1 through 255. You use a menu number to specify the top level menu on the server port, through the `DEFINE/SET PORT NESTED MENU TOP LEVEL menu-number` command.

The name you specify as the *title* appears at the top of the menu. The title can include up to 16 characters.

%menu_entry *n entry-description command-string*

This command defines a menu option. The value of *n* is the entry number for the item on the menu. Follow the option number with a delimiting character such as a carriage return, line feed, or form feed.

The text of *entry-description* appears on the menu next to the entry number. The description can include up to 30 characters. Follow the description with a delimiting character such as a carriage return, line feed, or form feed.

The *command-string* includes one or more Xyplex commands and `%menu` commands to be executed when the user chooses the entry number. If a port has the DECserver-like Interface (DLI) enabled — as it does by default — the server interprets the commands in the menu file as part of the DLI command set.

Using Nested Menus

If the port has the UNIX-like interface (ULI) enabled, the interface interprets the commands in the menu file as part of the ULI command set.

The *command-string* can include up to 132 characters. If you include more than one command, separate the commands with a semi-colon (;). For example, the following command sets the port to Privileged mode, executes a privileged command, and then resets the port to Nonprivileged mode:

```
set priv;monitor server status;set nopriv
```

This example opens the ULI from a DLI port to execute the command **alias**, then reopens the DLI:

```
uli;alias;dli
```

You can include a wildcard character in a command to prompt for user input. For example, the command **connect #** instructs the server to prompt the user to enter a network destination.

You can also execute a script file in a menu entry command string. Doing so provides a way of executing a string of commands that exceeds the 132-character limit of a menu entry. The format for the script command is:

```
SCRIPT "pathname/script-name"
```

%menu *n*

Opens the menu that you specify with the *n* variable. You can use this command in the *command-string* of the **%menu_entry** command.

%menu_wait

Wait for one command to complete, then prompt the user for input before executing the next command. This command is useful when you specify two or more Show/List/Monitor commands in a menu entry. The interface displays one screen, then waits and prompts for input before displaying another screen. Use the **%menu_continue** command to define this prompt.

%menu_noprompt

Redisplay the menu after executing commands, without prompting the user for input.

%menu_end

Indicates the end of a menu. You can begin another menu after this command appears in the file, or use other menu commands.

%menu_prompt *prompt-text*

This command specifies a text string that explains how to select a menu option. This prompt appears at the bottom of the menu where the user enters an option number. The prompt text can include up to 132 characters. The default prompt text is “Enter Number of Selection or Use Arrow Keys:”

%menu_continue *prompt-text*

This command specifies a text string that explains how to redisplay a menu after a Show/Monitor/List command executes. The prompt text can include up to 132 characters. The default text is “Press <RETURN> to continue...”

%menu_top x *text-string*

This command specifies the menu key character that the user types to open the top level menu. The *text-string*, which can include up to 20 characters, describes the purpose of the character. A typical entry might be **%menu_top T Top of Menu**. The key and the text appear at the bottom of the menu. No default exists for this key.

%menu_up x *text-string*

This command specifies the menu key character that the user types to open a menu one level up. The *text-string*, which can include up to 20 characters, describes the purpose of the character. A typical entry might be **%menu_up U Up One Level**. The key and the text appear at the bottom of the menu. No default exists for this key.

%menu_exit x *text-string*

This command specifies the menu key character that the user types to to exit from the menu interface. If nested menus are *enabled* at a port, typing this character recalls the command interface. If nested menus are *required* at the port, typing this character logs out the port. The *text-string*, which can include up to 20 characters, describes the purpose of the character. A typical entry might be **%menu_exit E Exit Menus**. The key and the text appear at the bottom of the menu. No default exists for this key.

%menu_logout x *text-string*

This command specifies the menu key character that the user types to to log out of the menu and the server port. The *text-string*, which can include up to 20 characters, describes the purpose of the character. A typical entry might be **%menu_logout Q Logout**. The key and the text appear at the bottom of the menu. No default exists for this key.

%menu_repaint *x text-string*

This command specifies the menu key character that the user types to refresh the menu screen. The *text-string*, which can include up to 20 characters, describes the purpose of the character. A typical entry might be **%menu_repaint R Refresh**. The key and the text appear at the bottom of the menu. No default exists for this key.

Using Comment Lines in the Menu File

Begin a comment line with an exclamation point (!) and follow it with the comment text. An example follows:

```
%menu_end
!start Menu 3.  Menu 3 displays CONNECT commands.
%menu_start 3 Connect Commands
```

The first line specifies the end of a menu. The second line is a comment explaining that the next section of the file defines a new menu, Menu 3. The third line is a menu command that begins Menu 3.

General Guidelines

When you create the menu file, observe these guidelines:

- The first line of every menu file is `%menu_file`.
- You can define menus in any order within the file.
- Each menu must have a menu number.
- You can define menu entries in any order within a menu.

The menu file requires that each menu entry have a number, subject to these conditions: (1) if the first entry does not have a number, the file assigns a number to each entry in ascending order, even if subsequent entries have numbers. (2) if the first entry has a number, all entries in the menu must have numbers.

The section “[Sample Nested Menu Files](#)” (page 156) shows two sample menu files, which use all `%menu` commands.

Debugging the Menu File

A syntax error in the menu file prevents the file from executing when a user attempts to log on to the port. If nested menus are *enabled* on the port, the interface displays an error message and opens the command interface. If nested menus are *required* on the port, the interface displays the error message and does not allow the user to log on.

If the Verbose Accounting feature is enabled, the server logs the error and the line number within the menu file where the error occurred. The Show Server Accounting display includes this information.

For example, the following entry appears if an error occurs on line 24 of the file:

```
12 August 1995 05:30:01 Illegal use of reserved Nested
Menu keyword : line # 24
```

Chapter 8 ([page 200](#)) explains how to enable Verbose Accounting.

Configuring the Server to Support Nested Menus

The following sections explain how to configure the server and individual ports to use the Nested Menu feature. Refer to this section after you have created the menu file.

Server Settings

To enable the Nested Menu feature on the server, you must allocate (reserve portions of) memory for the menu file and specify the menu filename. After defining these settings, reboot the server.

Allocate Memory for the Menu File

Determine how much memory you want to allocate for the menu file. You can allocate between 0 and 204,800 bytes. Allocating any memory for the menu file enables the Nested Menu feature on the server; allocating 0 bytes disables it. If you plan to add menus to the file at a later time, you can allocate more space than the current file requires.

To determine the approximate amount of memory the file requires, use the appropriate command on the script server. *On a UNIX host*, you can use the `ls -l` command; for example:

```
ls -l n.menu.file  
  
-rwxrwxrwx 1 216 158000 Aug 18 16:00 n.menu.file
```

The display shows that the size of the file is 158,000 bytes. (The file permissions shown here are only an example, not a requirement.)

On a PC, you can use the `dir` command:

```
B:> dir n.menu.file  
  
nmenufile      08-07-96 7:24p  
1 file(s) 158000 bytes
```

The display shows that the file size is 158,000 bytes.

When you determine the size of the menu file, allocate memory on the server through the Define Server Nested Menu Size command; for example:

```
Xyplex>> define server nested menu size 158000
```


IMPORTANT

After defining the menu file size, specify the name of the file before you reboot the server. Otherwise, the server cannot determine the name of the menu file on the script server. (Refer to the following section.)

Specify the Name of the Menu File

Use the Define Server Nested Menu Name command to specify the name of the menu file on the script server; for example:

```
Xyplex>> define server nested menu name "n.menu.file"
```

Wait until the “Storage State” field on the Monitor Server Parameter Server display goes to “Idle,” then reboot the server:

```
Xyplex>> init delay 0
```

Port Settings

Three commands specify Nested Menu port settings. The commands that specify the status of nested menus and the top level menu are mandatory. You can optionally enable the Privileged Nested Menu feature.

Specify the Status of the Nested Menus on Ports

You can enable or require Nested Menus in user login scripts or on one or more ports. Refer to Chapter 8 ([page 178](#)) for information about creating scripts. Use the following command to specify the status of Nested Menus at specified ports. Nested Menus are disabled by default at all ports.

```
DEFINE/SET PORT port-list NESTED MENU [ENABLED]  
[DISABLED]
```

Example:

```
Xyplex>> set port 10-15 nested menu enabled
```

If Nested Menus are enabled at a port, a user can access the command interface from the menu interface. Or, if the server cannot obtain the menu file, the command interface appears on the display when the user logs into the port.

If Nested Menus are required at a port, a user cannot access the command interface. If the server cannot obtain the menu file, the server does not allow the user to log on to the port.

Specify a Top-Level Menu

You must specify a top-level menu at each port where Nested Menus are enabled or required. Otherwise, the server cannot determine which menu in the file to display first. You do not need to specify the same top-level menu for every port. Use the menu number, which you assigned to each menu when you created the menu file, to specify the top-level menu. This example uses menu number 1:

```
Xyplex>> set port 2-6 nested menu top level 1
```

Specify Privileged Nested Menus, If Needed

If you enable the Privileged Nested Menu feature, you can include privileged commands on the menu without including the Set Privilege command in the menu file. This feature is disabled by default. Use the following command to enable it:

```
Xyplex>> set port 2-6 privileged nested menu enabled
```

NOTE: If you disable the Privileged Nested Menu feature, you also disable the Nested Menu feature on the port. If this happens, you must specify the status of Nested Menus on a port as enabled or required before you can use the feature.

Sample Nested Menu Files

Sample File 1 shows a section of a menu file that defines some of the menus in the 3-level menu structure shown in [Figure 15](#) (page 139). Sample File 2 shows a complete nested menu file, without comments. These examples use all nested menu commands.

Sample File 1

<code>%menu_file</code>	Indicates a valid menu file.
<code>!Start Menu 1 - Main Menu</code>	Comment line.
<code>%menu_start 1 Main Menu</code>	Defines the beginning of menu number 1, and assigns it the title "Main Menu."
<code>%menu_entry 1 <CR> Show Commands <CR> %menu 2</code>	Defines Menu Entry number 1, and assigns the name "Show Commands" to it. The <code>%menu 2</code> command instructs the server to display Menu 2 when a user selects this entry. The <code><CR></code> specifies a carriage return as the delimiting character after the entry number and the display text.
<code>%menu_entry 2 <CR> Connect Commands <CR> %menu 3</code>	Defines Menu Entry 2, and assigns the name "Connect Commands" to it. The <code>%menu 3</code> command instructs the server to display Menu 3 when the user selects this entry.
<code>%menu_entry 3 <CR> Port Settings <CR> %menu 4</code>	Defines Menu Entry 3, and assigns the name "Port Settings" to it. The <code>%menu 4</code> command instructs the server to display Menu 4 when the user selects this entry.

<code>%menu_entry 4 <CR> Enable Features <CR> %menu 5</code>	Defines menu entry 4, and assigns the name "Enable Features" to it. The <code>%menu 5</code> command instructs the server to display Menu 5 when the user selects this entry.
<code>%menu_end</code>	Indicates the end of Menu 1. You can add more options to Menu 1 as long as they appear in the file between the <code>%menu_start 1</code> command and the <code>%menu_end</code> command.
<code>!Start Menu 8 - Standalone Menu</code>	Comment line
<code>%menu_start 8 Menu Options</code>	Defines the beginning of Menu number 8, and assigns it the title "Menu Options."
<code>%menu_entry 1 <CR> Change Mode to ANSI <CR> set port type ANSI; %menu_noprompt</code>	Defines Menu Entry number 1, and assigns the name "Change Mode to ANSI" to it. The command line includes the Xyplex command that changes the port type, and the menu command that redisplay the menu without prompting the user.
<code>%menu_entry 2 <CR> Connect to Hosts <CR> connect #</code>	Defines Menu Entry number 2, and assigns the name "Connect to Hosts" to it. The wildcard character # means "prompt for a destination when the user selects this entry."

Using Nested Menus

<pre>%menu_entry 3 <CR> Port Information <CR> show port characteristics; %menu_wait;show port status</pre>	Defines Menu Entry number 3, and assigns the name “Port Information” to it. The interface runs the <code>SHOW PORT CHARACTERISTICS</code> command when a user selects this entry. After the Port Characteristics display appears, the interface displays the prompt “Press New Line to continue,” which is defined later in this menu file. When the user presses the New Line key, the interface runs the <code>SHOW PORT STATUS</code> command.
<pre>%menu_end</pre>	Indicates the end of Menu 8. You can add more options to Menu 8 as long as they appear in the file between the %menu_start 8 commands and the %menu_end command.
<pre>!Start Menu 2 - Displays</pre>	Comment line.
<pre>%menu_start 2 Show Commands</pre>	Defines the beginning of Menu number 2, and assigns it the title “Show Commands.” The position of Menu 2 after Menu 8 in the menu file does not affect how the server displays them. The server orders the menus correctly.
<pre>%menu_entry 1 <CR> Display Domains <CR> show domains</pre>	Defines Menu Entry number 1, and assigns the name “Display Domains” to it. The server executes the <code>SHOW DOMAINS</code> command when the user selects this entry.
<pre>%menu_entry 2 <CR> Display Nodes <CR> show nodes</pre>	Defines Menu Entry number 2, and assigns the name “Display Nodes” to it. The server executes the <code>SHOW NODES</code> command when the user selects this entry.

Using Nested Menus

<code>%menu_entry 3 <CR> Display Port Status <CR> show port status</code>	Defines Menu Entry number 3, and assigns the name "Display Port Status" to it. The server executes the <code>SHOW PORT STATUS</code> command when the user selects this entry.
<code>%menu_end</code>	Indicates the end of Menu 2. You can add more options to Menu 2 as long as they appear in the file between the %menu_start 2 commands and the %menu_end command.
<code>%menu_prompt Enter an option number or use arrow keys.<CR></code>	Specifies the menu entry prompt.
<code>%menu_continue Press New Line to continue. <CR></code>	Specifies the menu continue prompt.
<code>%menu_up U Up one level</code>	Specifies U as the character that a user types to display the menu that is up one level from the current menu.
<code>%menu_top T Top of menu</code>	Specifies T as the character that a user types to open the top level menu.
<code>%menu_logout Q Logout</code>	Specifies Q as the character that a user types to logout of the server port.
<code>%menu_repaint R Repaint</code>	Specifies R as the character that a user types to repaint the screen.
<code>%menu_exit E Exit Menus</code>	Specifies E as the character that a user types to exit from the menu file and use the command interface.

Using Nested Menus

Sample File 2

```
%menu_file
!-----
!                                     Level 1 Main Menu
!-----
!top menu
%menu_start 1 Main Menu
!
%menu_entry 1
Telnet session to host:
connect#;%menu_noprompt
!
%menu_entry 2
Rlogin session to host:
rlogin #;%menu_noprompt
!
%menu_entry 4
Enable CSLIP
set port internet cslip enabled;%menu_noprompt
!
%menu_entry 5
PPP Control
%menu 9
!
%menu_entry 6
Xremote to XDM Server:
xconnect #
!
%menu_entry 7
Set Terminal Type
set port type #
!
```



```
%menu_entry 8
Resume a session
sho session;%menu_wait;resume #
!
%menu_entry 9
Disconnect a session
disconnect #
!
%menu_entry 10
ABC UNIX Timesharing
c rcs.abc.edu;%menu_noprompt
!
%menu_entry 11
Port Status
%menu 4
!
%menu_entry 12
Set/Show Port Parameters
%menu 5
!
%menu_entry 13
Set/Show Alt Port Params
%menu 6
!
%menu_entry 14
Set/Show Telnet Parameters
%menu 7
!
%menu_entry 15
Set/Show TN3270 Parameters
%menu 8
!
%menu_entry 17
Server/Network Tools
%menu 2
!
```

Using Nested Menus

```
%menu_entry 18
Server Information
%menu 3
!
%menu_entry 20
Exit Menu to CLI
set port nested menu disabled;%menu_noprompt
!
%menu_end
!
!-----
!                               Level 2 Menu 2 Server/Network Tools
!-----
!
%menu_start 2 Tools
%menu_entry 1
Ping
set priv system;ping #;set nopriv
%menu_entry 2
Test Pattern
test
%menu_entry 4
Lock Terminal
lock
%menu_entry 5
Send message to port
broadcast #
%menu_entry 6
Show Server Users
sho users
%menu_entry 20
Help
help
%menu_end
```

```

!
!-----
!                               Level 2 Menu 3 Server Information
!-----
!
%menu_start 3 Server Information
%menu_entry 1
Main Parameters
show server
%menu_entry 2
Current Status
show server status
%menu_entry 3
Network Statistics
show server counters
%menu_entry 4
Domain Information
set priv system;show domain;set nopriv
%menu_entry 5
IP Information
show server internet
%menu_entry 11
Show Users
show users
%menu_entry 12
Arp Cache
uli;arp -a;dli
%menu_entry 13
Summary
show server summary
%menu_entry 14
Alternate Status
show server alternate status
%menu_entry 15
Routing Tables
show server ip route

```

Using Nested Menus

```
%menu_entry 20
Help
help
%menu_end
!-----
!
!                                     Level 2 Menu 4 Port Status
!-----
!
%menu_start 4 Port Status
%menu_entry 1
Show Port Summary
show port summary
%menu_entry 2
counters
show port counters
%menu_entry 3
Status
show port status
%menu_entry 4
Sessions
show sessions
%menu_entry 11
Show Users
show users
%menu_entry 20
Help
help
%menu_end
!
```

```

!-----
!                               Level 2 Menu 5 Set/Show Port Parameters
!-----
!
%menu_start 5 Port Parameters
%menu_entry 1
Show port Parameters
show port characteristics
%menu_entry 3
Set Terminal Type = Hardcopy
set port type hardcopy;%menu_noprompt
%menu_entry 4
Set Terminal Type = Dumb
set port type soft;%menu_noprompt
%menu_entry 5
Set Terminal Type = VT100
set port type ansi;%menu_noprompt
%menu_entry 6
Set Switch to Command Mode chr
set port local #
%menu_entry 13
Set Session Forward Switch chr
set port forward #
%menu_entry 14
Set Session Backwrdr Switch chr
set port back #
%menu_entry 15
Set char length (7-8)
set port cha #
%menu_entry 16
Set Parity
set port parity #
%menu_entry 20
Help
help
%menu_end

```

Using Nested Menus

```
!  
!-----  
!                               Level 2 Menu 6 Set/Show alt Port Param  
!-----  
!  
%menu_start 6 Alt Port  
%menu_entry 1  
Show Alternate parameters  
show port alt characteristics  
%menu_entry 3  
Set Backspace character  
set port line editor backspace ^h  
%menu_entry 4  
Set Delete Beg character  
set port line editor delete beginning ^u  
%menu_entry 5  
Set End of Line character  
set port line editor end ^e  
%menu_entry 6  
Set Previous Line character  
set port line editor previous line ^p  
%menu_entry 7  
Set Quoting character  
set port line editor quoting character ^v  
%menu_entry 8  
Set Cancel character  
set port line editor cancel ^z  
%menu_entry 13  
Set Forwards character  
set por line editor forwards ^f  
%menu_entry 14  
Set Delete Line character  
set port line editor delete line ^k  
%menu_entry 15  
Set Begin Line character  
set port line editor beginning ^a
```

```
%menu_entry 16
Set Next Line character
set port line editor next line ^n
%%menu_entry 17
Set Insert Toggle character
set port line editor insert toggle ^w
%menu_entry 18
Set Redisplay character
set port line editor redisplay ^r
%menu_entry 20
Help
help
%menu_end
!
!-----
!                               Level 2 menu 7 Set/Show Telnet Parameters
!-----
!
%menu_start 7 Telnet Parmns
%menu_entry 1
Show Telnet Parameters
show port telnet parameters
%menu_entry 3
Set Abort Output character
set port telnet abort output #
%menu_entry 4
Set Attention character
set port telnet attention #
%menu_entry 5
Set Echo Mode
set port telnet echo #
%menu_entry 6
Set Erase Keystroke
set port telnet erase char #
%menu_entry 7
Set Erase Line Character
```

Using Nested Menus

```
set port telnet erase line #
%menu_entry 8
Set Terminaltype
set port telnet terminaltype #
%menu_entry 9
Set TN3270 TranslationTable
set port telnet TN3270 trans #
%menu_entry 13
Set Newline Filtering
set port telnet newline filtering #
%menu_entry 14
Set Query character
set port telnet query char #
%menu_entry 15
Set Remote Port
set port telnet remote #
%menu_entry 16
Set Synchronize character
set port telnet synch #
%menu_entry 17
Set Transmit Mode
set port telnet transmit #
%menu_entry 18
Set Session Mode
set port telnet binary session mode #
%menu_entry 19
Set TN3270 Device
set port telnet TN3270 device #
%menu_entry 20
Help
help
%menu_end
!
```

```

!-----
!                               Level 2 Menu 8 Set/Show TN3270 Parameters
!-----
!
%menu_start 8 TN3270
%menu_entry 1
Show Telnet Parameters
show port telnet char
%menu_entry 2
Show keymap
show port keymap
%menu_entry 4
Set TN3270 Device
set port telnet TN3270device #
%menu_entry 5
Set TN3270 TranslationTable
set port telnet tn23170 trans #
%menu_entry 20
Help
help
%menu_end
!
!-----
!                               Level 2 menu 9 Set/Show PPP Parameters
!-----
!
%menu_start 9 PPP
%menu_entry 1
Show Parameters
show port ppp cha
%menu_entry 2
Show Status
show port ppp status
%menu_entry 3
Show Counters
show port ppp counter

```

Using Nested Menus

```
%menu_entry 4
Show IP Parameters
show port ppp ip cha;%menu_wait;sho port ppp ip counters
%menu_entry 6
Set Active Mode
set port ppp active enabled;%menu_noprompt
%menu_entry 7
Set Default Settings
set port ppp defaults enabled;%menu_noprompt
%menu_entry 8
Set Active Timer
set port ppp timer #
%menu_entry 9
Set Configure Limit
set port ppp configure limit #
%menu_entry 10
Set Failure Limit
set port ppp failure limit #
%menu_entry 11
Set CharacterMap
set port ppp charmap #
%menu_entry 12
Set Broadcast Range
set port ppp ip broadcast #
%menu_entry 13
Set VJ Compression
set port ppp ip vj compression #
%menu_entry 16
Enable PPP
set port ppp enabled
%menu_entry 20
Help
help
%menu_end
!
```

```
!-----  
!                                On the Level - Misc Parameters  
!-----  
!  
%menu_top t Top Menu  
%menu_up u Up a Menu Level  
%menu_logout x Logout  
%menu_repaint r Refresh  
%menu_continue  
CR to continue...
```


Chapter 7

Using Scripts

The Network Command Script feature enables you to create a script file that contains one or more commands, and to store the file on a host computer called a script server. You can configure an access server port to request the script file and execute the commands in it as soon as a user logs on to the port; or, you can allow the user to request the script file.

You use a text editor to create script files on the script server. The script server can be a host system that supports the Trivial File Transfer Protocol (TFTP) or a Xyplex device that can load files over a network.

This chapter explains how to use the script feature, how to create scripts on a script server, and how to set up the access server to use scripts.

- How the Script Feature Works
- Setting Up the Script Server
- Setting Up the Access Server to Use Scripts
- Sample Scripts

How the Script Feature Works

When a user logs on to an appropriately configured port, the server requests a script file for that user from each available script server. If the script is available at more than one script server, it obtains the file from the first script server that responds. If no script file for a given user is found, the server searches for a generic script file. Upon receiving the complete script file, the access server executes the commands within the file and completes the login sequence. You can configure ports to require a script file in order to complete the login sequence, or to log out the user after a suitable waiting period if a file cannot be found.

If the port is a “dialback” modem port, you can also create a separate dialback script. This script specifies the telephone number to dial when a specific user attempts to log on to the server through a modem. If the access server cannot find a script file for that user, it will not allow the user to log in. If it does find a script file for the user, the server instructs the modem to dial the user back at a designated telephone number. You can use the dialback script together with a login script for dialback ports.

The network command script feature provides great flexibility to users and to the access server manager:

- It allows users to “transport” their port configuration. This is especially useful when configuring a port with special attributes; for example, special TN3270 keyboard mapping.
- Network managers can reboot access servers with a single command, and use system set-up scripts — which can include welcome messages — from a central location.

-
- It provides a method of login control and security. For example, you can configure a port to execute the script when a user attempts to log on. If the user enters an incorrect username, the port will not find the script, and consequently will not log on the user.
 - The command script feature is particularly useful for controlling connections made through modems, since the software uses the network command script feature to support dialback operations.

Setting Up the Script Server

Follow these steps:

1. Determine which UNIX hosts or Xyplex Access Servers are to serve as script servers. Each access server can have up to four script servers. You can use two or more hosts as backup script servers; these can be a combination of dial-back or login script servers.
2. Set up directories to contain script files on each script server. For a UNIX-based script server, you must follow the TFTP guidelines in the section “[Directory Requirements](#)” (page 183).
 - a) For each user who needs a custom login or dialback script file, create a directory to contain the files. The directory name must match the name that the user enters when logging on to the port (at the Enter username> prompt). At a script server, all username directories must be located in the same top-level directory. For ease of use, you can create a top-level directory for script files only, rather than using a directory that already contains many files, such as /usr, /bin, /tftpboot, or /etc on a UNIX host.

- b) If two or more users share a single script, you do not need to create individual directories for them. You can store the script in a common directory. To use a combination of custom and common script files, place custom files in the individual user directories, and place the common file in the top-level directory.

Figure 18 shows how you can set up the directories to contain script files at a UNIX host. In the figure, the user jsmith has a custom login script file. When a user logs in as jsmith, the access server first requests the file `/tftpboot/SCRIPTS/jsmith/login` from the script server.

If the script server cannot find this file, it searches for the file `/tftpboot/SCRIPTS/login`.

For users who log in at a port that is configured to use scripts, but for whom no individual directory (such as the `/jsmith` directory) can be found, the server uses the script file `/tftpboot/SCRIPTS/login`, after the search for a custom script file has failed. Refer to [“Script File Execution and Processing”](#) (page 189) for more information about this process.

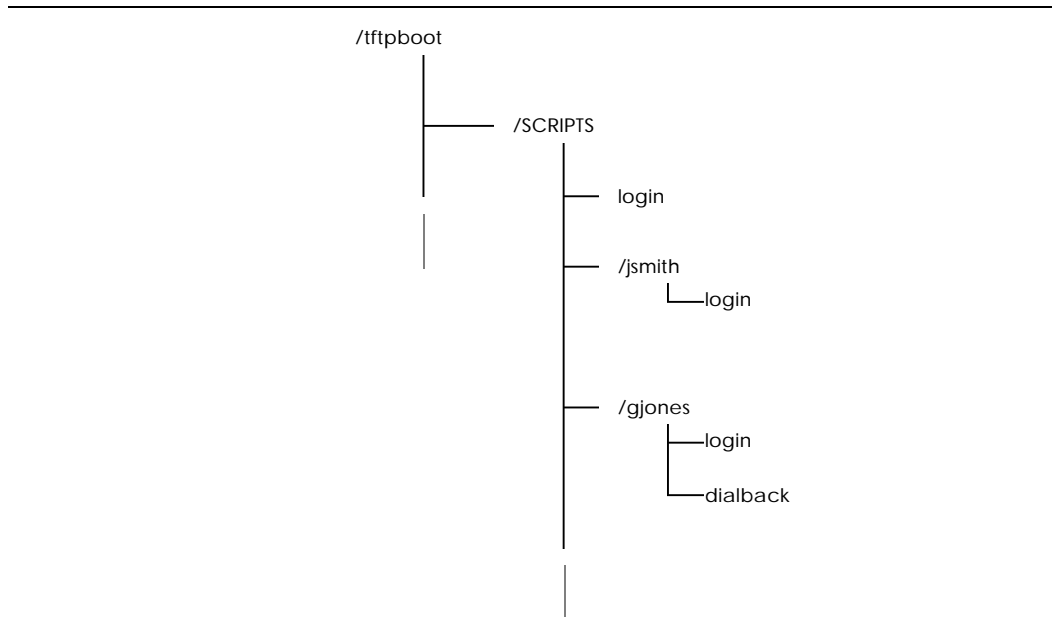


Figure 18 - Sample Script Server Directory Structure

The user “gjones” has both a login script file and a dialback script file. The dialback script contains information that tells the modem which telephone number to dial when a user named gjones attempts to log on to the server through a modem. If gjones’ modem answers when dialed, the user goes through the login routine, and the login script is also run.

Note that if a user logs in as gjones to a non-dialback port that is configured to use a login script, the script server sends the file `/tftpboot/SCRIPTS/gjones/login` to be run at the port.

In order for the server to locate a user’s custom script file, the port username must match the directory name at the host. Therefore, to use their custom login scripts, users must type the correct username when logging in.

Using Scripts

For example, to create a top-level directory named `SCRIPTS`, and a username directory for a user whose login name is “jsmith”, you would type the following commands:

UNIX Host

```
% mkdir SCRIPTS
% cd SCRIPTS
% mkdir jsmith
% cd jsmith
```

MAXserver 1820

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

```
B:> mkdir SCRIPTS
B:> cd SCRIPTS
B:> mkdir jsmith
B:> cd jsmith
```

For additional users, the directory for each username would be a subdirectory of the `SCRIPTS` directory.

Create the Script File

At the UNIX host or PC, use a text editor to create the script file. Use the name “login” for script files that run when a user logs on to a port. Use the name “dialback” for script files that contain dialback instructions. At a UNIX host, the filename must consist of all lower-case letters.

Follow the guidelines in the next section when developing a script file. The section “[Sample Scripts](#)” (page 192) shows examples.

Writing the Script

Observe the following rules when writing a script:

- The first line in the script file must begin with **#control_script** on the first column of the line.
- You can use most access server commands in a script. Use the same command syntax that you would enter with the command interface.

Exceptions: You cannot run another script file within a script file (“nested” scripts are not allowed). You cannot specify menu commands within a script file.

- You can include commands that require user input, such as commands that require a password. The access server will prompt the user for the password or other input before continuing. The interface displays the user prompt regardless of the setting of the Port Script Echo characteristic.
- You can set a port to privileged mode in two ways: (1) by including the `SET PRIVILEGE` command and displaying the privilege password prompt to the user. When the user enters the correct password, the port becomes privileged. (2) by including a line in the form:

```
set priv privilege-password
```

The command **set priv** must be in lower case letters; begin on column 1 of the script file; and contain exactly one space between the keywords `set` and `priv`. Note that the `set priv` command and *privileged-password* are not displayed, even if the Port Script Echo setting is enabled.

Using Scripts

- Each line of a script file can contain only one command with up to 132 characters. Each command must be on one line.
- You can include a Connect command in a script file. For clarity, Xyplex recommends that you include Connect commands at the end of the script file.

NOTE: When using edlin to create a script file for use on an MX-1800/1820 diskette, note that the default end-of-file, or EOF indicator (^Z or A1) causes a syntax error when the script executes. When you write the script file to the diskette, delete the EOF indicator from the file.

The # Character

Within command scripts, the access server recognizes the pound (#) character as a flag for special operations. When the pound character is the first character on a command line, the server attempts to treat the contents of the line as control information, which it must interpret. When the # character is followed by anything other than a script keyword, it indicates a comment. The server ignores the remainder of the line.

echo

When the word “echo” follows the # character, it indicates that a text message or a blank line is to be displayed to the user. The access server displays all characters after **#echo** when the user executes a script. For example, the *text-string* in the following line appears on the terminal screen when the user executes the script.

```
#echo text-string
```

The phrase **#echo** must be in lower case letters. The *text-string* is displayed to the user, even if the Port Script Echo setting is disabled. If the *text-string* consists of one or more spaces, a blank line is displayed to the user.

modem

When the word modem follows the # character in a dialback script, it specifies a modem command that the server must pass on to a modem. The phrase **#modem** must be in lower case letters. All white space following **#modem** is removed before the server passes the text on to the modem.

pause

When the word `pause` follows the `#` character, it causes the script to pause its execution for a specified number of seconds. This feature is useful in dialback scripts. The following command line causes the script to pause for five seconds:

```
#pause 5
```

priv

When the word `priv` follows the `#` character in a dialback script, it specifies that the rest of the script is to be run in privileged mode. (You do not need to enter the privilege password.) After the script executes, the port returns to the mode it was in before the script ran.

Directory Requirements

The script server downloads files to the access server through TFTP. Typically, UNIX systems require that you store all files to be transferred via TFTP in the TFTP “home directory.” Most UNIX systems provide a mechanism for specifying the TFTP home directory, or a default home directory. The default TFTP home directory varies from system to system. Follow the configuration instructions for the TFTP daemon (tftpd) in your UNIX system’s documentation (e.g., MAN pages), 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 defined in the `/etc/inetd.conf` file, and that the factory default home directory is `/tftpboot`. Therefore, you would check the tftp entry in the `/etc/inetd.conf` file to see if the system is using the default home directory or a user-specified home directory. Place the script files in the home directory.

Using a Link

To simplify configuration, such as adding users, or to prevent the TFTP home directory from becoming cluttered, you can place script files in a directory other than the TFTP home directory. To do this, create a link from the directory containing the script files to the TFTP home directory, so that the TFTP daemon knows where to locate the files. Give the link suitable file permissions using commands in this form (note that you must be superuser):

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

Using Scripts

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 these commands:

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

TFTP Security Mechanisms

Your UNIX system might have TFTP security mechanisms that limit TFTP to specific directories. For example, SunOS™ uses a TFTP daemon `-s` (for secure) option that restricts TFTP access to a specific directory and its subdirectories. Sun Workstations are normally configured with this option enabled. An entry similar to **`-s /tftpboot`** in the `tftpd` entry appears in the `/etc/inetd.conf` file. Read the MAN pages for `tftp`, `tftpd`, and `inetd.conf` to determine the directory/security requirements on your UNIX system.

Setting Up the Access Server to Use Scripts

Specify the locations where the access server can request script files, using an IP address or domain name, and the directory locations where the files reside. Also specify which ports use or require a script file for login.

1. Make sure that Telnet is enabled, and that you have defined the server's IP address. Also make sure that you have defined the address of the domain name server, if the server accesses the script server through a domain name.
2. Define one or more script servers:

```
DEFINE/SET SERVER SCRIPT SERVER [domain-name "directory-path"]  
                                [ip-address "directory-path"]
```

Note that you must define a path even if the script resides in the default directory, /tftpboot. When this is the case, specify the null path, “ ”.

Examples:

```
Xyplex>> define server script server host.xyplex.com " "  
Xyplex>> define server script server host.xyplex.com  
          "scripts"  
Xyplex>> define server script server 192.168.19.101  
          "scripts"
```

You can define up to four script servers.

3. Where applicable, specify whether the ports require the access server to download and execute the script in order to complete the login sequence. If the port requires the script at login time, it will not complete the login sequence if the access server cannot find the script.

```
DEFINE/SET PORT  [port-list] SCRIPT LOGIN  [DISABLED]
                                     [ALL]                [ENABLED]
                                               [REDUIRED]
```

Example

```
Xyplex>> define ports 2-5 script login required
```

If a port does not require the script, it completes the login sequence even if the access server cannot find the script. The following command enables the login feature, but does not require it on Ports 6-8:

```
Xyplex>> define ports 6-8 script login enabled
```

If neither of the preceding examples applies, the user can login normally and run the script by issuing the Script command. *This is the default for all ports.*

Define/Set Port Script Echo

Use the following command to specify whether a port displays the commands in the script file as it executes them. This setting is Disabled by default.

```
DEFINE/SET PORT  [port-list] SCRIPT ECHO  [ENABLED]
                                     [ALL]                [DISABLED]
```

Script, Define/Set Port Script

These commands allow users to specify the name and location of a script file to be run. The Set Port Script command is the same as the Script command. It is typically used with TSM. The “*script-name*” can include up to 64 characters.

```
DEFINE/SET PORT [port-list] SCRIPT "script-name"
                [ALL]
SCRIPT "script-name"
```

Examples

```
Xyplex>> define port 5 script "/usr/login"
Xyplex> script "usr/login"
```

Clear/Purge Server Script Server

These commands remove a UNIX host or Xyplex script server from the access server's databases:

```
CLEAR/PURGE SERVER SCRIPT SERVER [entry]
                                   [ALL]
```

An entry is an entry number in the List/Show/Monitor Server Script Server display. (See Figure 19.)

Show/List/Monitor Server Script Server

These commands show information about the script servers that are available for the unit. Figure 19 shows a sample display:

```
Xyplex> SHOW SERVER SCRIPT SERVER
TS/720 V6.0S65 Rom 470003 HW 00.02.00 Lat Protocol V5.2 Uptime: 5 03:55:21
Address:08-00-87-02-34-56   Name:X023456           Ethernet:A   Number:    0

Script Servers:

Entry 1: 192.168.240.183   /
Entry 2: 192.168.248.225   /
```

Figure 19 - Server Script Server Display

Script File Execution and Processing

The access server runs a script when a user logs on to a port, or when a user issues the Script command. The following events occur when the server runs the script file:

1. The user attempts to log on to a port where the Script Login setting is either Enabled or Required. The user enters a login username when the Enter username> prompt appears.

If the port is configured for dialback, the connection is immediately broken. The server saves the port username — in order to locate the script file and authenticate the user — and also saves the port speed.

2. The server makes a request to the TFTP process (e.g., a UNIX TFTP daemon) at each script server, to download a specific script file. The file to be downloaded is determined by the server, as follows:
 - If the script is being run when a user logs on to a server port, the access server requests a script file named **login** from a directory location that is based on two items: (1) the *pathname* specified in the Define/Set Server Script Server command and (2) the username of the port, with blank spaces removed. The Define/Set Server Script Server command designates the top-level directories to be searched. An example follows:

```
Xyplex>> define server script server 192.168.11.101  
          "scripts"
```

If a user named "John A. Smith" logs on to a port, the server requests the script file /usr/xyplex/JohnA.Smith/login from the script server at address 192.168.11.101.

If the TFTP process does not find the script at the specified location, it searches the directory immediately above the specified one.

- If the script is being run as the result of a user request, the name and location of the script file is supplied by the user in the Script or Set Port Script command. If the user does not specify a script to be run, the server requests the script that would normally be used at port login.
- If the script is being run at a dialback port, the server requests a script file named **dialback**. The directory location at the script server is the same as for a login script.
- If the requested file is found at a script server within 30 seconds, the script server downloads the script file to the server through TFTP.
- If the requested file is not found at any script server within 30 seconds, the action that the server takes depends on the value of the Port Script Login setting and, for dialback ports, the value of the Port Dialback setting.
- If the Port Script Login setting is set to Required, the server logs out the port. If it is set to Enabled, the port is logged on.
- If the Port Dialback setting is set to Enabled, the server logs out the port.

3. After the requested script file is found at the script server, the file is completely read into memory by the access server, before it is run.
 - When a script file contains a Connect command, the access server initiates the command, runs the remaining commands within the script, and logs on the port. After script finishes running, the access server completes the Connect command.
 - A dialback port passes the dialing information to the modem, which then dials the remote modem. The remote modem has only a limited time in which to respond (based on the Port Dialback Timeout setting). If the remote modem does not respond within this interval, or if the line is busy, the server logs out the port and drops the connection. If the remote modem responds within the required interval, the server begins the normal login sequence.

When the Enter username> prompt appears again, the user must type the same name that he typed originally (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 requests and runs the script.

- You can use Kerberos and other security features to provide additional security.

Sample Scripts

1. This login script temporarily defines a dedicated service. You might set up this script for a user who has only limited experience with the access server. The script automatically sets Privileged mode so that the Define command can run; it then resets the mode to Nonprivileged.

```
#control_script
# This script enforces a dedicated service
#PRIV
DEFINE PORT TELNET DEDICATED SERVICE FINANCESUN
SET NOPRIV
```

2. This login script defines TN3270 settings. It enables a user to “carry” the settings to other ports. The script requires the user to enter the privileged password.

```
#control_script
# Use this script to set up a access server for TN3270.
# TN3270 protocol must be enabled on the access server.
#
# Set up TN3270 port characteristics.
#PRIV
SET PORT TELNET TN3270 DEVICE VT100
SET SERVER INTERNET PRIMARY GATEWAY ADDRESS 192.168.1.1
SET DOMAIN VMS 192.168.1.101
SET NOPRIV
CONNECT VMS
```


3. This dialback script contains dialing information for a modem:

```
#control_script
# This is a dialback script.
#modem atdt5551978
#pause 3
#modem atdt33
```

4. This login script displays a system welcome message. The script uses the echo feature and a control character in an echo line.

```
#control_script
#echo welcome to Company X <Bel>
#echo
#echo The laser printer is down today.
```


Chapter 8

Using the Accounting Feature

This chapter explains how to use the Accounting feature. It covers the following topics:

- Enabling the accounting feature
- Enabling the syslogd daemon
- Information in the Account Log
- Associated commands

When the Accounting feature is enabled, the access server records information about successful and attempted connections made to or from its ports, as well as information about disconnected sessions. It stores this information in a log file.

The following accounting features require that the access server have at least 2 MB of memory and be running appropriate software:

- syslogd daemon
- Verbose Accounting mode
- Verbose Priority number
- Clear Server Accounting command.

The load images for Network 9000 Access Server 720 modules and MAXserver 1620/1640 access servers support all these features. Refer to the *Access Server Software Release Notes* for information about the appropriate load images for other access server types.

Enabling the Accounting Feature

Complete these steps:

1. Log on to an access server port and set privileged mode.
2. Specify the maximum number of accounting entries that the access server may store. This enables the accounting feature. For example, the following command specifies a maximum of 1000 accounting entries and enables the accounting feature:

```
Xyplex>> define server accounting entries 1000
```

The server displays a message (number 705 or 708) that indicates how many bytes of memory remain after the feature is enabled, or which are needed to enable the feature.

Specifying 0 entries disables the Accounting feature.

3. Reboot the access server. You can either push the Reset button on the front of the access server or enter the following command:

```
Xyplex>> init delay 0
```

4. To verify that Accounting is enabled after the access server boots, check the Show Server Characteristics display.

Enabling the `syslogd` Daemon

You can use the `syslogd` daemon to log accounting information to a remote UNIX host. The daemon supports both normal and verbose accounting entries.

As the access server logs accounting entries, it sends a message to the UNIX host that you specify in the command line. The `syslogd` daemon intercepts the message and routes it to one or more destinations, depending on the settings in the `/etc/syslog.conf` file on the host. For example, the host might display output on a console screen, or write the information to a log file.

To change the status of `syslogd`, use the Define Server Daemon Syslogd Enabled/Disabled command. The setting is disabled by default. When you enable it, you must specify the IP address of the host where the remote account log is to reside:

```
Xyplex>> define server daemon syslogd enabled 172.19.24.81
```

Reboot the access server for the command to take effect. When you disable the daemon, you do not need to specify an IP address.

Information In the Account Log

The access server can store information in two types of account log:

- Default account log
- Verbose account log

The default log is enabled automatically when you enable the accounting feature. You enable verbose accounting through a separate command. (Refer to [page 200](#).)

The Default Account Log

The default account log records the following information:

- The port numbers where connections were made or attempted
- The usernames of ports
- The sources of the connections, whether local or remote, and the network protocols that the connections used
- The destinations of the connections
- The time when connections were first made
- The times when connections were terminated
- The reasons why sessions were terminated
- The amount of data received from the connected devices
- The amount of data sent by ports to the connected devices

You can use this information for purposes such as:

- Identifying when and where a connection attempt occurred. This information can help determine if your network has a security problem, such as “hackers” attempting to log in.
- Identifying why the access server is rejecting connections, for troubleshooting purposes.
- Providing a record for “billing” time or services. This use requires a host-based application that can collect and store the accounting information from the server.

The access server can store up to 1000 session accounting entries. When the account log reaches its maximum size, the server discards the oldest entry each time it logs a new one. Depending on the number of entries in the log file, the accounting feature can use up to 210 KB of memory — as it does in a log file containing 1000 entries.

Memory Considerations

Before you enable this feature, be sure that the access server has adequate memory to support the accounting log. Some access servers have limited memory available. Enabling the feature on these servers limits the number of other features that you can use. Similarly, enabling other features can affect the amount of memory available for the Accounting feature.

Sample Default Account Log

When the Accounting feature is enabled, the server stores one accounting entry for each connection that is attempted — whether successful or not — and for each time a session is disconnected. Figure 20 shows an example of an accounting log displayed in 132 column mode:

ENTRY	ADDRESS	PORT	USERNAME	TYPE	DESTINATION	CONNECT TIME	DISCONNECT TIME	BYTES IN	BYTES OUT
1	08-00-87-00-4F-45	8	TN3270user	-Lte	140.179.80.75	25 May 1996 15:38:33			
1	08-00-87-00-4F-45	8	TN3270user	-D 0	140.179.80.75	25 May 1996 15:38:33	25 May 1996 15:40:55	13	92
2	08-00-87-00-4F-45	1		-Lte	140.179.80.31	25 May 1996 15:39:54			
2	08-00-87-00-4F-45	1		-D 0	140.179.80.31	25 May 1996 15:39:54	25 May 1996 15:40:57	13	92
3	08-00-87-00-4F-45	2		-Lte	140.179.80.32	25 May 1996 15:39:54			
3	08-00-87-00-4F-45	2		-D 0	140.179.80.32	25 May 1996 15:39:54	25 May 1996 15:40:58	13	92
4	08-00-87-00-4F-45	3		-Rte	140.179.80.33	25 May 1996 15:39:54			
4	08-00-87-00-4F-45	3		-D 0	140.179.80.33	25 May 1996 15:39:54	25 May 1996 15:40:59	15	92

Figure 20 - Sample Accounting Display

Enabling the Verbose Account Log

The Verbose account log records all the information included in the default account log, plus the following details:

- Messages from UNIX daemons (`lpd`, `rwhod`, `fingerd`, and `routed`) that are enabled.
- ARAP, PPP, SLIP, and CSLIP activity, if these protocols are enabled.
- Information about syntax errors in the Nested Menu file, if one exists. The log also indicates that the access server did not find the nested menu file when it booted, if it was unavailable.

Verbose accounting mode also displays information such as the protocol types of sessions, in more detailed form than the default log. For example, the verbose log displays “Rtelnnet” to indicate a remote Telnet session, rather than “Rte,” which appears in the default account log.

You enable or disabled Verbose Accounting through the following command. You do not have to reboot the access server if you issue the Set command.

```
DEFINE SERVER VERBOSE ACCOUNTING [ENABLED]  
[DISABLED]
```

Figure 21 shows an example of verbose output:

```
ACCOUNTING SUMMARY/SYSTEM LOG (ENTRIES WILL LOG AT OR BELOW PRIORITY LEVEL: 7  
  
22 Jun 1996 13:49:39 RWHOD Message from chris, IP Addr 140.179.240.254  
22 Jun 1996 13:50:44 source:08-00-87-01-59-5A dest: 140.179.248.81 port:16 user:john smith type:Ltelnnet  
22 Jun 1996 13:50:56 source:08-00-87-01-59-5A dest:140.179.248.81 port:16 user:john smith type:D  
reason:0 bytes in:37 bytes out:168  
22 Jun 1996 13:51:26 source:08-00-87-01-59-5A dest:140.179.248.81 port:16 user:john smith type:Ltelnnet  
22 Jun 1996 13:52:58 source:08-00-87-01-59-5A dest:140.179.248.81 port:16 user:john smith type:Lrlogin  
22 Jun 1996 13:54:12 source:08-00-87-01-59-5A dest:140.179.248.81 port:16 user:john smith type:D  
reason:0 bytes in:45 bytes out:111
```

Figure 21 - Verbose Account Log

Verbose Priority Number and Log File Location

Messages from UNIX daemons have associated priority numbers, which are based on the severity (importance) of the message. For example, a Priority 0 message has high severity, and can indicate an abnormal system shutdown. A Priority 6 message is normal and typically indicates status information.

Using the Accounting Feature

You can specify the message types that the log file accepts — based on severity — by assigning a priority number to the account log. The server then logs only those messages that have a priority number equal to or below the number you specify. Table 16 lists the message types and their priority numbers. The default priority number is 5.

You can also specify, through the same command, whether the UNIX host logs accounting information to a file in the kernel or to a UNIX facility. By default, the host logs the information to the kernel. If you want the host to log it to a facility instead, you must set up the facility on the host and specify it at the access server. Refer to the MAN pages on the UNIX host for instructions.

Table 16 - Priority Numbers for Messages from UNIX Daemons

Message Type	Priority	Description
LOG_EMERG	0	A severe condition. The access server usually broadcasts a priority 0 message to all users because it can affect their ability to work at the host.
LOG_ALERT	1	A condition that the system manager needs to correct immediately, such as a corrupted system database.
LOG_CRIT	2	A critical condition, such as a hard device error.
LOG_ERR	3	A software error condition.
LOG_WARNING	4	A warning message.
LOG_NOTICE	5	Conditions that are not error, but which might require specific procedures to adjust them.
LOG_INFO	6	Normal, informational messages.
LOG_DEBUG	7	Messages that contain information useful for testing only.

Use this command to specify a priority number:

```
DEFINE/SET SERVER VERBOSE PRIORITY number LOG FACILITY [USER  
number]  
[LOCAL number]
```

To use this command, Verbose Accounting must be enabled. The following example sets the verbose priority to 7, thereby allowing all messages to be logged.

```
Xyplex>> set server verbose priority 7
```

Clearing the Account Log

To clear information from the account log, you can reboot the access server or use this command:

```
Xyplex>> clear server accounting
```

Associated Displays

The following sections describe the commands that you use to enable, disable, and view information associated with the accounting feature.

Show/Monitor Server Accounting

These commands display the accounting information that the access server has logged. Figure 20 on [page 200](#) shows a sample display. Xyplex recommends that you set up your terminal to display 132 characters per line.

Show/List/Monitor Server Characteristics

These commands display the maximum number of accounting entries that the server can store, and indicate whether verbose accounting is enabled.

Show/List Unit and Show/List/Monitor Server Alternate Status

These displays indicate:

- Whether SYSLOGD is enabled
- The IP address of the UNIX host where the account log resides
- The location of the account log on the host

Chapter 9

Maintaining Boot Records

This chapter describes commands that change settings in the Initialization Records (also called “boot records”) of access and printer servers that have internal, NonVolatile Storage (NVS). These commands provide an alternative to the Initialization Configuration menu for changing these settings.

The MAXserver 1608A, 1620, and 1640 Access Servers have three Initialization Records and support the commands covered in this chapter.

NOTE: Network 9000 Access Server 720 modules also have three Initialization Records. The commands for changing these records are described in the manual *Managing Network 9000 Modules and Power Supplies*.

This chapter describes each setting in an Initialization Record and explains how to change it. It also includes information about how to reset all initialization settings to defaults. The chapter covers the following topics:

- Viewing initialization parameters
- Status of MAXserver 1608A/1620/1640 Initialization Records
- Enabling and disabling protocols
- Resetting parameters to defaults

Viewing Initialization Settings

You can view the settings in an Initialization Record by issuing the List/Monitor Server Loaddump Characteristics command. On MAXserver 1608A/1620/1640 units you can specify the primary (Record 1), secondary (Record 2), or tertiary (Record 3) Initialization Records; the primary record is the default. Figure 22 shows a sample display for a MAXserver 1620:

```
Xyplex> LIST SERVER LOADDUMP PRIMARY CHARACTERISTICS

MX1620 V6.0      Rom 460000 HW 00.00.00 Lat Protocol V5.2 Uptime: 21 06:24:56
Address:  08-00-87-03-45-67  Name:    X034567                Number:    0

Primary record: Enabled

Internet Address: 192.168.181.199

Internet Load Host: 192.168.240.183
Internet Load Gateway: 0.0.0.0
Internet Load File: xpcsrv20.sys
Internet Delimiter:  None

Software: XPCSRV20

Image Load Protocols Enabled: CARD, DTFTP, RARP

Dump Protocols Enabled: XMOP, MOP, BOOTP, RARP

Parameter Load Protocols Enabled: NVS
```

Figure 22 - Server Loaddump Primary Characteristics - MX-1620

The following sections describe the settings in each field of the display, the possible values for each setting, and how to change them.

Status of MX-1608A/1620/1640 Initialization Records

A MAXserver 1608A/1620/1640 maintains three Initialization Records in NVS. The server first attempts to load its software using the information in Initialization Record 1, if it is enabled. If it cannot load successfully, it attempts to load using Record 2, if enabled. If it fails again, it attempts to use Record 3, if enabled. If all three attempts fail, the server begins the sequence again with Record 1.

The Primary Record field in Figure 22 ([page 206](#)) shows the status of Initialization Record 1 — either Enabled or Disabled. By default, only Record 1 on a MAXserver 1608A/1620/1640 is enabled. Initialization records have values for all settings, whether they are enabled or not. The Primary Record field does not appear in the Loadump Characteristics screen for servers with a single Initialization Record.

This command changes the status of an Initialization Record:

```
DEFINE SERVER LOADDUMP [record] [ENABLED]  
                        [DISABLED]
```

The *record* specifies either Primary, Secondary, or Tertiary. The default is primary. An example follows:

```
Xyplex>> define server loadump secondary enabled
```

Issuing this command enables the access server to attempt to load software and parameters using Initialization Record 2, if it fails when using Record 1.

Enabling and Disabling Protocols

The network protocols that are enabled by default for loading and dumping vary with different server models. Table 17 shows the protocols that are enabled by default and the other available protocols:

Table 17 - Protocols for Loading and Dumping

Product	Software Load Image	Parameter File	Dump File
MAXserver 1608A/1620/1640 Access Servers	CARD XMOP MOP BOOTP RARP DTFTP*	NVS XMOP MOP BOOTP RARP	XMOP MOP BOOTP RARP
* Not enabled by default.			

The Show/Monitor Server Status display shows the name and location of the software load host, if the access server obtained the software load image from the network. The Show/List/Monitor Parameter Server display shows the name and location of the parameter server.

By default, a MAXserver 1608A/1620/1640 obtains its software file (load image) from a flash memory card, and its parameters from NVS. If a memory card is not present, or NVS is disabled, the server requests the software load image and parameter files from a Xyplex loader, or one or more hosts on the network. The enabled protocols determine where the access server obtains the files: XMOP indicates another Xyplex loader; MOP indicates a VAX/VMS host; and RARP, BOOTP and DTFTP indicate UNIX hosts.

Each Initialization Record has several network loading and dumping protocols. These protocols are only used when the software load image or parameters cannot be loaded from the flash memory card.

Changing Protocols through Commands

Use this command to enable or disable software and parameter loading protocols:

```
DEFINE SERVER LOAD [record] usage PROTOCOL [protocol] [ENABLED]
                  [ALL]                  [ALL]          [DISABLED]
```

The *record* specifies either Primary, Secondary, or Tertiary. The *usage* variable specifies either IMAGE or PARAMETERS. You cannot disable all protocols in an Initialization Record. An example follows:

```
Xyplex>> define server load primary image protocol xmop
          disabled
```

Use this command to enable or disable dumping protocols (protocols that the server uses when dumping its memory contents):

```
DEFINE SERVER DUMP [record] PROTOCOL [protocol] [ENABLED]
                  [ALL]                  [ALL]          [DISABLED]
```

The *record* variable specifies either Primary, Secondary, or Tertiary. An example follows:

```
Xyplex>> define server dump primary protocol rarp disabled
```

CARD, NVS, XMOP, and MOP Protocols

The access server uses the CARD, NVS, XMOP, and MOP protocols to search for a software load image or parameter file at a particular location, which is determined by the protocol. You can change the load image filename by using the command described in [“Changing the Software Filename”](#) (page 211). The current load image filename appears in the Software field of the Loaddump Characteristics display. The XMOP and MOP protocols can also send a memory dump file to a dump server.

CARD Protocol

The CARD protocol looks for a software load image on a memory card. This is the default load image protocol for these products. This protocol does not apply to dump files, since Xyplex memory cards do not function as dump servers.

NVS Protocol

The NVS protocol looks for a parameter file in NVS. This is the default protocol for loading parameters on all server models. This protocol does not apply to the load image or dump files.

XMOP and MOP Protocols

The XMOP and MOP protocols look for a software load image file, a parameter file, or a dump server. The access server first attempts to use XMOP, then MOP.

The Xyplex Maintenance Operations Protocol (XMOP) looks for the files on a Xyplex loader on the network. Refer to the *Software Installation Guide for Xyplex Loader Kits* for more information about using XMOP.

The DEC Maintenance Operations Protocol (MOP) looks for a software load image and parameter file on a DEC host that runs the MOP protocol. The access server can also use MOP to send information to a dump server. Refer to the *Software Installation Guide for VMS Kits* for information about setting up a MOP host.

Changing the Software Filename

The software load image filename appears in the Software field of the Loadump Characteristics display. For MAXserver 1608A, 1620, and 1640 units, it is XPCSRV20.sys. Use this command to specify a name for the software load image file:

```
DEFINE SERVER LOAD [record] SOFTWARE filename
                  [ALL]
```

The *record* specifies either Primary, Secondary, or Tertiary. The *filename* variable specifies the software load image filename. An example follows:

```
Xyplex>> define server load primary software XPSPECIAL
```

BOOTP and RARP Protocols

The Bootstrap Protocol (BOOTP) and the Reverse Address Resolution Protocol (RARP) look for the software load image and parameter filenames on a UNIX host. The access server first tries BOOTP, then RARP. You do not need to supply any additional information in the Initialization Records if the BOOTP and RARP hosts are configured correctly. The *Software Installation Guide for UNIX Hosts* describes BOOTP and RARP setup.

DTFTP Protocol

The Directed Trivial File Transfer Protocol (DTFTP) searches for a specific software load image file on a UNIX host. You supply the

name of the software load image file, the IP address of the load host, and the IP address of the access server. If the load host exists on a remote branch of the network, you must also provide the IP address of a gateway. Default values do not exist for this information. The access server does not use DTFTP to obtain a parameter file or to send information to a dump server. Unless you use DTFTP, you do not have to specify IP settings in the Initialization Record.

The following commands provide information for loading through DTFTP. Each setting is covered in a subsequent section.

Internet Load Address:

```
DEFINE SERVER LOAD [record] INTERNET ADDRESS ip-address
[ALL]
```

Internet Load Host:

```
DEFINE SERVER LOAD [record] INTERNET HOST ip-address
[ALL]
```

Internet Load Gateway:

```
DEFINE SERVER LOAD [record] INTERNET GATEWAY ip-address
[ALL]
```

Internet Load File:

```
DEFINE SERVER LOAD [record] INTERNET FILE "/path/filename"
```

Internet Address

The Internet address specifies the IP address of the access server, which the load server host uses for DTFTP loading. (This is the address assigned through the Define Server Internet Address command.)

MAXserver 1608A/1620/1640 Access Servers allow you to change the Internet Load Address in the three Initialization Records. Figure 22 (page 206) shows 0.0.0.0 in this field; this is the default address on these units. An example follows:

```
Xyplex>> define server load primary internet address  
          192.168.118.2
```

Internet File

The Internet File field specifies the pathname and filename of the software load image on the Internet Load Host. The access server searches for this filename when it uses the DTFTP protocol to obtain the software load image during initialization. An example follows:

```
Xyplex>> define server load internet file "/usr/xyplex/images/  
      xpcsrv20.sys"
```

Internet Host

The Internet Host specifies the IP address of the host on the network where the software load image file resides. Figure 22 (page 206) shows 0.0.0.0 in this field; this is the default. An example follows:

```
Xyplex>> define server load primary internet host  
          192.168.119.3
```

Internet Gateway

The Internet Gateway specifies the IP address of a gateway on the network, which the access server uses to gain access to the Internet load host. Not all network configurations require a gateway. Figure 22 (page 206) shows 0.0.0.0 in this field; this is the default. An example follows:

Maintaining Boot Records

```
Xyplex>> define server load primary internet gateway  
          192.168.111.5
```

Resetting Parameters to Defaults

The following command resets all initialization settings to defaults. Note that the command is not supported by all server models.

```
DEFINE SERVER LOADDUMP [record] DEFAULT  
                        [ALL]
```

The *record* specifies either Primary, Secondary, or Tertiary.

If you specify Secondary or Tertiary, this command also disables the Initialization Record. (This is because these records are disabled by default.)

Chapter 10

How the Server Obtains the Current Time

Xyplex Access Servers provide a user-settable tool for specifying where the server obtains the current time. The tool is supported by access server units with at least 2 MB of memory. Telnet must be enabled to support the tool.

You can *require* the access server to obtain the time from a specific time server, in which case the server may only obtain the time from that server. Alternately, you can *enable* it to obtain the time from a specific time server, in which case if the server is unable to obtain the time from the specified time server, it tries to obtain it from the first available time server, following this order:

- Kerberos server — through UDP, Port 37 (only if Kerberos is enabled)
- SecurID — through UDP, Port 37 (only if SecurID is enabled)
- XMOP/MOP load server — obtain time from VAX load server
- UDP broadcast (through Port 37)
- Clock starts at 00:00:00

The specific time server option is disabled by default. Use these commands to enable or require it, or to disable it later:

```
DEFINE/SET SERVER TIME SERVER [ENABLED ip-address]  
                                [REQUIRED ip-address]  
                                [DISABLED]
```

How the Server Obtains the Current Time

The *ip-address* that you enter with the **Required** keyword specifies the only time server that the access server may use. If the access server fails to obtain the time from this unit, it will issue time queries every 60 seconds until it is successful. The *ip-address* that you enter with the **Enabled** keyword specifies the “preferred” time server. If the access server fails to obtain the time from this unit within 60 seconds, it will try an alternate server — which typically means broadcasting a time query.

NOTES: Issuing the SET command instructs the access server to issue a time query to the designated time server immediately.

For access servers that run Kerberos or SecurID, set the Time Server address to zero. Otherwise, if you have a Secondary Kerberos/SecurID server defined, the access server will never query the Secondary server.

Example

```
Xyplex>> define server time server enabled 172.19.1.101
```


Obtaining the Time

If the access server has a configured time server, it sends a directed UDP query for TIME service to that unit shortly after booting. If it receives no response, and the Server Time Server setting is Enabled (not Required), the access server attempts to get the time from one of the other servers.

The access server attempts to resynchronize with the designated time server on a daily basis, between midnight and 2:00 AM. Similarly, the access server attempts to resynchronize with Kerberos or SecurID servers daily.

Server Alternate Status Display

The “Time Received From:” field on the Server Alternate Status display indicates the IP address of the time server where the access server obtained the current time.

How the Server Obtains the Current Time

Show Server Display

The Show Server display also indicates the current time server:

```
Xyplex> SHOW SERVER

TS/720 V6.0.1  Rom 4A0000 HW 00.02.00 Lat Protocol V5.2 Uptime: 0 00:26:56
Address:08-00-87-02-34-56   Name:X023456           Ethernet:A   Number:    0

Identification:  Xyplex Terminal Server
Welcome:        Welcome to the Xyplex Terminal Server.

Circuit Timer:      80           Password Limit:      3
Console Port:       0           Queue Limit:        24
Inactivity Timer:   30           Retransmit Limit:   8
Keepalive Timer:    20           Session Limit:      64
Multicast Timer:    30           Software:           XPCSRV20
Node Limit:         100          Identification Size: 63
Textpool Size:      16384        Timezone:           04:00
Accounting Entries: 500          Packet Count:       80
Nested Menu Size:   0           Menu Name:
Userdata Delay:     50

Service Groups: 0
Time Server: 172.19.1.101 Enabled

Enabled Characteristics:
Announcements, Broadcast, Console Logout, Dump, Lock, Parameter Polling,
TFTP Parameters, Proprietary Parameters, TFTP Read Broadcasts, Purge Node,
Verbose Accounting
```

Figure 23 - Show Server Display

Appendix A

Compatibility Issues

LAT Compatibility

The Access Server Software is compatible with LAT Versions 5.1 and greater. It is not compatible with LAT Version 5.0, or earlier versions. LAT Version 5.1 is available with VMS Software Release 4.7, and later releases. For sites running VMS Software Releases 4.4 through 4.6, Digital Equipment Corporation provides support for LAT through a separate layered application product, LATplus.

TCP/IP Compatibility

The access server's TCP/IP software only supports the use of the Berkeley Internet Name Domain Protocol (BIND) for Domain name servers, which map user-specified domain names to IP addresses.

IP Router Compatibility

Xyplex Access Servers support directed TFTP load requests through IP routers that support the Proxy Address Resolution Protocol (Proxy ARP).

Loading Parameters with TFTP Using Backup (.bck) Parameter File

If the server detects that it has received a corrupted (damaged) parameter (.prm) file while loading parameters through TFTP, it will request a backup (.bck) parameter file from the same parameter server. Assuming that the backup parameter file is not corrupted, the server then loads parameters using the backup file.

When this occurs, the Show Parameter Server display shows the IP address of the parameter server in the “Loaded From” field and also indicates “Status: Failed” and “Reason: Invalid” — even though valid parameters were loaded via the .bck file. If this happens, overwrite the .prm file using the .bck file, so that both files contain the correct information. If you change any server settings and reboot the server before overwriting the .prm file with the .bck file, the settings you have assigned will be lost.

Appendix B

DEC Software Installation and Management Tools

You can use standard DEC LAT tools to manage a VAX/VMS load server or parameter server. The tools are described in the following sections. The tools are covered in detail in the documentation supplied by DEC. Refer to the Master Index supplied with your release of VMS software for more information.

DECnet Network Control Program (NCP)

The NCP utility enables you to make connections to the console port of the access server through the Maintenance Operations Protocol (MOP) Remote Console Facility. You can use the NCP utility to define the access server as a DECnet node and to troubleshoot network problems. The NCP utility is supplied as part of the DECnet software.

DSVCONFIG Utility

You can use the DSVCONFIG utility in place of some of the NCP commands when adding an access server to the network. This utility is supplied as part of DECserver software. Note that if you do not have the DSVCONFIG utility, you can use NCP commands instead.

Terminal Server Manager (TSM)

You can use this utility to manage the configuration of all access servers from a central location. TSM enables you to make connections to the console port of the access server through the Maintenance Operations Protocol (MOP) Remote Console Facility. TSM is available from DEC.

LAT Control Program (LATCP)

You use this program to set up and control the portion of the LAT software that runs at host computers/service nodes. LATCP is supplied with the VMS operating system for Releases 4.7 and later. LATCP is also supplied by DEC as part of the LATplus layered product. Typically, you use LATCP to create a service at a host, to which users can then log on, and to start LAT software running at the host. (You can use a DEC-supplied command procedure, LTLOAD.COM for this.)

LATCP is also useful when setting up host-initiated connections to an access server port. (An example of this is setting up a shared printer that is connected to an access server port.) In this case, you use LATCP to create an application port at a host, to map the application port to an access server port, and to control queue operations for the application port.

VMSINSTAL Utility

You use this program to install software on a VAX/VMS load server.