Unicenter

NetMaster Network Management for TCP/IP Administrator Guide

(Incorporating NetSpy and NetMaster Operations for TCP/IP)

Version 6.2

Computer Associates

The Software That Manages eBusiness

Booklist

This documentation and related computer software program (hereinafter referred to as the "Documentation") is for the end user's informational purposes only and is subject to change or withdrawal by Computer Associates International, Inc. ("CA") at any time.

This documentation may not be copied, transferred, reproduced, disclosed or duplicated, in whole or in part, without the prior written consent of CA. This documentation is proprietary information of CA and protected by the copyright laws of the United States and international treaties.

Notwithstanding the foregoing, licensed users may print a reasonable number of copies of this documentation for their own internal use, provided that all CA copyright notices and legends are affixed to each reproduced copy. Only authorized employees, consultants, or agents of the user who are bound by the confidentiality provisions of the license for the software are permitted to have access to such copies.

This right to print copies is limited to the period during which the license for the product remains in full force and effect. Should the license terminate for any reason, it shall be the user's responsibility to return to CA the reproduced copies or to certify to CA that same have been destroyed.

To the extent permitted by applicable law, CA provides this documentation "as is" without warranty of any kind, including without limitation, any implied warranties of merchantability, fitness for a particular purpose or noninfringement. In no event will CA be liable to the end user or any third party for any loss or damage, direct or indirect, from the use of this documentation, including without limitation, lost profits, business interruption, goodwill, or lost data, even if CA is expressly advised of such loss or damage.

The use of any product referenced in this documentation and this documentation is governed by the end user's applicable license agreement.

The manufacturer of this documentation is Computer Associates International, Inc.

Provided with "Restricted Rights" as set forth in 48 C.F.R. Section 12.212, 48 C.F.R. Sections 52.227-19(c)(1) and (2) or DFARS Section 252.227-7013(c)(1)(ii) or applicable successor provisions.

© 2002 Computer Associates International, Inc.

All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.

June 02

Contents

Chapter 1: Setting Up Connection Awareness

About Connection Awareness 1-2
- How Connection Awareness Works 1-2
- How Events and Messages are Passed to NetMaster 1-3
- Events Reported 1-4

Setting Up Stack Connection Awareness 1-5
Setting Up User ID Connection Awareness 1-6
- Generating User ID Connection Awareness with EASINET 1-6
- Generating User ID Connection Awareness with RACF 1-7
- Generating User ID Connection Awareness with Other Exits 1-7

Setting up Application Connection Awareness 1-8
Setting Up Event Processing and Reporting 1-9
- Task 1—Check the Setup 1-9
- Task 2—Implementing System Event Receivers and Log Options 1-9
- Task 3—Implementing Reporting 1-10
- Task 4—Running Self Test 1-12

Chapter 2: Setting Up IP Resource Monitoring

About IP Resource Monitoring 2-2
- System Images 2-2
- Resource Definitions 2-3
- Monitoring Resources in a Multisystem Environment 2-3

Defining a System Image 2-4
Defining Resources 2-4
- Example—Defining the Stack 2-5
- Defining Specific Resources 2-7

Using Templates 2-8

Chapter 3: Setting Up the IP Node Monitor

About the IP Node Monitor 3-2
Setting Up an IP Node Monitor 3-3
Maintaining Monitor Groups 3-5
Controlling Monitor Group Data Collection 3-5
Creating an Attribute for Monitoring 3-5

Chapter 4: Setting Up Performance Monitoring

Reporting Levels 4-2
Restricting the Size of the IPLOG File 4-2
Enabling Performance Monitoring 4-2
Setting Up TCP/IP Logging and Monitoring 4-3
Defining a Report Center or Reporter Data Region 4-4
Configuration A 4-4
Configuration B 4-5
Defining a Report Center 4-5
Defining a Reporter Data Region 4-5

Chapter 5: Setting Up Access Control

About Access Control 5-2
How Access Control Works 5-2
Controlling Access to TCP/IP 5-3
Task 1—Enabling Access Control 5-3
Task 2—Defining TCP/IP Stacks 5-3
Task 3—Defining Ports 5-4
Task 4—Defining Hosts 5-4
Connection Access Checking 5-5
Controlling User Access 5-6
Displaying a User Signon Panel and Application Selection List 5-6
Defining Access to Applications 5-7
Using EASINET with Access Control 5-7
How SAF User ID Checking Works 5-8
Defining a Port to SAF 5-9
Defining a Host to SAF 5-9
Controlling Access to Administration Functions 5-10

Chapter 6: Advanced Configuration Tasks

Enabling Packet Tracing 6-2
Enabling Communications Server Systems 6-2
Enabling TCPaccess Systems 6-4
Starting a Trace 6-5
Enabling Multisystem Support 6-6
Enabling the Management of SNA/VTAM Resources and Sessions 6-7
About the Network Control System (NCS) 6-7
About the Network Tracking System (NTS) 6-7
Specifying Telnet Translation Tables 6-8
Setting up the Web Interface 6-9
Setting Up Internet Explorer 6-9
Setting Up Netscape 6-9
Setting Security Prompts for Web Applications 6-10
Synchronizing Time Zones 6-11
Working with Data Space Definitions 6-11
Listing Data Space Definitions 6-12
Exporting Data Space Definitions 6-12
About SNMP 6-13
About SNMP Security 6-13
Predefining SNMP Community Names 6-14
Using IP Address Ranges and Masks 6-15
About NetMaster Socket Management for CICS 6-15
Configuring NetMaster Socket Management for CICS 6-15
Customizing NetMaster Socket Management for CICS 6-16
Issuing CICS Commands in a Command Entry Environment 6-17
Maintaining Connection List Criteria 6-18

Chapter 7: Setting Up Proactive Monitoring

About Proactive Monitoring 7-2
Event Detectors 7-3
Sample Event Detectors 7-3
Defining an Event Detector 7-4
Defining Criteria for a Detector 7-5
Defining an Alert 7-6
Defining Actions to Be Initiated 7-7
Monitoring Connections 7-9
Criteria for Connection Detectors 7-9
Defining a Connection Detector 7-10

Monitoring Custom Events 7-11
Defining a Custom Event Detector 7-11
Monitoring FTP Failures 7-11
Criteria for FTP Failure Detectors 7-11
Defining an FTP Failure Detector 7-12
Monitoring Listeners 7-12
Criteria for Listener Detectors 7-12
Defining a Listener Detector 7-13
Monitoring Messages—Cisco Channel Card TN3270 Log 7-14
Criteria for Cisco Channel TN3270 Log Messages to Be Monitored 7-14
Defining a Cisco Channel TN3270 Log Detector 7-14
Monitoring Messages—ICMP 7-15
Criteria for ICMP Message Detectors 7-16
Defining an ICMP Message Detector 7-16
Tips for Using ICMP Message Detectors 7-16
Monitoring Messages—OS/390 Console 7-17
Criteria for OS/390 Console Message Detectors 7-17
Defining an OS/390 Console Message Detector 7-17
Tips for Using OS/390 Console Message Detectors 7-17
Implementing a Trouble Ticket Interface 7-18
Defining a Trouble Ticket Interface 7-19
Setting Up the Trouble Ticket Data Definition 7-21
Implementing the Alert History Function 7-22
Reorganizing Files and Monitoring Space Usage 7-22
Applying Alert Monitor Filtering 7-23
Forwarding Alerts 7-24
Implementation 7-24
The SNMP Trap Definition 7-24
$AMEVFWD 7-25
Destination Details 7-25
Filtering Details 7-28

Chapter 8: Initializing Multiple Systems

Configuring Multiple Systems with Initialization Files 8-2
Task 1—Generating an Initialization File 8-2
Task 2—Configuring the Initialization File 8-2
Task 3—Initializing Your Region From an Initialization File 8-4

Chapter 9: Setting Up Security

About Security 9-2
Security Considerations for Existing Users 9-3
Checking Existing User Group Definitions 9-3
Customizing Existing Background User Definitions 9-4
Defining NetMaster Users 9-5
Managing Users in a Multisystem Environment 9-6
Implementing User Definition Synchronization Across Linked Regions 9-6
Synchronization Report 9-7
Troubleshooting 9-7
External Security Packages 9-8

Chapter 10: Troubleshooting

About Troubleshooting 10-2
Performing NetMaster for TCP/IP Self Test 10-2
Accessing Self Test 10-3
Displaying the NetMaster for TCP/IP Initialization Log 10-3
Troubleshooting OS/390 SNMP Problems 10-4
Commonly Encountered Errors 10-6
Interpreting Socket Error Codes 10-9

Appendix A: UNIX Configuration Issues

Defining SNMP Managers to NetMaster for TCP/IP A-2
SNMP Manager Support A-2
Recording an SNMP Manager Definition A-3

Appendix B: Enhanced Automation

$IPCALL and $TNCALL Procedures B-2
$IPCALL Procedure B-2
$TNCALL Procedure B-2
$IPCALL ATTR=DEVLINKS B-3
$IPCALL ATTR=SYSINFO B-4
$IPCALL ATTR=IFS B-5

Setting Up Application Connection Names B-7
Reacting to IP Node Monitor State Changes B-9
Trapping FTP, Telnet, Connection, and Message Events B-10
Sample Code B-10
References B-12

Appendix C: &SOCKET Verbs

About the Socket Interfaces C-2
TCP Sockets C-2
UDP Sockets C-3
NCL Verb Set for PING and TRACEROUTE C-3
Socket Built-in Functions C-4
&SOCKET C-8
System Variables C-10
Sample Code for TCP and UDP &SOCKET Verbs C-11
Examples of Using TCP &SOCKET Verbs C-11
Example of Using UDP &SOCKET Verbs C-13
TCP/IP Vendor Interface Restrictions and Limitations C-13
Communications Server C-13
TCPaccess C-14
Interpreting Socket Error Codes C-14
Interpreting IBM Systems Error Codes C-14
Interpreting TCPaccess Systems Error Codes C-15

Appendix D: SMF Record Structure

General SMF Record Format D-1
Field Identifier D-2
Access Control Audit Log SMF Record Format D-2

Glossary

Index

Chapter

1

Setting Up Connection Awareness

NetMaster for TCP/IP can be set up to record FTP, Telnet, and other TCP connection events. The events can then be logged, reported, or used to provide fast connection lists.

This chapter describes how to implement connection awareness.

This chapter contains the following topics:

- About Connection Awareness
- Setting Up Stack Connection Awareness
- Setting Up User ID Connection Awareness
- Setting up Application Connection Awareness
- Setting Up Event Processing and Reporting

About Connection Awareness

Connection awareness allows you to identify the stacks and applications associated with your TCP/IP system. This information is then used in:

- Online diagnosis—to associate connections with user IDs and applications
- Performance monitoring—to provide statistics on stacks and applications
- Graphical reporting tools such as NetMaster Reporter—to review the performance of your stacks and applications as well as the load on them

To do this you need to name the stacks, users, and applications associated with your TCP/IP system.

How Connection Awareness Works

When a TCP/IP connection is started the TCP/IP software generates an event. The event is intercepted by the system and recorded in the data space. As further events occur, for example a user signs on to a TN3270 session, the record of the connection in the data space is updated with any user ID and/or LU names.

The data space also records the rules that govern:

- Which TCP/IP stacks have their record connection events recorded
- The criteria applied to each connection to determine the application name it relates to

To enable connection awareness you must perform the following tasks:

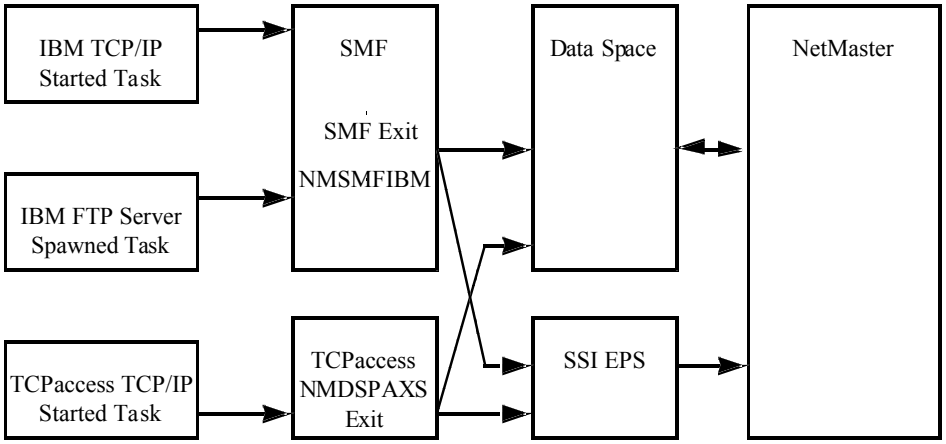
- Ensure that the data space is running before TCP/IP accepts any connections. The data space is only aware of connections that start after it is running.
- Define the stacks that you want to record details for.
- Define the applications that you want to record details for.
- Enable TCP/IP event collection.

The data space is defined during installation. For more details see the *Unicenter Mainframe Installation and Setup Instructions.*

How Events and Messages are Passed to NetMaster

NetMaster for TCP/IP receives events from standard SMF exits and directly from TCPaccess. These events are recorded in the data space. The events in the data space can be accessed by NetMaster.

Selected events can be passed to NetMaster using End Point Services (EPS), a component of Management Services that requires the Sub System Interface (SSI). The event flow is shown below.



Events Reported

The events passed on to your system are determined by the TCP/IP stack that you are using. The table below shows what events are available with each stack.

	Communications Server	TCPaccess
FTP Server Logon fail	Yes	No
FTP Server End of transfer	Yes	Yes
FTP Client End of transfer	Yes	No
Telnet Server Start of session	Yes	Yes *
Telnet Server End of session	Yes	Yes
Telnet Client Start of session	Yes	No
Telnet Client End of session	Yes	No
TCP Connection start	Yes	Yes
TCP Connection end	Yes	Yes

* Requires TCPaccess V5R3 (with PTF TP08546) or later.

Setting Up Stack Connection Awareness

The stack connection awareness function allows you to:

- Give a meaningful name to each stack on your system
- Determine which stacks connection awareness applies to

These names are then used to describe the stack in:

- NetMaster for TCP/IP web applications
- Graphical reporting tools such as NetMaster Reporter
- Connection lists
- Performance monitoring

Stacks are defined to connection awareness by mapping a meaningful name (for example, the stack host name) to the job name and job step of the TCP/IP started task. Stacks that do not have their job name and step name specified are not detected by connection awareness.

To set up TCP/IP stack connection awareness, do this:

1. Enter **/IPADMIN.CS** at the ===> prompt. The Connection Awareness : Stack Software Entries panel is displayed. This panel lists the TCP/IP stacks defined to your system.
2. To add a new stack press F4 (Add). The Connection Awareness : Stack Software Entry Description panel is displayed.
3. Complete the fields on the panel. For a description of the fields press F1 (Help).
4. Press F3 (File) to save the new entry and return to the Connection Awareness : Stack Software Entries panel.
5. Repeat steps 4 to 6 for each TCP/IP stack that you want to define to your system.

Setting Up User ID Connection Awareness

NetMaster for TCP/IP allows you to associate user IDs with connections. This makes problem diagnosis easier. A user is more likely to know their user ID than their IP address.

User ID associations are collected from:

- Telnet connections—if the SOLVE:Access or the NetMaster for TCP/IP access control option is used
- Telnet connections—if installation exits are used
- FTP connections after a file transfer (GET/PUT) on an IBM TCP/IP stack
- FTP connections on TCPaccess

Generating User ID Connection Awareness with EASINET

If you have SOLVE:Access or the NetMaster for TCP/IP access control option, you can use EASINET to generate user ID connection awareness. To do this:

- Make the region the VTAM default application for terminals that connect via TN3270
- Ensure that the region using EASINET is in the same LPAR as the stack that the terminal connects to
- Ensure that the stack is defined in the data space

If you are using SOLVE:Access, ensure that:

- The region is connected to the data space with a Management Services level of at least 4.1. For information the on data space see the chapter 'Advanced Configuration Tasks'.
- The INIT procedure has SYSPARMS IPCHECK=REGISTER

Generating User ID Connection Awareness with RACF

A sample user verification exit (ICHRIX02) for use with RACF is supplied with your system. When there is a logon event under RACF control, the exit passes the user ID and LU name of the logon to your system. The IP connection, user ID, and LU name are then associated in the data space.

To set up RACF to generate user ID connection awareness, do this:

1. Copy the IPRACEAL sample member from the *dsnpref*.IP620.IPSAMP dataset into a source dataset.
2. Submit the IPRACEAL member. This member assembles and link edits IPRACEX2. Follow the instructions in IPRACEAL. You will need to know:
   - The dataset qualifier
   - The NetMaster for TCP/IP version
   - The Management Services version
   - The job name
   - Accounting information
   - The target LPA load library

   The ICHRIX02 member is created in the specified LPA load library where RACF can find it.

   IPRACEX2 is compiled from the IPSAMP dataset. If you make changes to IPRACEX2, change IPRACEAL to compile from your source dataset.

Generating User ID Connection Awareness with Other Exits

To set up CA Top Secret, ACF2, or other session management products to generate user ID connection awareness, do this:

1. Find the exit point of the product where the user ID and LU name are provided.
2. Copy the IPRACEX2 and IPRACEAL sample members from the *dsnpref*.IP620.IPSAMP dataset into a source dataset.
3. Examine the sample members and make changes as necessary to suit the exit point.
4. Refer to the documentation of the target product to enable the exit.

Setting up Application Connection Awareness

The application awareness function allows you to:

- Apply meaningful names for the applications on your system
- Specify the criteria used to match connections to your application name
- Control how connections associated with an application are handled

These names are then used to describe the application in:

- NetMaster for TCP/IP web applications
- Graphical reporting tools such as NetMaster Reporter
- Connection lists
- Performance monitoring

Applications are identified to the system with a set of application name entries. Each entry in the list specifies criteria that must be met by a connection to match it to an application name.

The criteria are ranked by order. As soon as a connection starts it is tested by the first criterion in the list. If the criterion is met, the application name in the entry is applied. If the criterion is not met, the next one is tried. This process continues until a match is found. If a match is not found, the last entry in the list is a default.

To set up application connection awareness, do this:

1. Enter **/IPADMIN.CA** at the ===> prompt. The Connection Awareness : Application Name Entries panel is displayed. This panel lists the applications that have been defined to your system.
2. To add a new application press F4 (Add). The Connection Awareness : Application Name Entry Definition panel is displayed.
3. Complete the fields on the panel. You must specify at least one of the Connection Match Criteria fields. For a description of the fields press F1 (Help).
4. Press F3 (File) to save the new application entry and return to the Connection Awareness : Application Name Entries panel.
5. Repeat steps 2 to 4 for each application that you want to define to your system.

Setting Up Event Processing and Reporting

Setting up event processing enables you to trigger proactive monitoring rules and produce reports on:

- The local network interface workload
- MIB statistics
- FTP events
- Telnet events
- Connection events

Task 1—Check the Setup

1. Check that the SOLVE SSI was implemented during installation and setup. You can test the SOLVE SSI with the self test. For details about the self test see the chapter, "Troubleshooting".

2. Check that your TCP/IP stack is enabled to generate the events and call the exits required for connection awareness. For more information see the *Unicenter NetMaster Network Management for TCP/IP Implementation Guide*.

Task 2—Implementing System Event Receivers and Log Options

There are three options to specify for TCP/IP logging and monitoring. You can:

- Receive events—setting this enables you to receive events and trigger proactive monitoring rules.
- Log events—setting this enables you to log event history in the activity log.
- Report events—setting this enables you to save events in the IPLOG file, run online reports with the web or 3270 interface, and produce history reports.

To implement the receivers and log options, do this:

1. Enter **/ICS** at the ===> prompt. The ICS : Customization Parameters panel is displayed. This panel lists all the parameter groups that set up the characteristics of the region.
2. Press F8 (Forward) until the parameter group $IP IPMONITOR is displayed.
3. Enter **U** in front of the parameter group. The ICS : Initialization Parameters panel is displayed.
4. Specify **YES** or **NO** in each of the event processing fields. For a description of the fields press F1 (Help).
5. Press F3 (File) to save the settings.

Note: The SMF exit for the system event receivers that you have activated must have been be installed during implementation. See the *Unicenter NetMaster Network Performance for TCP/IP Implementation Guide* for details.

Task 3—Implementing Reporting

You can report on the events received by NetMaster for TCP/IP. The events can be:

- Used to generate online and printed reports
- Exported to the IPTREND and IPDETAIL files for reporting with graphical reporting packages
- Exported to an NT server for reporting with NetMaster Reporter

Specifying the Reporting Configuration

To specify the reporting configuration, do this:

1. Enter **/ICS** at the ===> prompt. The ICS : Customization Parameters panel is displayed.
2. Enter **U** in front of the $IP IPFILES parameter group. The IPFILES - TCP/IP File Specifications panel is displayed. This panel specifies:
   - The storage details for your TCP/IP data
   - SMF configuration

   The datasets on this panel were created during the setup process. See the *Unicenter Mainframe Network Management Installation and Setup Instructions* for details.

3. In the IPLOG Event Database entry field, enter a fully qualified dataset name with no quotes to which the following is logged:

   - FTP, Telnet, and API connection events
   - Performance data

   You should monitor the size of the IPLOG dataset. It will be necessary to reorganize the dataset from time to time. To reorganize the dataset, copy the data to a backup file, and delete and redefine the log dataset. A sample reorganize job is provided in *?dsnpref*.IP620.IPSAMP(IPREORG).

   If the file fills, the oldest records are deleted and it is reorganized. The IPLOGSEQ file is used as a backup if this occurs.

4. In the IPDETAIL Extract Dataset entry field, enter the dataset name to which events are to be extracted.

   The IPDETAIL dataset is in a format that can be processed by graphical reporting packages.

5. In the IPTREND Dataset entry field, enter the dataset name to which performance trend information is to be written.

   If you omit the trend dataset name, then no data will be produced for graphical trend reporting.

6. In the IPLOGSEQ Reorg Dataset entry field, enter the dataset name used by the IPLOG reorganization process.

7. In the Field separator character entry field set the character to separate the fields in the IPTREND and IPDETAIL files. The default is a comma (,). In some countries you may need to select another character.

8. In the Days to Keep IPLOG entry field, specify the number of days (between 0 and 7) that reported events are to be kept in the IPLOG Event Database.

9. In the Number of Days to Keep IPDETAIL entry field, specify the number of days (between 0 and 7) that reported events are to be kept in the IPDETAIL extract dataset.

10. Specify the time of day you want the detail and trend report data to be extracted. If this field is left blank extraction only occurs when requested manually.

    At the specified time each day:

    - Data for the number of days specified is extracted from IPLOG and transferred to IPDETAIL. After the transfer, records older than the specified IPLOG retention period are deleted from IPLOG.
    - Data is extracted from IPFILE, consolidated, and transferred to the IPTREND file.
    - The IPDETAIL and IPTREND extract procedures write status and completion messages to the NETWORKIT log.

Your system must be running at the time of the archive for these to be effective.

As well as having data from the log periodically extracted in this way, you can also:

- Extract data from the IPLOG database on an ad hoc basis for processing and analysis. For instructions on extracting data to an external file, see the chapter, "Producing Reports", in the *Unicenter NetMaster Network Management for TCP/IP User's Guide*.
- Extract data to IPTREND at any time by issuing the TRENDEX command from the OCS panel.

11. In the Write Samples to SMF? field, specify NO unless you want to write your own SMF reporting program to use the data collected by NetMaster for TCP/IP. If you specify YES, individual samples and hourly summaries of the IP node monitor and other monitored devices are written to SMF as SMF user records.
12. See the appendix, "SMF Record Structure", for information about the format of the SMF records.
13. If you specified YES in the Write Samples to SMF? field, enter an SMF record identifier in the SMF Record Identifier field.
14. Press F3 (File) to save the settings.

Task 4—Running Self Test

When you have completed the implementation of logging and reporting you should run the self test to ensure that they are working correctly. To run the self test see Performing NetMaster for TCP/IP Self Test in the chapter "Troubleshooting".

Chapter

2

Setting Up IP Resource Monitoring

This chapter describes how to set up the IP resource monitoring facilities available with NetMaster for TCP/IP.

This chapter contains the following topics:

- About IP Resource Monitoring
- Defining a System Image
- Defining Resources
- Using Templates

About IP Resource Monitoring

IP resource monitoring enables you to:

- View performance information about the IP resources in your network
- Perform diagnostics on selected IP resources

Before you can use the IP resource monitor you must have a system image that defines the resources you want to monitor. The express setup facility builds a system image for you by discovering the resources in your environment.

You can specify whether or not performance monitoring is done for each resource defined in the system image.

Performance monitoring uses data sampled at regular intervals. The information retrieved by data sampling is used to:

- Trigger alerts if the monitored performance is outside defined boundaries
- Generate online reports that can be viewed from the IP resource monitor
- Produce historical reports

System Images

The system image represents the set of resources you can monitor and control. Each system image has a name and a version number. You can define multiple system images, but only one system image can be active in a region at a time.

The system image becomes active when it is loaded. A system image is loaded:

- At region startup
- By issuing the LOAD command

During system image load:

- Performance monitoring is started for the resources defined in the system image
- The resources are defined to the IP resource monitor

Resource Definitions

Resource definitions are qualified by:

- The system image name and version
- The resource name and class

The resource classes are:

- ASMON—Address Space Monitor
- CICMON—CICS Resource Monitor (if NetMaster Socket Management for CICS is configured in your region)
- CIP—Cisco Channel cards
- CSM—Communication Storage Manager
- EE—Enterprise Extender
- OSA—Open Systems Adapter
- ROUTER—2216 Routers
- STACK—TCP/IP Stack

Working with Resources

The resource administration facilities allow you to:

- Update, copy, and delete the resources defined by the express setup
- Add new resources
- Define which resources are monitored
- Set the attributes to be monitored for each resource
- Define the type of data to be stored (this affects the reporting methods available)
- Define the conditions that cause alerts to be raised and actions taken

Monitoring Resources in a Multisystem Environment

In a multisystem environment, you can view and perform diagnostics from a single monitor on the resources from the connected systems.

In a multisystem environment, each region must load a different system image. Each resource's system image name is visible on the IP resource monitor. For subordinate regions, the system image name must match the name supplied during the multisystem linking process.

Defining a System Image

Express setup defines a system image to your region. To define a system image manually, do this:

1. Enter **/RADMIN.I** at the ===> prompt. The System Image List panel is displayed. This panel lists the system images defined to your system. You can:
   - Add new system image definitions
   - Browse, change, copy, and delete existing system images
2. To add a new image, press F4 (Add). The System Image Definition panel is displayed.
3. Specify the name of the system image, its version, and a short description of the system image in this panel.

   One system image is required for each region. If you are defining a system image for a subordinate, use the name assigned during the multisystem linking process.
4. Press F3 (File). You are returned to the System Image List panel, and a message is displayed indicating that the system image has been successfully added to the knowledge base.

Defining Resources

Your IP resources are defined by the express setup. After the express setup you can use the resource definition facility to:

- Review the results of the express setup
- Update, copy, or delete existing definitions
- Add new resource definitions

There are five tasks involved in defining your IP resources. Each task is done on a unique panel.

- Monitor General Description—this panel defines the IP resource to your system and determines if monitoring is active
- Monitoring Definition—this panel determines which attributes are monitored as well as the frequency and level of monitoring
- Status Monitor Message Details—this panel enables you to take an action when a specific OS/390 console message occurs
- Automation Log Details—this panel controls the resource transient log
- Owner Details—this panel describes the owner of the resource

Example—Defining the Stack

In this example you will add a TCP/IP stack to your IP resources.

Task 1—Selecting the Stack Resource Class

1. Enter **/RADMIN.R** at the ===> prompt. The ResourceView : Resource Definition panel is displayed. This panel displays the system image name and lists the resource classes that you can maintain.
2. Enter **S** in front of the STACK (TCP/IP Stack) class. The ResourceView : TCP/IP Stack List panel is displayed. The stacks already defined to the system image are listed on this panel.

   You can use this panel to:

   - Add new stack definitions
   - Browse, update, copy, and delete existing stack definitions

Task 2—Completing the General Description Panel

1. Press F4 (Add). The TCP/IP Stack General Description panel is displayed.
2. Complete the three TCP/IP Stack fields. These define the stack to your system.
3. For TCPaccess systems only, enter the Subsystem ID.
4. Set Monitoring to Active and provide a description of the stack.

Task 3—Setting Up the Monitoring Definition

1. Press F8 (Forward). The STACK *name* Monitoring Definition panel is displayed.
2. Set the frequency for monitor samples to be taken in the Monitor Interval field—this can be from 5 to 60 minutes.
3. Set the reporting level. This setting determines the amount of detail available for reports on this stack. For this monitor, reporting levels trend or summary are recommended.
4. Press F10 (EditLst). A list of monitoring attributes for the stack is displayed. For a description of the attributes, press F1 (Help).
5. Enter **A** to activate a monitoring attribute. The status changes to ACTIVE.

6. Enter **S** in front of the attribute. The type of attribute that you select determines the panel that is displayed:
   - Alert Details (Gauge)
   - Alert Details (Counter)
   - Alert Details (Enumerated)
7. Enter the values that you require for an alert to be raised and press F3 (Exit). The Monitoring Definition panel is displayed.
8. Repeat steps 5 to 7 for each attribute you want to define alert criteria to.

Task 4—Defining Stack Management

1. Press F8 (Forward). The Stack Management Definition panel is displayed. From this panel you can define the stack parameter dataset names.
2. Complete each of the Dataset Name fields (optional). Doing this enables you to browse the datasets from the IP resource monitor.

Note: This panel is unique to the stack resource class.

Task 5—Defining the Message Monitor, Automation Log, and Owner Details

1. Press F8 (Forward). The Status Monitor Message Details panel is displayed. From this panel you can define the actions to take when specific OS/390 console messages occur.
2. If you require this feature, complete the fields on the panel. For information on this feature see the *Automation Services Common User Guide*.
3. Press F8 (Forward). The Automation Log Details panel is displayed. This panel defines the resource transient log.
4. It is recommended that you accept the default settings for this feature. For more information, press F1 (Help).
5. Press F8 (Forward). The Owner Details panel is displayed. The fields on this panel are for documentation purposes only.
6. If required, complete the fields on the panel and press F3 (Save). The TCP/IP Stack List panel is displayed with the new definition added.

Defining Specific Resources

Some of the resources that you can define to your system have specific needs. These are set out below.

Defining Cisco Channel Cards

The minimum IOS levels supported by NetMaster for TCP/IP are 11.2 and 11.3. If you have a later IOS level, check with technical support for its supportability.

The SNMP GET Community name must be the same as the name used for the router. The SNMP SET Community name is not used in this release.

If the channel card that you are defining does not run TN3270 server, specify TN3270 logging active = **NO**.

Enabling Collection of Data from Cisco Routers

To collect SNMP data from Cisco routers, including channel cards, issue the following Cisco IOS command:

```
snmp-server community public RO
```

where RO indicates Read Only—you cannot update the routers via this access.

If an access list is being used to control SNMP, ensure that the IP address of the host running NetMaster for TCP/IP is in the access list.

See your Cisco documentation for further details.

Setting Up the OSA Support Facility

IBM's OSA Support Facility (OSA/SF) must be running to enable you to monitor and configure your OSAs. For information on setting up OSA/SF, see IBM's *OS/390 : OSA/SF User's Guide for OSA-2* (SC28-1855).

Defining 2216 Routers

2216 router support requires that your 2216 routers have Release 3.2 or later of IBM's Multiprotocol Access Services (MAS) installed.

Using Templates

If you have several similar resources in the same resource class, you can set up a template to create and populate the resource definitions.

To set up a template, do this:

1. Enter **/RADMIN.T.R** at the ===> prompt. The Resource Template Definition panel is displayed. This panel lists the resource classes that you can use templates with.
2. Enter **S** in front of the required class (for example STACK). The TCP/IP Stack List panel is displayed. From this panel you can:
   - Browse, update, copy, or delete an existing template
   - Apply an existing template
   - Create a new template
3. To create a new template, press F4 (Add).
4. Complete the fields on each of the definition panels. Press F1 (Help) for information on completing the panels.
5. After completing the definition panels the TCP/IP Stack List panel is displayed with the new template.

Associating Templates with Resource Definitions

To associate a template with a resource definition, do this:

1. Edit the resource definition.
2. Select the general description panel.
3. Enter the template name and apply the template to the resource definition.

To apply template changes to all associated resources, do this:

1. Enter **AP** in front of the new template. The Automation Services : Apply Template panel is displayed.
2. Define how you want the template to be applied and press F6 (Action). The ResourceView : System Image List panel is displayed.
3. Select the system image you want to apply the template to. The Automation Services : Messages List panel is displayed with details of the process.
4. Press F3 (File). The ResourceView : TCP/IP Stack List panel is displayed.

For more information on using templates see, the *Automation Services Common User Guide*.

Chapter

3

Setting Up the IP Node Monitor

This chapter describes how to set up the IP node monitoring facility available with NetMaster for TCP/IP.

This chapter contains the following topics:

- About the IP Node Monitor
- Setting Up an IP Node Monitor
- Maintaining Monitor Groups

About the IP Node Monitor

The IP node monitor facility allows you to monitor:

- Response times
- Network status
- SNMP attributes

When you define a host to be monitored you also specify a monitor group. The monitor group defines the checks performed.

For example, the availability and response time to critical IP nodes could be tested with a ping at frequent intervals. The possible responses are:

- OK—the test is successful
- TIMEOUT—there was no response
- ERROR—an ICMP error was received

You can have alerts raised if a node is unreachable. These alerts are automatically cleared by the IP node monitor when the status returns to OK.

You can also have an alert raised if the round trip time of a ping is too long. Such an alert is cleared when the time taken to reach the node falls below a specified threshold.

The results of the tests can be viewed in real time or sent to a file for later analysis.

For further information about using the IP node monitor, see Monitoring Alerts and Routers, in the *NetMaster for TCP/IP User's Guide*.

Setting Up an IP Node Monitor

To set up a node in the IP node monitor, do this:

1. Enter **/IPNODE** at the ===> prompt. The TCP/IP : IP Node Monitor panel is displayed.
2. Press F4 (Add). The TCP/IP : IP Node Monitor panel is displayed.

   This panel is used to define an IP node and assign it to a monitor group. You can enter an existing monitor group or add/update a specified group.
3. Enter the IP address or host name of the node to be monitored and the name of the monitor group. Enter a question mark (?) to obtain a list of existing monitor groups. Four sample monitor groups are provided. These are:
   - LOWLEVEL—checks the network status hourly.
   - STANDARD—checks the network status every 10 minutes and raises an alert if unreachable.
   - CISCOPERFINTENS—monitors the general availability of Cisco routers. It includes; 10 minute network status check and raises an alert if unreachable, SNMP monitoring of interface in/interface out discards, and CPU 5 minute average.
   - CISCOMONINTENS—extends CISCOPERFINTENS to include SNMP monitoring of interface in errors, interface out errors, packet rates, and router memory usage.

     Note: If you are adding the IP node to an existing monitor group, skip the following steps.
4. Press F5 (Group).
   - If the named monitor group exists, the TCP/IP : Monitor Group Details panel is displayed in update mode
   - If the named monitor group is new, the TCP/IP : Monitor Group Details panel is displayed with blank fields ready for the details to be added
5. Set the collection status to ACTIVE.
6. Set the Reporting Level to the value that you require—the setting in this field determines the reports available for the nodes in this group.

   Reporting levels TREND or SUMMARY are recommended for this group.
7. To add one or more attributes to be monitored in this group press F4 (AddAttr). The TCP/IP : Attributes List panel is displayed.

   Press F1 (Help) for a description of the attributes.

8. Select the attributes to be tested for this group, for example PING, and press Enter. The TCP/IP : Attribute Details panel is displayed.

   The following steps are determined by the attribute that was selected. The procedure for setting the PING attribute is shown.

   Note: All attributes, with the exception of PING, send SNMP GET requests to the node. Ensure that:

   – The node allows SNMP access from the stack that NetMaster for TCP/IP is associated with

   – The community name is defined in the NetMaster for TCP/IP administration facility.

9. Set the rate for the ping in the Rate field. This determines the period between samples.
10. Enter a value in the Create Alert if Timeout Status? field. To create an alert when the ping times out, specify **YES**; otherwise, specify **NO**.
11. Enter a value in the Create Alert if Error Status? field. To create an alert when an error is reported, specify **YES**; otherwise, specify **NO**.
12. (Optional) To define an action to be taken if an 'unreachable' alert is created, press F5 (UActions). The Alert Automated Actions panel is displayed.

    See Defining Actions to Be Initiated in the chapter, "Setting Up Proactive Monitoring" for details of the actions available.

13. To create alerts for round trips outside the defined limits enter values (in milliseconds) in the Alerts for Round Trip Time section of the panel.
14. (Optional) To define an action to be taken if an alert is created for an excessive round trip time, press F6 (RActions). The Alert Automated Actions panel is displayed. See Defining Actions to Be Initiated in the chapter, "Setting Up Proactive Monitoring" for details of how to define the automated actions.
15. (Optional for ping) Press F8 (Forward). The TCP/IP Ping Details panel is displayed. Set the way you want ping to work with this panel. You can set the:

    - Packet size—the size of the packet in bytes
    - Count—the number of pings sent
    - Wait—the period before a time out occurs

16. Press F3 (Save) to save the settings.

Maintaining Monitor Groups

Your existing monitor groups can be displayed, updated, copied, deleted, and have their members listed. You can also create new monitor groups.

Monitor group maintenance is done on the TCP/IP : Monitor Group List panel. To display the panel, enter **/IPMOND** at the ===> prompt..

Controlling Monitor Group Data Collection

The IPMON command is used to activate or deactivate data collection for a specified monitor group. You can use this command in OCS or in a batch file to have data collection active at specified times. The syntax of the command is shown below:

```
IPMON GROUP=name COLLECT= {NO | YES}
```

where *name* is the name of the monitor group.

Creating an Attribute for Monitoring

You can define additional SNMP attributes for monitoring (for example, to monitor the DLSW state in a busy router). To do this:

1. Copy the sample NCL procedure $IPSASNQ supplied in *?dsnpref*.IP620.IPTEXEC to your TESTEXEC library.
2. Rename the copy of $IPSASNQ NCL procedure to suit your installation standards.
3. Ensure that the NCL procedure runs at system initialization by invoking it from your INIT procedure.

   For example: `-EXEC MYDEFATR ACTION=DEFINE`

If you change an attribute definition, delete it from the existing monitor group definitions and then add it. This ensures that your changes are made effective.

Chapter 4

Setting Up Performance Monitoring

This chapter describes how to set up the performance monitoring facilities available with NetMaster for TCP/IP.

This chapter contains the following topics:

- Reporting Levels
- Enabling Performance Monitoring
- Defining a Report Center or Reporter Data Region

Reporting Levels

To enable NetMaster Reporter to collect data and send it to the Reporter SQL database, you must specify a performance monitor reporting level (other than *none*). The reporting levels control the amount of detail collected for the device being monitored. The four levels are:

- Reporting level *none*—No data is sent to the database; only online reports are available for the device being monitored.
- Reporting level *trend*—Hourly summary data is accumulated in IPFILE, from where it is rolled up and extracted to the IPTREND file.

  Note: If you do not want to accumulate this data in IPFILE and IPTREND, leave the name of the IPTREND file blank in the $IP IPFILES parameter group in ICS. Reporter can still collect data and send it to the SQL database.
- Reporting level *summary*—The same as reporting level *trend* plus hourly summaries of the data samples are written to the IPLOG and IPDETAIL files.
- Reporting level *detail*—The same as reporting level *summary* plus each data sample is written to IPLOG and IPDETAIL.

  Note: If you do not want to accumulate this data in IPLOG and IPDETAIL, leave the name of the IPDETAIL file blank in the $IP IPFILES parameter group in ICS. Reporter can still collect data and send it to the SQL database.

Restricting the Size of the IPLOG File

Some systems will produce several hundred thousand events each day. This causes the IPLOG file to become very large. To restrict the size of IPLOG, you can:

- Select summary or trend performance monitoring reporting levels
- Consider reducing the number of connection types that reporting applies to
- Keep detail information for a short period
- Review the number of attributes being monitored for each device

Enabling Performance Monitoring

To enable performance monitoring, you must:

- Ensure that the reporting files have been enabled. For more information, see the chapter, “Setting Up Connection Awareness”.
- Set up the appropriate monitoring.

Setting Up TCP/IP Logging and Monitoring

The TCP/IP logging and monitoring facility allows you to set up monitoring for:

- The local network interface workload
- Stack IP performance
- FTP workload
- Telnet workload
- Connection workload

To set up TCP/IP logging and monitoring:

1. Enter **/ICS** at the ===> prompt. The ICS : Customization Parameters panel is displayed.
2. Enter **U** in front of the $IP IPMONITOR parameter group. The TCP/IP Logging and Monitoring panel is displayed.
3. Check that the Receiver Status fields are set to Running for the events you want to sample. If these are not set to Running, check the settings specified in Task 2—Implementing System Event Receivers and Log Options in the chapter, "Setting Up Connection Awareness".
4. Enter **YES** into the TCP/IP workload sampling event fields that you want to sample.

You can control which applications have their connections monitored by setting up application definitions. For more information, see the chapter, "Setting Up Connection Awareness".

Performance Monitoring of IP Resources and Nodes

A NetMaster for TCP/IP region can monitor the performance of:

- The IP resources defined in the active system image. The Monitoring Activity field of the resource definition general description panel activates and inactivates monitoring. The monitoring definition panel defines the reporting level.
- The IP nodes visible on the IP node monitor. The monitoring status and reporting level are controlled by the monitor group definition.

Defining a Report Center or Reporter Data Region

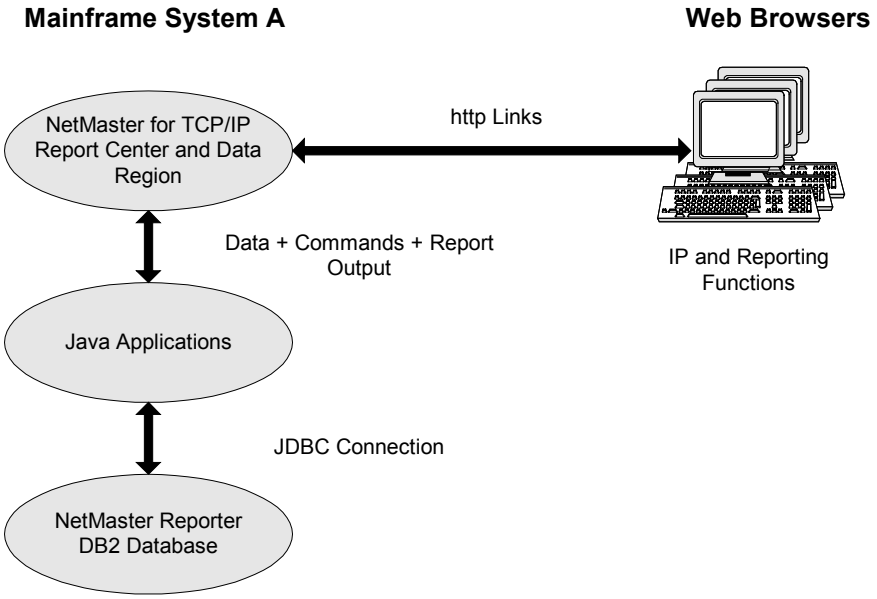
NetMaster Reporter provides a comprehensive range of reports that analyze your TCP/IP environment.

NetMaster Reporter controls all database updates and report generation from a central region known as the Report Center. The Report Center region must run on the same system as your database.

Reporter data regions collect performance data from your NetMaster for TCP/IP region and pass it to the Report Center for storage and processing. The Report Center region can also be a data region. There are two possible configurations.

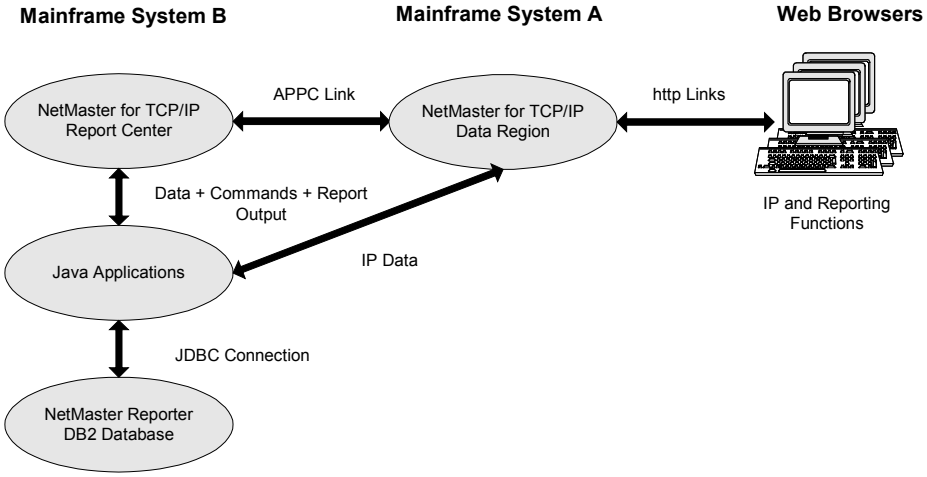
Configuration A

If your NetMaster for TCP/IP region runs on the same system as your database, it should function as a Reporter data region and also as the Report Center. You can use the Reporting option from this region's web interface.



Configuration B

If your NetMaster for TCP/IP region does not run on the same system as your database, it can function as a Reporter data region. As a data region, your NetMaster for TCP/IP region will send data to the database over an IP socket connection. You can use the Reporting option from this region's web interface.



See the 'Concepts' chapter of *Working with NetMaster Reporter* for further information.

See the 'Implementation' chapter of *Working with NetMaster Reporter* for detailed information about defining a Report Center or a Reporter data region.

Defining a Report Center

To define your NetMaster for TCP/IP region as a Report Center, use the $WR REPORTER parameter group in the NetMaster for TCP/IP region to implement the connectivity with the NetMaster Java Framework and the SQL database, and the HFS directories for various product files.

Defining a Reporter Data Region

To define your NetMaster for TCP/IP region as a data region, use the $WR REPORTER parameter group in the NetMaster for TCP/IP region to specify the APPC link name of the Report Center region.

Chapter

5

Setting Up Access Control

This chapter describes the tasks to control access to TCP/IP.

This chapter contains the following topics:

- About Access Control
- Controlling Access to TCP/IP
- Controlling User Access
- Controlling Access to Administration Functions

About Access Control

Access control allows you to:

- Control access to your applications
- Set permissions for file transfers
- Set permissions for Telnet access
- Validate users with a user sign on panel
- Set a user's default applications
- Define which users will have access to the functions on the administration menu

How Access Control Works

TCP/IP requests are intercepted by the system and passed to the data space. The system then checks the access privileges for the TCP/IP stacks, ports, and hosts defined to it.

To enable access control you must perform the following tasks:

- Apply SAF security to the administration menu
- Define the TCP/IP stacks, ports, and hosts to your system
- Set up the TCP/IP access control parameters

Controlling Access to TCP/IP

You can set up NetMaster for TCP/IP to control access to and from your TCP/IP stacks. You can control access to:

- Ports—do this to control access to your applications
- Local and remote hosts—do this to control file transfers and set telnet access

The tasks to complete to set up access control are described below.

Task 1—Enabling Access Control

To enable the access control functionality, do this:

1. Access the *dsnpref.rname*.TESTEXEC(DSPSYSIN) member and set SEC=YES.
2. Restart the data space for the parameter to take effect.
3. Set the PROD=TCPIPACNTL parameter in your RUNSYSIN member.

Task 2—Defining TCP/IP Stacks

To define a TCP/IP stack to Access Control, do this:

1. Enter **/IPADMIN.A.S** at the ===> prompt. The Access Control : Define Stacks panel is displayed. This panel lists the TCP/IP stacks you can control access to.
2. Press F4 (Add). The Access Control : Add/Update Stack panel is displayed.
3. Enter the stack details and parameters in the appropriate fields. For a description of the fields press F1 (Help).
4. Press F3 (File). The Access Control : Define Stacks panel is displayed with the new stack definition added to the list.
5. Repeat steps 2 to 4 for each stack you want to control access to.

Task 3—Defining Ports

Most applications are associated with a specific TCP/IP port. You can control the access an application has to the stack by controlling the port.

To define a port to Access Control, do this:

1. From the Access Control : Primary Menu select **P** - Define Ports. The Access Control : Define Ports panel is displayed. This panel lists the ports that Access Control is applied to.
2. Press F4 (Add). The Access Control : Port Definition panel is displayed.
3. Enter the port details, security restrictions, and parameters in the appropriate fields. For a description of the fields press F1 (Help).
4. Press F3 (File). The Access : Control Manage Ports panel is displayed with the new port details added to the list.
5. Repeat steps 2 to 4 for each port that you want to apply access control to.

Task 4—Defining Hosts

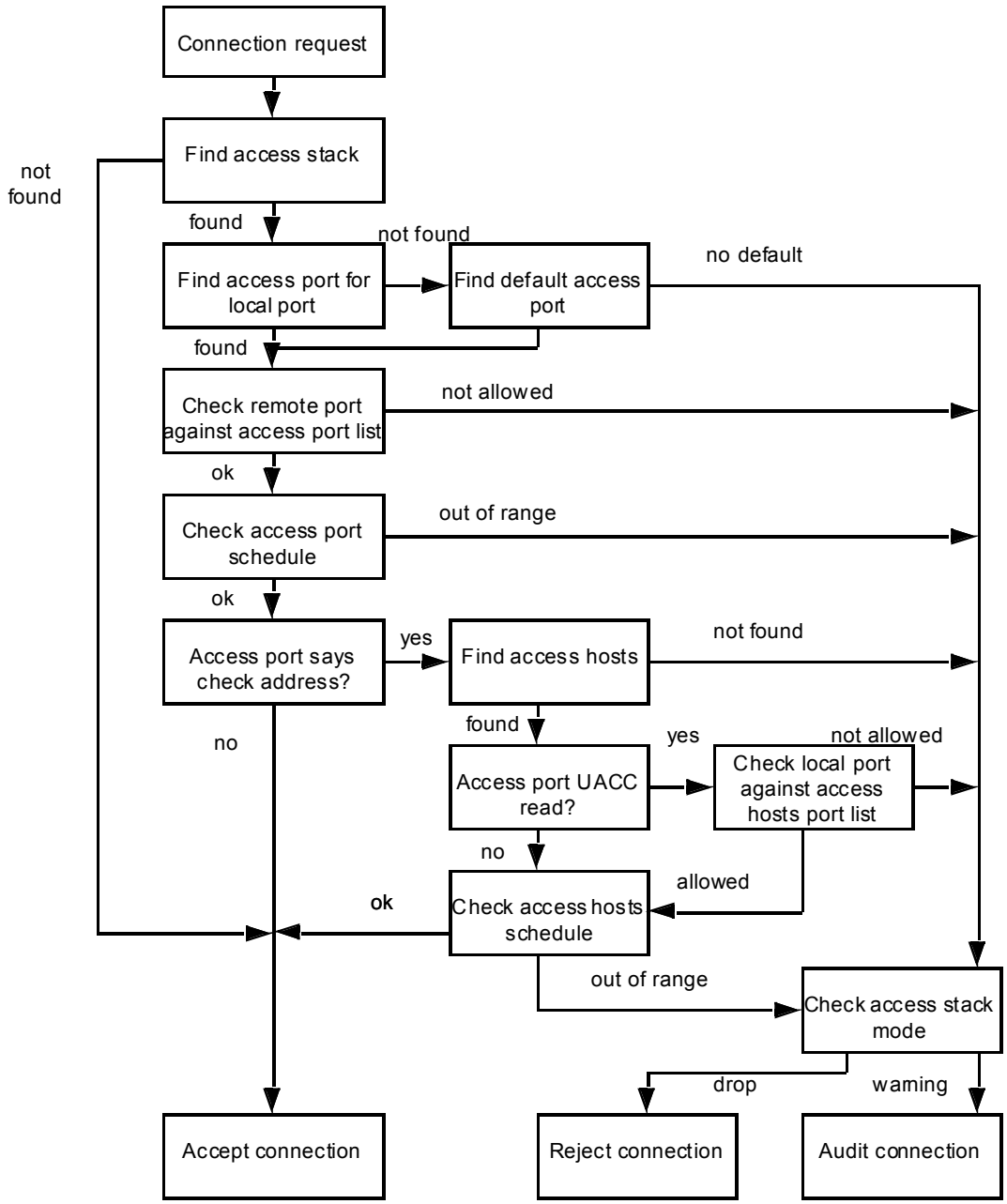
Specific remote hosts can be identified by their IP address range. You can control inbound and outbound connections between your stack and remote hosts by controlling access to the range of IP addresses associated with the remote host.

To define a remote host to Access Control, do this:

1. From the Access Control : Primary Menu select **H** - Define Hosts. The Access Control : Define Hosts panel is displayed. This panel lists the hosts that Access Control is applied to.
2. Press F4 (Add). The Access Control : Host Definition panel is displayed.
3. Enter the host details, security restrictions, and parameters in the appropriate fields. For a description of the fields, press F1 (Help).
4. Press F3 (File) to save the host and access details. The Access : Control Manage Hosts panel is displayed with the new host details added to the list.
5. Repeat steps 2 to 4 for each host that you want to apply access control to.

Connection Access Checking

When a connection request arrives, the access checking shown in the following diagram is applied.



Controlling User Access

NetMaster for TCP/IP uses two procedures to control user access and functionality. Both of these procedures are defined on the TCP/IP : Access Control Parameters panel.

The network solicitor procedure allows you to:

- Validate users with a user signon panel
- Set a user's default applications with an application selection list

The application definition procedure allows you to set application access privileges by setting port and remote host access privileges.

To set up TN3270 access control, do this:

1. Enter **/ICS** at the ===> prompt. The ICS : Customization Parameters panel is displayed.
2. Enter **U** in front of the $AX TN3270ACCESS parameter group. The ICS : Initialization Parameters panel is displayed.
3. Complete the fields on the panel. For a description of the fields press F1 (Help).
4. Press F3 (File) to save the settings.

Displaying a User Signon Panel and Application Selection List

The specified network solicitor procedure determines:

- The user signon panel
- The contents of the user's application selection list

A sample network solicitor procedure ($AXCNTL) is provided in the *dsnpref*.MS500.MSTEXEC dataset. Modify this procedure to your requirements.

To modify the signon panel and application selection list, do this:

1. Copy the supplied $AXCNTL member into your TESTEXEC dataset.
2. Make the changes that you require to the $AXCNTL procedure in your TESTEXEC dataset.

Defining Access to Applications

The specified application definition procedure defines what applications are accessible to your users. A sample application procedure ($AXAPPL) is provided in the *dsnpref*.MS500.MSTEXEC dataset. Modify this procedure for your systems application names and signon user data requirements.

To define access to the applications on your system, do this:

1. Copy the supplied $AXAPPL member into your TESTEXEC dataset.
2. Make the changes that you require to the $AXAPPL procedure in your TESTEXEC dataset.

Using EASINET with Access Control

If you have SOLVE:Access you can use your existing EASINET to implement user ID based IP connection security.

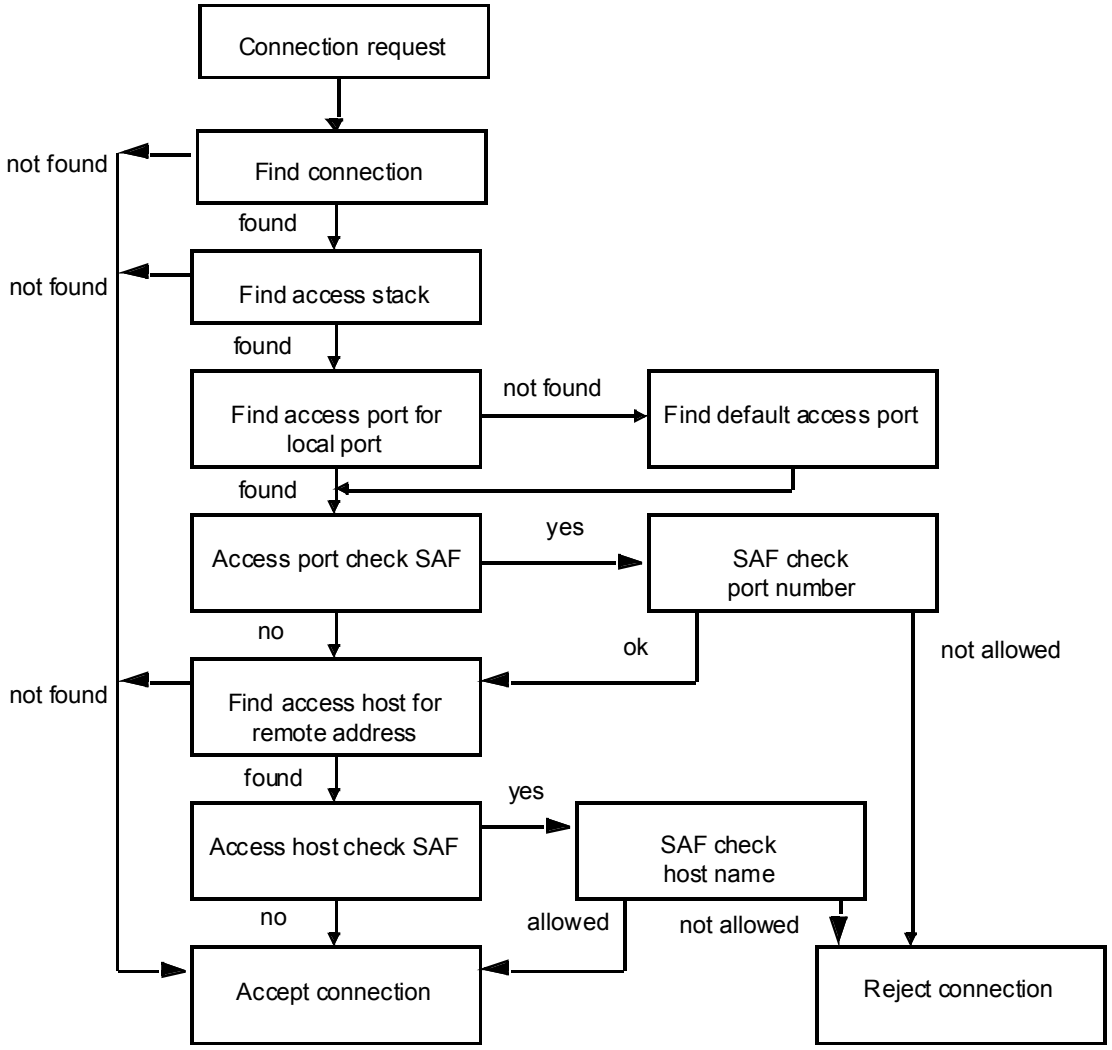
To use EASINET to implement user ID based IP connection security, do this:

1. Ensure that the system running EASINET is the VTAM default application for the terminals that connect via TN3270.
2. Use the SELFTEST DSP command to check that the system is connected to the data space.

   This system and the data space must be connected for the access control definitions to be available.
3. Add the SYSPARMS IPCHECK=VERIFY command to the INIT procedure.

How SAF User ID Checking Works

Connections such as Telnet access to EASINET and FTP access to TCPaccess can be checked by SAF. When a connection request that can be checked by SAF arrives, the SAF checking process shown in the following diagram is applied.



Defining a Port to SAF

To setup SAF security to a port, do this:

1. Use your security product to define a FACILITY class rule named PORT.*portnum*.

   For example, in a RACF system the command would be:

   ```
   RDEF FACILITY PORT.portnum UACC(NONE)
   ```

2. Use your security product to add permits to the users of the port.

   The security levels available are:

   - Read—the port can be used
   - None—the port cannot be used

   For example, in a RACF system the command for setting the security level to read for user userid01 would be:

   ```
   PE PORT.portnum CLASS(FACILITY) USER(userid01) ACCESS(READ)
   ```

3. Refresh your security rules.

   For example, in a RACF system the command would be:

   ```
   SETROPS RACLIST(FACILITY) REFRESH
   ```

Defining a Host to SAF

To setup SAF security to a host, do this:

1. Use your security product to define a FACILITY class rule named HOST.*hostname*.

   For example, in a RACF system the command would be:

   ```
   RDEF FACILITY HOST.hostname UACC(NONE)
   ```

2. Use your security product to add permits to the users of the host.

   The security levels available are:

   - Read—the host can establish a connection
   - None—the host cannot establish a connection

   For example, in a RACF system the command for setting the security level to read for user userid01 would be:

   ```
   PE HOST.hostname CLASS(FACILITY) USER(userid01) ACCESS(READ)
   ```

3. Refresh your security rules.

   For example, in a RACF system the command would be:

   ```
   SETROPS RACLIST(FACILITY) REFRESH
   ```

Controlling Access to Administration Functions

Access to the Access Control : Primary Menu and $DICMD functions is not permitted until you apply SAF security.

To setup SAF security to the Access Control : Primary menu and $DICMD functions, do this:

1. Use your security product to define a FACILITY class rule named $SOLVE.ACCESS.CONTROL.

   For example, in a RACF system the command would be:

   ```
   RDEF FACILITY $SOLVE.ACCESS.CONTROL UACC(NONE)
   ```

2. Use your security product to add permits to the users of the access control definition functions.

   The security levels available are:

   - Read—the access control definitions can be browsed but not changed
   - Update—existing access control definitions can be updated
   - Alter—access control definitions can be updated, added, and deleted

   For example, in a RACF system the command for setting the security level to update for user userid01 would be:

   ```
   PE $SOLVE.ACCESS.CONTROL CLASS(FACILITY) ID(userid01) ACCESS(UPDATE)
   ```

3. Refresh your security rules.

   For example, in a RACF system the command would be:

   ```
   SETROPTS RACLIST(FACILITY) REFRESH
   ```

Chapter

6

Advanced Configuration Tasks

This chapter describes the tasks to configure the advanced features of NetMaster for TCP/IP.

This chapter contains the following topics:

- Enabling Packet Tracing
- Enabling Multisystem Support
- Enabling the Management of SNA/VTAM Resources and Sessions
- Specifying Telnet Translation Tables
- Setting up the Web Interface
- Working with Data Space Definitions
- About SNMP
- About NetMaster Socket Management for CICS
- Maintaining Connection List Criteria

Enabling Packet Tracing

The tasks that you perform in this section are determined by the TCP/IP system that you have:

- For Communications Server systems go to the section "Enabling Communications Server Systems".
- For TCPaccess systems go to the section "Enabling Communications Server Systems".

Enabling Communications Server Systems

To use the packet tracing facility in Communications Server systems you need to set up an external writer and RACF security. For information on RACF security requirements see the chapter "Setting up Connection Awareness".

Creating an External Writer

The OS/390 Component Trace facility (CTRACE) collects trace data from Communications Server.

For packet tracing to work you need to create source JCL that invokes a CTRACE external writer. The external writer is used to write trace data to a dataset each time packet tracing is used.

The external writer must use single datasets only for recording trace data. This is because NetMaster for TCP/IP only reads one trace dataset.

To ensure that CTRACE is available for use, do this:

1. Tailor this example JCL to suit your specific requirements. The JCL procedure should be added to your SYS1.PROCLIB system library.

   ```
   //PTTCP PROC
   //*
   //* CTRACE External writer for TCP/IP Packet Tracing
   //*
   //IEPROC EXEC PGM=ITTTRCWR,TIME=1440
   //*
   //*TRCOUT01 DD DSN=TCPIP.PTRACE.PTRACE,DISP=OLD
   //
   //TRCOUT01 DD DSN=TCPIP.PTTRACE.PTRACE,DISP=(NEW,CATLG),
   //            VOL=SER=????,UNIT=SYSDA,DSORG=PS,
   //            SPACE=(4096,(100,10))
   ```

   PTTCP is the name of the external writer. You can change this to suit your installation; it must be between one and seven characters long. This name is also used in Step 2.

Do one of the following:

- Preallocate DS with DSORG=PS, RECFM=27994, and LRECL=VB
- Leave DSORG, RECFM, and LRECL unspecified and allow the program to determine their values

After the trace dataset is allocated do one of the following:

- Modify the JCL so that TRCOUT01 is specified with DISP=OLD
- Delete or rename the TRCOUT01 dataset before rerunning the procedure

2. Determine the command that you are going to use to start the CTRACE external writer. You must specify this command on the CTRACE panel when you start CTRACE for the first time from NetMaster for TCP/IP. For example:

```
TRACE CT,WTRSTART=PTTCP,NOWRAP
```

where:

- TRACE is the MVS TRACE command.
- CT indicates that it is a component trace.
- PTTCP is the name of the external writer. This can be changed to suit your installation; it must be between one and seven characters long.

For more information, see the *OS/390 MVS Diagnosis: Tools and Service Aids* manual.

3. Skip to the section “Starting a Trace”.

Enabling TCPaccess Systems

To use the packet tracing facility in TCPaccess systems you need to set up an external writer and RACF security.

For packet tracing to work you need to create source JCL that invokes an external writer. The external writer is used to write trace data to a dataset each time packet tracing is used.

The external writer is used by the Component Trace facility (CTRACE). CTRACE collects trace data from TCPaccess.

The external writer must use a single dataset only for recording trace data. This is because NetMaster for TCP/IP only reads one trace dataset.

To ensure that CTRACE is available for use, complete the following steps:

1. Tailor this example JCL to suit your specific requirements. The JCL procedure should be added to your SYS1.PROCLIB system library.

   ```
   //PTTCP PROC
   //*
   //* CTRACE External writer for TCP/IP Packet Tracing
   //*
   //IEPROC EXEC PGM=ITTTRCWR,TIME=1440
   //*
   //TRCOUT01 DD DSN=TCPIP.PTRACE.PTRACE,DISP=OLD
   //
   ```

   where:

   PTTCP is the name of the external writer. You can change this to suit your installation. This name is also used in step 2.

   Do one of the following:

   - Create a trace dataset with the attributes DSORG=PS, RECFM=VB, and LRECL=27994
   - Leave DSORG, RECFM, and LRECL unspecified and allow the program to determine their values

2. Set up the TCPaccess trace component.

   The sample JCL member is RUNTRACE. This is found in the CNTL library used to originally install TCPaccess. Tailor this to suit your site specific requirements. The JCL procedure is added to your SYS1.PROCLIB system library.

   Ensure that the subsystem name in the RUNTRACE procedure matches the trace name parameter in the IJTCFG*xx* member used by TCPaccess.

Starting a Trace

To start a trace, do this:

1. Enter **/IPPKT.S** at the ===> prompt. The TCP/IP : Start CTRACE panel appears. The details of the panel will depend on the TCP/IP software that you have.

2. For TCPaccess, enter the trace job name created in step 2 (above) in the TCPaccess Trace Job field.

3. Enter the command that you are going to use to start the CTRACE external writer in the Command to Start TCPaccess CTRACE field. For example:

   ```
   TRACE CT,WTRSTART=PTTCP,NOWRAP
   ```

   where:

   TRACE is the MVS TRACE command.

   CT indicates that it is a component trace.

   PTTCP is the name of the external writer created in step 1 of the section "Enabling TCPaccess Systems". You can change the name to suit your installation.

4. Press F6 (Action) to start the trace.

Note: You must specify this command on the Start panel when you start CTRACE for the first time from NetMaster for TCP/IP.

Enabling Multisystem Support

If you have NetMaster for TCP/IP regions on different OS/390 images, you can link them together to form a multisystem configuration.

A multisystem configuration enables you to log onto your local NetMaster for TCP/IP region and view and control the resources of linked NetMaster for TCP/IP regions. You can do things such as:

- View a single IP resource monitor display of the resources in all the linked regions
- Display the alerts raised from all the linked regions
- Display a consolidated list of IP connections across multiple stacks and regions

Multisystem configurations are set up and administered from the Automation Services : Multi-System Support Menu. To access this menu enter **A.M** at the ===> prompt of the Primary Menu.

For more information press F1 (Help).

Important! *See the chapter "Administering a Multisystem Environment" in the* Automation Services Administrator Guide *before setting up a multisystem environment.*

Enabling the Management of SNA/VTAM Resources and Sessions

NetMaster provides two features to help you to manage your SNA/VTAM resources and sessions. These are:

- The Network Control System (NCS)
- The Network Tracking System (NTS)

About the Network Control System (NCS)

Network Control System (NCS), for managing SNA resources, is included in your NetMaster for TCP/IP license. You can use NCS from the Telnet connection displays or lists. NCS enables you to display and control the SNA/VTAM resources that are involved in the connections.

Full function access to the NCS menu is only available if you have a NetMaster for SNA license.

For further information, see the *NetMaster for SNA User's Guide* and the *NetMaster for TCP/IP User's Guide*.

There is no specific action required to enable NCS.

About the Network Tracking System (NTS)

You may be set up to use the Network Tracking System (NTS) feature of NetMaster to manage your SNA sessions. If so, you can also use it from the Telnet connection displays or lists.

NTS enables you to display and trace the SNA/VTAM sessions that are involved in the connections. You can also use NTS to provide SNA transaction time details for the NetMaster for TCP/IP Telnet Transaction Path Analyzer function.

For further information, see the *NetMaster for SNA User's Guide* and the *NetMaster for TCP/IP User's Guide*.

If you are using SOLVE:Access, the NTS data displays can show associated MAI sessions. To do this, activate MAI session visibility, as described in the chapter "Tailoring NTS" of the *NetMaster for SNA Administrator Guide*.

Specifying Telnet Translation Tables

Telnet connections normally operate using the ASCII character set. Messages are translated between ASCII and EBCDIC when using Telnet connections from NetMaster for TCP/IP.

Note: There is no need to change the defaults unless you have problems with the way data is displayed on Telnet connections. For example, you might be using a national language character set.

Data translation between ASCII and EBCDIC is determined by the:

- Translate Table Dataset—a partitioned dataset containing the translation tables you want to use.
- Translate Table—a default value for the table name being used for Telnet connections. The table name is the name of the member in the partitioned dataset. If you specify a dataset without a table name, then your Telnet connections will use a default of TELNET.

To add your own translation tables add a member to the dataset. Each member in this dataset contains two tables: the first translates from ASCII to EBCDIC, and the second from EBCDIC to ASCII. The following table formats are supported:

- Source form—used by Communications Server
- Binary form—used by Communications Server and TCPaccess Open Edition Support

Note: If you are using Communications Server, see Using Translation Tables in the IBM *Communications Server Customizing and Administration Guide* for more information.

If you are using TCPaccess, you can generate the binary tables by using the TSO CONVXL8 or TSO LOADXL8 commands supplied with TCPaccess Open Edition Support.

For information on using Telnet connections, see the *NetMaster for TCP/IP User's Guide*.

To specify the Telnet translation tables:

1. Enter **/ICS** at the ===> prompt. The ICS : Customization panel is displayed.
2. Select the $NM TELNETTRT parameter group. The Telnet Translate Parameter is displayed.
3. Specify a translate table DSN and translate table.
4. Press F3 (File) to save your settings.

Setting up the Web Interface

The web interface can be used with Internet Explorer and Netscape.

Setting Up Internet Explorer

The NetMaster for TCP/IP web interface will run on Internet Explorer version:

- 4.01 or higher
- 5.0 or higher with Java VM component installed

For your installation to run correctly, ensure the following internet options are enabled:

- Security
  - Java
  - High safety
  - Allow cookies to be stored on your computer
  - Scripting of Java applets
- Advanced
  - JIT
  - Show images

Setting Up Netscape

The NetMaster for TCP/IP web interface will run on:

- Netscape Navigator Version 4.06 or higher
- Netscape Communicator Version 4.5 or higher

Ensure that the following advanced settings are enabled:

- Java
- JavaScript
- Style sheets
- Automatically load images
- Cookies
- Smart update

Troubleshooting

If applets fail to appear in some pages the browser may have reached the maximum heap size. To rectify this problem, do one of these:

- Minimize and restore the browser and then click the OK/Redo button on the current page.
- Restart the browser.

Setting Security Prompts for Web Applications

You may be prompted with a series of security warning dialog boxes:

- The first time you log onto NetMaster for TCP/IP
- When NetMaster for TCP/IP is updated
- The first time you use the alert monitor
- The first time you use the IP node monitor

The dialog boxes displayed are determined by your browser security settings.

Whenever you are prompted to install a software component from Computer Associates, Inc. click on the Yes button—this ensures that all software components are installed.

Note: Some software installs may take several minutes to complete.

Disabling Internet Explorer Security Prompts

You will be prompted with security warning dialog boxes only if you have a medium or high security setting.

You can stop Internet Explorer prompting with the Security Warning dialog boxes by selecting the Always trust content from Computer Associates, Inc. checkbox on the first Security Warning dialog box.

Disabling Netscape Security Prompts

To stop Netscape prompting with the Java Security dialog boxes, click the Remember this decision checkbox before you click the Grant button.

Revoking Netscape Security Settings

To revoke the security setting, do this:

1. Click on the Padlock icon at the bottom left hand corner of the Netscape window.
2. Set Netscape security to the required settings.
3. Click on the OK button.

Synchronizing Time Zones

By default the web interface uses mainframe local time. Web information is only displayed in the user's PC local time if the PC is in the same time zone as the mainframe.

If your mainframe is set to GMT with a time zone offset you can display web information in PC local time. To do this add the following line to the READY member of the TESTEXEC dataset:

```
&&000$DFGONZO = YES
```

Working with Data Space Definitions

Each LPAR has only one data space. This means multiple regions on the same LPAR use the same stack, port, host, stack software, and application data space definitions.

You can use the $DICMD OML procedure to:

- List data space definitions
- Export data space definitions to other LPARS
- Add and delete definitions to your data space without using the definition panels
- Reclaim data space capacity

The $DICMD OML procedure must have the required authority to add and delete definitions. The authority level required for the $DICMD OML procedure is the same as the Access Control : Primary Menu functionality. For information on setting authority for $DICMD see Controlling Access to Administration Functions in the chapter, "Setting Up Access Control".

Listing Data Space Definitions

You can list data space definitions with the $DICMD procedure LIST action. For example, to list the stack definitions, do this:

From the Command Entry panel enter:

```
$DICMD ACTION=LIST CLASS=STACK
```

Exporting Data Space Definitions

You can export your data space definitions to an NCL procedure. The procedure is then used to import the definitions to data spaces on other LPARS.

To export a data space definition, do this:

1. From the Command Entry panel enter the command:

   $DICMD ACTION=LIST CLASS=STACK FORMAT=CMD DSN=*MY.PDS*(*STACKADD*) REPLACE=*N*

   An NCL procedure is generated that contains the details of the stacks on this data space.
2. Copy the NCL procedure to a TESTEXEC library.
3. Repeat steps 1 and 2 for CLASS=PORT, HOST, SSW and APPL.

   PORTADD, *HOSTADD*, *SSWADD* and *APPLADD* NCL procedures are generated and copied.
4. From the Command Entry panel issue the command:

   $CMDENT *STACKADD*

   The stack definitions are imported to the data space.
5. Repeat step 4 for the *PORTADD*, *HOSTADD*, *SSWADD* and *APPLADD* NCL procedures.

For more information on the $DICMD procedure see the online help—from the OCS panel type **$DICMD ?**

About SNMP

NetMaster for TCP/IP uses SNMP to collect the following data from routers and other network devices:

- Cisco channel card information
- Interface information
- General system information
- Routing tables
- IP node monitor attributes

NetMaster for TCP/IP uses SNMP GET requests to query information.

See the *NetMaster for TCP/IP User's Guide* for information on how to display the information obtained using SNMP.

About SNMP Security

SNMP uses the following to control user access to data from devices:

- Community names
- Access lists

About Community Names

Community names can be used by SNMP to determine the level of access you have to a particular SNMP device. For example, different community names might be required depending on whether you want to browse or have write access to MIB objects.

Note: NetMaster for TCP/IP needs only read access.

A default value of *public* in lower case is used as the community name for NetMaster for TCP/IP. If you have any devices that use a different community name use the community name access facility (see the section "Predefining SNMP Community Names").

About Access Lists

An access list is used to specify the IP addresses from which a device responds to SNMP requests. The IP address of the host making SNMP requests (the host running NetMaster for TCP/IP) should be defined in the devices' SNMP access list; otherwise SNMP requests will fail.

Predefining SNMP Community Names

You can set up predefined associations between IP addresses and SNMP community names. This means that you can perform TCP/IP monitoring functions without having to enter the community name each time.

Note: You must have TCP/IP update authority to define a community name.

To define an SNMP community name, do this:

1. Enter **/IPADMIN.SC** at the ===> prompt. The TCP/IP : SNMP Community List panel is displayed. This panel displays a list of the currently defined community names.
2. Press F4 (Add) to create a new community name. The TCP/IP SNMP Community Definition panel is displayed.

```
PROD---------------- TCP/IP : SNMP Community Definition ------------------------
Command ===>                                                Function=ADD

Identify the SNMP Community
  IP Address/Mask............
  Description ...............

Community Names

  SNMP GET Community Name ...
  SNMP SET Community Name ...









 F1=Help      F2=Split     F3=File      F4=Save
                           F9=Swap                                F12=Cancel
```

3. Enter the details of the community definition that you require.
4. Press F3 (File) to save the definition. The new definition appears on the TCP/IP : SNMP Community List panel.

Using IP Address Ranges and Masks

The IP address of a community name must always be made up of four elements separated by periods (.). You can enter a specific value, a range, or a mask for each element. The dash (-) is used to specify a range and the asterisk (*) is used as a mask. These can be used in combinations. See the following examples:

198.0.80.*

204.28.10.15-30

192.10.16-24.*

Important! *You must not use ranges that overlap when you define community names. The example IP address ranges following will cause an error condition:*

204.28.10.15-30

204.28.10.20-40

About NetMaster Socket Management for CICS

NetMaster for TCP/IP provides an interface to NetMaster Socket Management for CICS. When this interface is enabled, it does the following:

- Passes additional CICS information about TCP connections to NetMaster for TCP/IP, such as user ID, CICS transaction name, and CICS transaction number.

  This information then becomes available through central network management displays within NetMaster for TCP/IP.
- Allows you to monitor CICS IP resources (resource class CICMON).

See the *Unicenter NetMaster Socket Management for CICS Getting Started* guide for more information.

Configuring NetMaster Socket Management for CICS

To configure NetMaster Socket Management for CICS in your region, set the PROD=SOCKETMGMT parameter in your RUNSYSIN member.

Customizing NetMaster Socket Management for CICS

If NetMaster Socket Management for CICS is configured in your region, it is automatically enabled when you implement your region. You can then customize it if required.

To customize NetMaster Socket Management for CICS:

1. Enter **/ICS** at the ===> prompt. The ICS : Customization panel is displayed.
2. Enter **U** beside the $SK SOCKETMGMT parameter group. The SOCKETMGMT - SocketMgmt Agents parameter is displayed.
3. Specify (YES or NO) whether resources are to be dynamically added to the System Image.
4. Specify a background user ID and password, known to your CICS region. This enables NetMaster for TCP/IP to perform background health checks.
5. Specify (YES or NO) whether NetMaster signon details are to be used for user access to CICS.
6. Press F3 (File) to save your settings.

Issuing CICS Commands in a Command Entry Environment

To issue a CICS command in a Command Entry environment:

1. Enter **CMD** (Command Entry - SocketMgmt) next to a CICMON resource on the IP resource monitor (/IPMON). A Command Entry panel is displayed.
2. Type a Socket Management command in the SocketMgmt Command field.
3. Press F4 (Execute). The command output is displayed on the Command Entry panel.

```
PROD--------------------CICS SocketMgmt : Command Entry -----------------------
Command ===>                                                   Scroll ===> CSR

SocketMgmt Command CPTSTATUS

            Server QATS13D3

LINE   <---+----10---+----20---+----30---+----40---+----50---+----60---+----70>
000001 ==> cptstatus
000002   * COMMAND ACCEPTED
000003     Output for command CPTSTATUS:
000004     TCP/IP Jobname ................. WTMCC600
000005            SSID ................... WTM1
000006            Trace SSID ............. WTMV
000007     CICS Jobname ................... QATS13D3
000008          VTAM Appl .................
000009     CPT Product .................... Unicenter TCPaccess CICS Programmer
000010         Version ..................... 6.0.0
000011         T09CONxx ................... T09CONFG
000012     SocketMgmt Product ............. Unicenter Network Socket Manager fo
000013                Version ............. 1.0.0
000014     Running Products ............... +CPT+SOCKETMGMT
000015     ACDC Control block address ..... 00042004
000016     ACDC Security Exit .............
000017     Statistic Type    Bytes      Calls
000018     Received:          5186        114
000019     Send:             11697         90
000020     *END*
****** *************************** BOTTOM OF DATA ***************************


F1=Help      F2=Split     F3=Exit      F4=Execute   F5=Find      F6=Refresh
F7=Backward  F8=Forward   F9=Swap     F10=Left     F11=Right    F12=Recall
```

For details of the information displayed and actions available, press F1 (Help).

Maintaining Connection List Criteria

You can create and store connection list criteria. See "Managing Connection Lists" in the *Unicenter NetMaster Network Management for TCP/IP User Guide*.

To maintain these connection list criteria:

1. Enter **/IPACRIT** at the ===> prompt. The TCP/IP : Connection List Criteria Definitions panel is displayed.

```
PROD---------- TCP/IP : Connection List Criteria Definitions ---------/IPACRIT
Select Option ===>

   T   - List Telnet Connections Criteria
   CS  - List CICS Socket Connections Criteria
   C   - List Connections Criteria
   X   - Exit
```

2. Enter the mnemonic for the type of criteria that you want to maintain (for example, enter **T** for Telnet connections criteria). A list of criteria is displayed.

```
PROD------------- TCP/IP : Telnet Connection List Criteria --------------------
Command ===>                                                Scroll ===> CSR

                                                            S/=View D=Delete
    Search Name  Description
    TELPROD1     Telnet PROD1 connections
    TELPROD2     Telnet PROD2 connections
    **END**
```

3. To view a criteria definition, enter **S** next to its name. The TCP/IP : Connection List Search Criteria Definition panel is displayed.

```
PROD-------- TCP/IP : Connection List Search Criteria Definition --------------
Command ===>                                                Scroll ===> CSR

    Type ................... List Telnet Connections
    Name ................... TELDENM1
    Description ............ Telnet DENM1 connections
    Last Updated ........... 27-FEB-2002 23.03 PCSIV1

 Criteria
    Fast Search? ........... NO
    History? ............... NO
    Link/Channel Card ...... DENM1
****************************** BOTTOM OF DATA ******************************
```

4. To delete a criteria definition, enter **D** next to its name on the connection list criteria panel. When you confirm the deletion, the definition is shown as deleted.

Chapter 7

Setting Up Proactive Monitoring

This chapter describes how to implement and administer proactive monitoring.

This chapter contains the following topics:

- About Proactive Monitoring
- Event Detectors
- Defining an Event Detector
- Monitoring Connections
- Monitoring Custom Events
- Monitoring FTP Failures
- Monitoring Listeners
- Monitoring Messages—Cisco Channel Card TN3270 Log
- Monitoring Messages—ICMP
- Monitoring Messages—OS/390 Console
- Implementing a Trouble Ticket Interface
- Implementing the Alert History Function
- Applying Alert Monitor Filtering
- Forwarding Alerts

About Proactive Monitoring

Proactive monitoring has two parts:

- Event detectors which detect potential problems
- The alert monitor that tells you about these potential problems

The alert monitor tells you about the following problems or potential problems in your TCP/IP system:

- Connections that violate your defined criteria; for example, large FTPs
- FTP failures
- Critical TCP or UDP listeners not being active
- Cisco Channel Card TN3270 server log messages
- ICMP messages
- OS/390 console messages

This chapter describes how to set up proactive monitoring for your system. For details of using the alert monitor, see the *NetMaster for TCP/IP User's Guide*.

Event Detectors

There are seven types of event detectors available:

Use this type ...	To monitor ...	As described in the section...
CCTN3270	Cisco channel TN3270 log messages	"Monitoring Messages—Cisco Channel Card TN3270 Log"
CONNECT	Connections	"Monitoring Connections"
CONSOLE	OS/390 console messages	"Monitoring Messages—OS/390 Console"
CUSTOM	Custom Event	"Monitoring Custom Events"
FTPFAIL	FTP failures	"Monitoring FTP Failures"
ICMP	ICMP messages	"Monitoring Messages—ICMP"
LISTENER	Listening ports	"Monitoring Listeners"

You can use the TCP/IP : Event Detector Controls List panel to:

- Update detector definitions
- Add further detectors to the list
- Delete obsolete detectors from the list

Sample Event Detectors

A set of sample detector definitions is supplied with NetMaster for TCP/IP. To access the Event Detector Control List from the Primary Menu enter **A.IP.E**. Each type of event is represented in the samples. The CUSTOM event has two samples—one shows how to create an alert and the other how to clear it. Use these definitions as examples when you create your own event detectors.

```
PROD------------- TCP/IP : Event Detector Controls List -------------------
Command ===>                                                Scroll ===> PAGE

                                        S/U=Update C=Copy RC=RemoteCopy D=Delete
Detector Type  Description                                  Status
CCTN3270       SAMPLE: XID Rejected                         INACTIVE
CONNECT        SAMPLE: FTP > 5 Mb                           INACTIVE
CONSOLE        SAMPLE: EZB4518E CTCA shutdown               INACTIVE
CUSTOM         SAMPLE: TCPIP Interface Inactive             INACTIVE
CUSTOM         SAMPLE: TCPIP Interface Active               INACTIVE
FTPFAIL        SAMPLE: FTP RETR Failed                      INACTIVE
ICMP           SAMPLE: Fragmentation Timeout                INACTIVE
LISTENER       SAMPLE: Check WEB Serv (port 80)             INACTIVE
 **END**


 F1=Help      F2=Split     F3=Exit      F4=Add       F5=Find      F6=Refresh
 F7=Backward  F8=Forward   F9=Swap                   F11=Right    F12=Skip
```

Defining an Event Detector

To define a new detector to the Event Detector Controls List, do this:

1. Press F4 (Add) on the TCP/IP Event Detector Controls List panel. The CAS : Valid Value List is displayed.

```
PROD---------------------- CAS : Valid Value List ----------------------------5
Command ===>                                                   Scroll ===> PAGE

                                                            S/=Select (one only)
   Field: Event Detector Type

   Full Value   Description
   CCTN3270     Cisco Channel TN3270 Log Messages
 S CONNECT      Connection Monitor
   CONSOLE      OS/390 Console Messages
   CUSTOM       Customized Event detector
   FTPFAIL      FTP Failures
   ICMP         ICMP Message Monitor
   LISTENER     Monitor Listening Ports
   **END**




 F1=Help      F2=Split     F3=Exit                   F5=Find      F6=Refresh
 F7=Backward  F8=Forward   F9=Swap
```

2. Select a value from the list and press ENTER. The corresponding detector definition panel is displayed.

```
PROD------------------- TCP/IP : Connection Detector ----------------------
Command ===>                                                Function=Add


Short Description  TEST sends more than 1000 bytes_          Status ACTIVE__

Find Connections Where:                                       (F4 to set)
**NONE**




Create Alert:                                                 (F5 to set)
Text........
Severity ...


Initiate Actions:                                             (F6 to set)
**NONE**


 F1=Help      F2=Split     F3=File      F4=Criteria  F5=Alert     F6=Actions
                           F9=Swap                                F12=Cancel
```

3. Complete the Short Description field. This description appears on the Event Detector Controls List. Use this in your own documentation.

4. Complete the Status field (ACTIVE or INACTIVE). Status controls whether this rule detects events. Making a detector inactive means that you can keep a definition, but not have it checked.
5. Press F4 (Criteria) to define the criteria for events to be monitored (see the section "Defining Criteria for a Detector" for details of how to do this).
6. Press F5 (Alert) to define the alert to be created (see the section "Defining an Alert" for details of how to do this).
7. Press F6 (Actions) to define any action to be taken by the system in response to an alert (see the section "Defining Actions to Be Initiated" for details of how to do this).
8. Press F3 (File). The TCP/IP : Event Detector Controls List is displayed with the new detector added.

Defining Criteria for a Detector

To define the criteria for events to be monitored, do this:

1. From the detector definition panel, press F4 (Criteria). The corresponding criteria panel is displayed.
2. Complete the fields on the criteria panel.

   Use F1 (Help) to obtain information about completing the fields on the various criteria panels.
3. Press F3 (OK). The detector definition panel you came from is displayed with the new criteria.

For information about specific criteria for detecting events, see the sections shown in the following table:

For this detector type ...	See section ...
Connection	"Monitoring Connections"
Custom	"Monitoring Custom Events"
FTP failures	"Monitoring FTP Failures"
Listener	"Monitoring Listeners"
Messages—Cisco channel TN3270 log	"Monitoring Messages—Cisco Channel Card TN3270 Log"
Messages—ICMP	"Monitoring Messages—ICMP"
Messages—OS/390 console	"Monitoring Messages—OS/390 Console"

Defining an Alert

To define the alert created when a monitor detects a defined event, do this:

1. From the detector definition panel, press F5 (Alert). The TCP/IP : Alert Definition panel is displayed.

   Note: For some detector types, this panel contains only the Description and Severity fields.

2. Complete the fields on the TCP/IP : Alert Definition panel.

   Use F1 (Help) to obtain information about completing the fields on this panel.

4. Press F3 (OK). You are returned to the detector definition panel that you came from where the alert definition details that you have entered are displayed.

Using Variables to Define an Alert

You can use variables to customize the text of alerts produced by a particular event detector. For details of the variables available for alerts produced by each type of detector, use F1 (Help) from the TCP/IP : Alert Definition panel.

Defining Actions to Be Initiated

The alert monitor provides a number of actions you can set up to run automatically when alerts arrive.

Action Name	Description
Notify	Send a user notification. This can be delivered in any of the following ways: ■ Broadcast message ■ TSO message ■ Electronic mail ■ Your own exit (NCL) routine To specify your own preferred method of receiving notifications, do this: 1. Enter the PASSWORD command on the User Notification Details panel. 2. Press Enter. The UAMS : User Password Maintenance panel is displayed. 3. Press F8 (Forward) to page forward to the Notification Details panel. 4. Enter details on the panel and press F3 (File).
Command	Issue a Management Services or system command.
Execute NCL	Run an NCL procedure.
Trouble Ticket	Create a trouble ticket based on characteristics of the alert. You would usually set this up to send a request via electronic mail.

If you want a particular detector to use automatic actions, define the actions when you are defining the detector.

To use the Trouble Ticket action, you must also implement a trouble ticket interface. See the section “Implementing a Trouble Ticket Interface” for details of how to do this.

To define an automatic action taken by the system in response to an alert, do this:

1. From the detector definition panel, press F6 (Actions). The TCP/IP : Alert Automated Actions panel is displayed.
2. Press F4 (Add). The Available Actions panel is displayed.
3. Select the action to use. An action-specific details panel is displayed.

```
PROD------------------- Alert Monitor : Run Command Details-------------------
Command ===>

Short Description... ______________________________________________
Command & Parameters.. ____________________________________________________
Command Parameters.... ____________________________________________________
**END**




 F1=Help      F2=Split     F3=File
                           F9=Swap                                F12=Cancel
```

4. Complete the action-specific details on the panel.

   Use F1 (Help) to obtain information about completing each panel.
5. Press F3 (File). You are returned to the TCP/IP : Alert Automated Actions panel, with a message that the selected action has been added.
6. Press F3 (File). You are returned to the detector definition panel that you came from.

Monitoring Connections

You can set up detectors to poll connection information at defined intervals and to create alerts according to the criteria that you define. The poll interval is set in the $IP IPMONITOR parameter group.

Criteria for Connection Detectors

You can use combinations of any of the following as criteria for a connection detector to create an alert:

- A job name or protocol—as for the Taskname column on connection lists
- TCP status—as for the Status column on connection lists, except that connections with Listen status are not monitored
- Byte count—bytes in and bytes out on connection lists
- A full or local remote IP address or local port number—must match connection list values
- A full or generic remote IP address or remote port number—must match connection list values
- Idle or elapsed time threshold

Tip: Before setting up an event detector for connections, use the LC—List Connections option to find the kind of connection that you want to monitor, and take note of the values displayed in the various columns.

Defining a Connection Detector

To define a connection detector to the Event Detector Controls List, perform the steps described in the section "Defining an Event Detector". Select CONNECT as the Alert Detector Type.

An Example of Defining a Connection Detector

Suppose you want to drop FTP data connections that have been idle for more than 10 minutes. To define an event detector for this purpose, do this:

1. From the TCP/IP : Connection Detector panel, press F4 (Criteria.) The TCP/IP : Connection Criteria panel is displayed.
2. Enter the following criteria:
   - Task Name = FTPSRV (Job name of FTP server)
   - Local Port = FTP-D (FTP data transfer)
   - Idle Time Over = 00:10 (10 minutes)
3. Press F3 (OK). You are returned to the TCP/IP : Connection Detector panel.
4. Press F6 (Action). The Alert Automated Actions panel is displayed.
5. Press F4 (Add). The Alert Available Actions panel is displayed.
6. Select RUN-COMMAND and press ENTER. The Run Command Details panel is displayed.
7. Enter a short description of the connection detector, and then enter **NETSTAT DROP &$IPCONNID** in the Command & Parameters field.
8. Press F3 (File). You are returned to the Alert Automated Actions panel, with RUN-COMMAND added to the list of actions.
9. Press F3 (OK). You are returned to the TCP/IP : Connection Detector panel, with RUN-COMMAND added to Initiate Actions section.

Monitoring Custom Events

You can set up customized event detectors to monitor events other than those monitored by the standard types of event detectors.

This option is intended for the advanced administrator. You can trigger a detector based on Event Distribution Services (EDS) events. The criteria that you specify correspond to the values that you can specify on the PROFILE EDS enable.

Events can be raised by your own processes or by system events; for example, LINK START.

Defining a Custom Event Detector

To define a custom event detector to the Event Detector Controls List, perform the steps described in the section "Defining an Event Detector". Select CUSTOM as the Alert Detector Type.

Monitoring FTP Failures

FTP failures detected by the FTP logging function can be declared as alerts. An FTP is considered to have failed if there is a response code of other than 0 or 250 in the FTP client or server event.

To have this facility you must be running the FTP logging receiver. For information on this see the *NetMaster Network Performance for TCP/IP Implementation Guide*.

Criteria for FTP Failure Detectors

The criteria that you can use to define an FTPFAIL detector are:

- An FTP subcommand (for example, RETR or STOR), or * for all
- A full or generic remote IP address
- A full or generic dataset name
- A full FTP server name, or * for all.

Defining an FTP Failure Detector

To define an FTP failure detector to the Event Detector Controls List, perform the steps described in the section "Defining an Event Detector". Select FTPFAIL as the Alert Detector Type.

An Example of Defining an FTP Failure Detector

To create an alert if the receiving of a production dataset fails, enter the following details on the TCP/IP : FTP Failure Criteria panel:

- FTP Command—STOR
- Dataset Name—PROD.ERROR.LOG

Monitoring Listeners

You can set up an event detector to monitor listeners to ensure that certain services are available, for example, ROUTED. This detector uses SNMP technology to poll MIB-II variables for listening ports on the local system.

The poll interval is set in the $IP IPMONITOR parameters group (option **/ICS**).

Criteria for Listener Detectors

The criteria that you can use for a listener detector are:

- The port number of the listener
- The type of listener (TCP or UDP)
- The local IP address of the listener, if applicable

Normally you need not specify a local IP address. The only cases in which you need to use this field are those in which a server binds to a specific local address.

Do You Need to Specify a Listener Local Address?

To check if you need to specify a local IP address for a listener, do this:

1. Enter **/LISTCON** at the ===> prompt. The TCP/IP : Connection List Criteria panel is displayed.
2. Change Fast Search to NO.
3. Press F6 to action. The TCP/IP : Connection List panel is displayed.
4. Press F11 (Right) to display the Status column.
5. Locate an entry whose status is Listen or UDP.
6. Check the Local Host value for this entry.

   If an IP address or host name is shown, then you need to specify the IP address in the criteria for your listener detector.

Defining a Listener Detector

To define a listener detector to the Event Detector Controls List, perform the steps described in the section "Defining an Event Detector". Select LISTENER as the Alert Detector Type.

An Example of Defining a Listener Detector

To ensure that ROUTED is available on UDP port 520, enter the following details on the TCP/IP : Listener Criteria panel:

- Port Number **520**
- Type **UDP**

Monitoring Messages—Cisco Channel Card TN3270 Log

You can create alerts based on the messages logged by the channel card TN3270 server to the TN3270 server log, for example connection errors.

You must have the channel card defined with logging active. See the chapter, "Setting up IP Resource Monitoring".

Note: To report on Telnet connections for Cisco TN3270 server, you do not need to set up event detectors. Reporting is enabled by the TN3270 Reporting Active field of the Cisco Channel Card Definition panel.

Criteria for Cisco Channel TN3270 Log Messages to Be Monitored

The criteria for monitoring Cisco channel TN3270 log messages are:

- The message number
- The Cisco channel card name (optional)—as defined to NetMaster for TCP/IP. See the chapter, "Setting up IP Resource Monitoring".

Defining a Cisco Channel TN3270 Log Detector

To define a Cisco channel TN3270 log detector to the Event Detector Controls List, perform the steps described in the section "Defining an Event Detector". Select CCTN3270 as the Alert Detector Type.

Monitoring Messages—ICMP

You can monitor ICMP error messages that potentially indicate problems in your TCP/IP network.

ICMP messages produced by a TCP/IP network are of two kinds:

- Queries
- Errors

Note: Some errors are not monitored, because they are already handled by problem diagnosis utilities such as ping and traceroute. These include 'hop limit exceeded' and 'port unreachable' messages, which are normal, expected responses to traceroute.

This category also includes ICMP error messages about other ICMP messages.

The types of ICMP error messages that you can monitor are:

Type	Description	Code
3	Destination unreachable	All except code 3—port unreachable, which is a standard trace route result
4	Source quench	Not applicable
5	Redirect	All
11	Time exceeded	1—fragment reassembly timeout
12	Parameter problem	Any

Criteria for ICMP Message Detectors

The three criteria by which you can filter ICMP messages to be monitored are:

- ICMP message type
- ICMP message source
- Original message destination address

Valid values for the ICMP message type are:

Abbrev	Full Value	Description
D	DEST-UNREACHABLE	Destination unreachable
F	FRAGMENT-TIMEOUT	Fragment reassembly timeout
P	PARAMETER PROBLEM	Parameter problem in datagram
R	REDIRECT	Redirect
S	SOURCE-QUENCH	Source quench

Defining an ICMP Message Detector

To define an ICMP message detector to the Event Detector Controls List, perform the steps described in the section "Defining an Event Detector". Select ICMP as the Alert Detector Type.

Tips for Using ICMP Message Detectors

FRAGMENT-TIMEOUT can be a good early warning indicator of performance and unexpected disconnection problems.

REDIRECT is a normal part of IP operation. However, repeated REDIRECT messages may indicate a routing problem, such as a loop caused by a normal route being unavailable.

If you have *any* ICMP rules, then *all* ICMP messages must be processed by the system. Therefore, having no active rules saves CPU usage. You may want to activate ICMP rules only when diagnosing a difficult problem.

Monitoring Messages—OS/390 Console

You can monitor OS/390 console messages to identify problems with your TCP/IP environment.

Criteria for OS/390 Console Message Detectors

The criteria for screening messages are:

- Message text
- Job name

Defining an OS/390 Console Message Detector

To define an OS/390 console message detector to the Event Detector Controls List, perform the steps described in the section “Defining an Event Detector”. Select CONSOLE as the Alert Detector Type.

Tips for Using OS/390 Console Message Detectors

Important messages for which you may want to create alerts are messages about network interface status.

For example, on TCPaccess you could create a console message rule for ACC169A Lni-1 Device Hung.

Implementing a Trouble Ticket Interface

The alert monitor provides a number of actions you can set up to run automatically when alerts arrive:

- Notify
- Execute a command
- Execute NCL
- Generate a trouble ticket

If you want a particular event detector to use automatic actions, you need to define the actions when you are defining the detector.

To use the trouble ticket action, you must also implement a trouble ticket interface.

The alert monitor supports two interfaces to trouble ticket systems.

- Electronic mail, where an e-mail describing the problem can be sent to a trouble ticket application or to a particular person. This method can be used to send problems to many types of trouble ticket applications.
- Custom, where you can write your own NCL code to deliver the trouble ticket to an application by whatever means you choose.

To implement a trouble ticket interface, you need to define the trouble ticket interface between your NetMaster for TCP/IP system and the alert monitor (see the section “Defining a Trouble Ticket Interface”).

If you want the operator to supply information when requesting creation of a trouble ticket, you also need to set up the trouble ticket data entry definition (see the section “Setting Up the Trouble Ticket Data Definition”).

Defining a Trouble Ticket Interface

To define a trouble ticket interface between your NetMaster for TCP/IP system and the Alert Monitor, do this:

1. Enter **/ALADMIN** at the ==> prompt. The Alert Monitor : Administration Menu is displayed.
2. Select option **I** - Define Trouble Ticket Interface. The Alert Monitor : Interface Definition panel is displayed.
3. In the Interface Type field, specify the type of interface you want to define.

   Enter a question mark (**?**) in this field to obtain a selection list of valid values.
4. Press F6 (Action). A panel is displayed where you can define your trouble ticket interface. The type of panel displayed varies, depending on the interface type that you specified. See the sections "Defining an E-mail Trouble Ticket Interface" and "Defining a Custom Trouble Ticket Interface" for further details.

Defining an E-mail Trouble Ticket Interface

If you specified EMAIL as the interface type on the Alert Monitor : Interface Definition panel, press F6 (Action) to display the Alert Monitor : Email a Trouble Ticket panel.

```
PROD------------Alert Monitor : Email a Trouble Ticket---------------------
Command ===>                                Function=Update Scroll ===> PAGE

Mail Address                 ______________________________________________
Host Name      (IBM)         ________
SMTP Node Name (IBM)         ________
SMTP Job Name  (IBM)         SMTP32__
SMTP DEST Id (TCPaccess)     ________
Exit Procedure Name          ________
Subject                      ______________________________________________

                                Enter Mail Text Below

**** ****************************** TOP OF DATA ******************************
0001 Application ID             : &$AMAPPLID
0002 Alert Creation Date        : &$AMDATE
0003 Alert Creation Time        : &$AMTIME
0004
0005 Severity                   : &$AMSEVERITY
0006 Priority                   : &$AMPRIORITY
0007
**** ***************************** BOTTOM OF DATA ****************************


F1=Help       F2=Split      F3=File      F4=Save     F5=Find       F6=Change
F7=Backward   F8=Forward    F9=Swap     F10=Left    F11=Right     F12=Cancel
```

To define your e-mail trouble ticket interface, do this:

1. Enter values in the input fields in the top section of the panel.

   Use F1 (Help) to obtain information about completing these fields.

2. Complete the Enter Mail Text Below section of the panel, which is free format.

   Use F1 (Help) to obtain information about completing this section.

3. Press F3 (File). You are returned to the Alert Monitor Administration Menu and your interface definition is saved.

Defining a Custom Trouble Ticket Interface

If you specified CUSTOM as the interface type on the Alert Monitor : Interface Definition panel, press F6 (Action) to display the Alert Monitor : Custom Trouble Ticket panel.

```
PROD-------------Alert Monitor : Custom Trouble Ticket---------------------
Command ===>                                Function=Update Scroll ===> PAGE

Procedure Name  ________

                            Enter Parameters Below

**** ****************************** TOP OF DATA ******************************
0001
**** ***************************** BOTTOM OF DATA ***************************





F1=Help       F2=Split      F3=File      F4=Save     F5=Find       F6=Change
F7=Backward   F8=Forward    F9=Swap     F10=Left    F11=Right     F12=Cancel
```

To define your custom trouble ticket interface, do this:

1. In the Procedure Name input field, enter the name of your NCL procedure for delivering trouble tickets.

2. In the Enter Parameters Below section of the panel, specify any parameters that you want the NCL procedure to receive. This section is free format.

   Use F1 (Help) to obtain information about completing this section.

3. Press F3 (File). You are returned to the Alert Monitor Administration Menu and your interface definition is saved.

Setting Up the Trouble Ticket Data Definition

If you want the operator to supply information when requesting a trouble ticket, you need to set up the trouble ticket data entry definition.

To define the information you want the operator to supply, do this:

1. On the Alert Monitor Administration Menu, select option **D** - Trouble Ticket Data Definition. The Alert Monitor : Trouble Ticket Data Entry Definition panel is displayed.

```
PROD-------- Alert Monitor : Trouble Ticket Data Entry Definition ----------
Command ===>                                Function=Update Scroll ===> PAGE

**** **************************** TOP OF DATA ****************************
0001 FIELD NAME=PRIORITY
0002  VALUE="3"
0003  DESC="Trouble Priority"
0004  COMMENT="                  (1=High, 4=Low)"
0005  REQUIRED=NO
0006  LENGTH=2
**** ***************************** BOTTOM OF DATA *****************************









F1=Help       F2=Split      F3=File      F4=Save     F5=Find       F6=Change
F7=Backward   F8=Forward    F9=Swap     F10=Left    F11=Right     F12=Cancel
```

2. In the free-format data entry section of the panel, enter the data entry definition for the panel that the operator will use when creating a trouble ticket.

   Use F1 (Help) to obtain information about completing this section.

3. Press F3 (File). You are returned to the Alert Monitor Administration Menu and your trouble ticket data entry definition is saved.

Implementing the Alert History Function

The alert monitor retains data in an alert history file. To specify how long alerts are to be retained in this file, do this:

1. Enter **/ICS** at the ===> prompt. The ICS : Customization Panel appears.
2. Enter **U** in front of the $NM ALERTHIST parameter group. The Alert History File Specification details appear.
3. In the Days to Retain Alerts in History File field, specify the number of days that you want alerts to be retained in the history file.
4. In the Time of Day for Alert Purge field, specify the time of day (in the format *hh.mm*) at which the Alert Monitor will delete alerts that have been in the history file longer than the retain setting.
5. Press F3 (File) to save your settings.

Reorganizing Files and Monitoring Space Usage

To reorganize the Alert History database to reclaim dead space, do this:

1. Copy (REPRO) the History to a backup file.
2. Delete and redefine the original file.
3. Copy the data back in to the redefined file.

You should also monitor the amount of disk space used by the dataset, to estimate the optimal file size and optimal frequency of reorganization.

Applying Alert Monitor Filtering

You can filter the alerts raised by the alert monitor by applying a set of criteria to each of the fields within the alert. The filters that you create can be named and stored for later use. To apply filtering to the alerts that are raised, do this:

1. Enter **/ALADMIN.F** at the ===> prompt. The Alert Monitor : Filter Definition List panel is displayed.
2. To create a new filter press F4 (Add). The Alert Monitor Filter panel is displayed.

```
PROD----------------- Alert Monitor : Filter Definition ---------------Func=ADD
Command ===>                                                   Scroll ===> PAGE

| Filter Definition -------------------------------------------------------|
| Name .......... ___________                                               |
| Description ... ___________________________________________________________ |
| Last updated at          On                  By                           |
-----------------------------------------------------------------------------|
--Filter Expression -----------------------------------------------------------|
|                                                                            |
|                                                D=Delete I=Insert R=Repeat|
|   "("  Field    Opr  Value                                   Gen ")" Bool |
|    ___ _________ ___ ___________________________________________ ___ ___ ___  |
|    ___ _________ ___ ___________________________________________ ___ ___ ___  |
|    ___ _________ ___ ___________________________________________ ___ ___ ___  |
|    ___ _________ ___ ___________________________________________ ___ ___ ___  |
|    ___ _________ ___ ___________________________________________ ___ ___ ___  |
|    ___ _________ ___ ___________________________________________ ___ ___ ___  |
|    ___ _________ ___ ___________________________________________ ___ ___ ___  |
|  **END**                                                                   |
|                                                                            |
|  F1=Help      F2=Split    F3=File     F4=Save                              |
|  F7=Backward  F8=Forward  F9=Swap                          F12=Cancel      |
 ----------------------------------------------------------------------------
```

3. Enter the name and description of the filter.
4. Enter the values that you require into the Field, Opr, Value, Gen, and Bool fields. For a full description of these fields press F1(Help).
5. Press F3 (File) to save the changes.

Forwarding Alerts

Alerts are normally displayed on the Alert Monitor display. However, you can also forward them to other platforms:

- UNIX platforms as SNMP traps
- NetMaster for SNA (NEWS) or NetView (TME10) systems, as generic alert NMVTs
- Unicenter TNG

You can apply filter criteria to forward different types of alerts to different platforms.

Implementation

To enable alert forwarding, execute the $AMEVFWD command. This command has two groups of parameters:

- Those which describe the destination platform
- Those which supply the filtering criteria

To implement Alert Forwarding you need to execute an $AMEVFWD command for each combination of platform and filtering parameters. It is recommended that you include these commands in your READY procedure so that they are enabled at initialization and remain in effect while the system is running.

See the section "$AMEVFWD" for the syntax of the $AMEVFWD command.

The SNMP Trap Definition

The MIB definition for alerts forwarded as SNMP traps is provided in member $AMTRAP, supplied in the *dsnpref*.MS500.INSTAL dataset. You can download this member to your UNIX system and compile it.

Note: When copying this member to your UNIX system, you can rename it to avoid problems on some UNIX systems where the $ sign has special meaning.

The supplied MIB defines two traps with the following object identifiers:

- $AMTRAP = 1.3.6.1.4.1.1126.1.2.1.2 (for an alert)
- $AMTRAPC = 1.3.6.1.4.1.1126.1.2.1.3 (when an alert is cleared)

$AMEVFWD

Function

Implements alert forwarding.

```
$AMEVFWD    DESTTYPE={ TNGTRAP|NMVT | SNMPTRAP }
            DESTADDR=ipaddr
           [ DESTPORT=ipport ]
           [ COMMNAME=community ]
           [ NETVALRT=ppiname ]
           [ FILTER=filtername ]
           [ CLEAR={ YES | NO } ]
            ALERTS= { NEW | ALL }
           [ IPINACT={ WAIT | IGNORE } ]
```

Use

The $AMEVFWD command is used to implement alert forwarding.

Operands

The destination platform is specified by the DESTTYPE, DESTADDR, DESTPORT, COMMNAME, and NETVALRT parameters. The filtering criteria are specified by the FILTER parameter. Only alerts that meet these criteria are forwarded.

Destination Details

Destinations can be TNG traps , SNMP traps, or (Generic Alert) NMVTs.

Forwarding Alerts to Unicenter TNG

```
$AMEVFWD    DESTTYPE=TNGTRAP
            DESTADDR=ipaddr
           [ DESTPORT=ipport ]
           [ COMMNAME=community ]
           [ FILTER=filtername ]
           [ IPINACT={ WAIT | IGNORE } ]
```

DESTTYPE=TNGTRAP

Specifies that the alerts are forwarded as SNMP traps optimized for the Unicenter TNG Event Console (TNGTRAP). It differs from SNMPTRAP as follows:

- Providing four fields only: system, product, severity, and text
- Putting the text field last
- Passing new alerts only

See the *dsnpref*.MS500.INSTAL($AMTRAP) MIB definition for a list of the fields (variables) that are present in the standard SNMP traps.

DESTADDR=*ipaddr*

Specifies the IP address to which alerts are to be forwarded. The address can be in dotted notation (for example, 123.45.6.78) or a host name address (for example, network.operations.com).

COMMNAME=*community*

Specifies a community name at the destination. The default is *public* in lower case.

DESTPORT=*ipport*

Specifies a port number at the destination. The default is 162.

IPINACT=WAIT|IGNORE

Specifies what to do when SNMP traps cannot be forwarded because the TCP/IP interface is inactive:

- WAIT—suspends alert forwarding until the interface becomes active. This value is the default.
- IGNORE—discards the alerts when the interface is inactive.

Sending Alerts as SNMP Traps

```
$AMEVFWD        DESTTYPE=SNMPTRAP
                DESTADDR=ipaddr
               [ DESTPORT=ipport ]
               [ COMMNAME=community ]
               [ FILTER=filtername ]
               [ CLEAR={ YES | NO } ]
                ALERTS= { NEW | ALL }
               [ IPINACT={ WAIT | IGNORE } ]
```

DESTTYPE=SNMPTRAP

For SNMP traps, specific destination details are required, as follows:

DESTADDR=*ipaddr*

This is mandatory. It specifies the destination (IP) address as either a valid IP address in dotted notation (for example, 123.45.6.78) or a host name address (for example, network.operations.com).

DESTPORT=*ipport*

This is optional. It specifies the port number (at the destination address). If specified, it must be a number in the range 1 to 65535. If omitted, it defaults to 162.

COMMNAME=*community*

Specifies the community name (at the destination address). Defaults to *public* (in lower case). Community names are case-sensitive. When a value is specified, ensure that the value is in the correct case.

If the procedure is started from another procedure, for example within the READY procedure, ensure that the statement remains case-sensitive at execution time. By default, NCL procedures such as the READY procedure convert data to upper case during assignment. Disable upper case translation over the $AMEVFWD procedure invocation with the &CONTROL UCASE|NOUCASE statement. For example:

```
&CONTROL NOUCASE
START $AMEVFWD DESTTYPE=SNMPTRAP DESTADDR=xxx +
  DESTPORT=162 COMMNAME=public
&CONTROL UCASE
```

IPINACT=WAIT|IGNORE

Specifies what to do when SNMP traps cannot be forwarded because the TCP/IP interface is inactive:

- WAIT—suspends alert forwarding until the interface becomes active. This value is the default.
- IGNORE—discards the alerts when the interface is inactive.

Sending Alerts as NMVTs

```
$AMEVFWD        DESTTYPE=NMVT
                [ NETVALRT=ppiname ]
                [ FILTER=filtername ]
                [ CLEAR={ YES | NO } ]
                  ALERTS= { NEW | ALL }
```

DESTTYPE=NMVT

NMVTs are queued to the NETVALRT PPI receiver queue and are then processed by either NetMaster for SNA or NetView.

NETVALRT=*ppiname*

The default name for the NETVALRT PPI receiver is NETVALRT. If you have used an alternate name, use this operand to specify the alternate name.

Forwarding to NetMaster for SNA

To receive alerts you must have the PPI receiver active. This is started on the destination NetMaster for SNA system.

The forwarded alerts are recorded as Operator Notification EVENTS for the specific resource, which is the primary system name (PRI= init parm) of system which forwarded the alert.

Forwarding to NetView

To receive alerts in a NetView region you must have the CNMCALRT task defined and active. To do this:

1. Check the DSIDMN (or DSIDMNB) member in the DSIPARM PDS.
2. Ensure that the CNMCALRT task is included and is initialized (INIT=Y). For example:

```
TASK  MOD=CNMCALRT,TSKID=CNMCALRT,PRI=6,INIT=Y
```

Note: This statement is necessary for the OS/390 software alert forwarding function.

The alerts are formatted as Operator Notification generic alerts.

Filtering Details

There are three options for filtering. You can:

- Apply a named filter using FILTER=*filtername*.
- Apply filtering criteria with the CLEAR and ALERTS operands—these can be applied, regardless of the destination details. Both are optional.
- Not apply a filter.

Applying a Named Filter

FILTER=*filtername*

This specifies the name of the filter to apply. If specified, the named filter must exist. For more information see Applying Alert Monitor Filtering.

Predefined Filter Criteria

CLEAR=YES|NO

This parameter relates to alerts that are cleared. If set to YES alerts for cleared alert events are forwarded. This parameter is optional.

ALERTS=NEW|ALL

This parameter controls how alerts generated before alert forwarding is enabled are handled.

If set to NEW—only new alerts are forwarded. This option would normally be used if alert monitoring is restarted.

If set to ALL—current alerts, and new alerts as they arrive, are forwarded. Use this option to make the alert forwarding function behave the same as the Alert Monitor option at a terminal. This option would normally be used when the alert monitor is started during initialization of the system region.

Examples

To send all CIPLOG alerts from the local system as SNMP traps, enter:

```
START $AMEVFWD DESTTYPE=SNMPTRAP+
DESTADDR=network.operations.com +
DESTPORT=4567 FILTER=CIPLOG
```

To send all alerts from SOLVE5 as NMVTs, enter:

```
START $AMEVFWD DESTTYPE=NMVT SYSTEMID=SOLVE5
```

Chapter

8

Initializing Multiple Systems

This chapter describes how to configure multiple systems with an initialization file.

This chapter contains the topic:

- Configuring Multiple Systems with Initialization Files

Configuring Multiple Systems with Initialization Files

Each NetMaster system is configured for the environment that it is installed in. Multiple systems can be configured with an initialization file. This removes the need to configure each system with ICS.

The tasks outlined below show how to configure a system from an initialization file.

Task 1—Generating an Initialization File

The initialization file is produced from a running NetMaster system. To generate an initialization file, do this:

1. From the Primary Menu enter **/INIT**. The Initialization and Customization Services panel is displayed.
2. Select option **G** - Generate INI Procedure. The ICS : Generate INI Procedure panel is displayed.
3. In the Output Dataset Information section, enter the dataset name and the member name of the file. The output destination dataset must be in the commands concatenation of the RUNSYSIN member for the system it will be used in.
4. If replacing an existing member, enter **YES** in the Replace Member field.
5. Press F6 (Action) to generate the INI file. The INI file is generated.
6. Make a note of the dataset and member names and press F3 (Exit).

Task 2—Configuring the Initialization File

The initialization file must be configured before it can be used on other systems. There are two ways of doing this:

- Configure an individual initialization file for each system
- Configure a common initialization file for multiple systems

You can use system variables and static system variables with both of these methods. The variables substitute for the initialization parameters in the INI file.

Subtask 2.1—Configuring Individual Initialization Files

To configure an individual initialization file for each system, do this:

1. Use your TSO editing tool to open the INI file in edit mode.
2. Substitute the parameters in the initialization file with one of the following:
   - Hard coded dataset names for the system the file will be used in
   - System variables

   This enables the INI file to work in regions with different datasets than the region it was generated in.
3. Save the changes to the initialization file.
4. Copy the initialization file to the system that will use it.
5. Repeat steps 1 to 4 for each initialization file needed.

Subtask 2.2—Configuring a Common Initialization File

To configure a common initialization file, do this:

1. Create a dataset that is available to every system to be initialized from the common initialization file, for example, PROD.TCPIP.INIFILES.
2. Add the dataset to the commands concatenation of the RUNSYSIN member to every system to be initialized from the common initialization file.
3. Copy the initialization file generated in Task 1 into the dataset.
4. Use your TSO editing tool to open the initialization file in edit mode.
5. Replace the relevant generated variables in the initialization file with the system variables shown in the following table.

System Variable	Description
&ZDSNQLCL	The local dataset qualifier
&ZDSNQSHR	The shared dataset qualifier
&ZACBNAME	The primary VTAM ACB name used by the system
&ZNMDID	The domain identifier
&ZNMSUP	The system user prefix

6. Replace the relevant generated variables in the initialization file with the OS/390 static system symbols as shown in the following table.

Static System Symbols	Description
&SYSCLONE	The short name for the system
&SYSNAME	The name of the system
&SYSPLEX	The name of the sysplex
&SYSR1	The IPL VOLSER

7. Save the changes to the initialization file.

Task 3—Initializing Your Region From an Initialization File

The name of the initialization member must be specified by the INIFILE parameter in the RUNSYSIN Member.

Updating your RUNSYSIN member will cause your region to set its initialization parameters from the initialization file. To update your RUNSYSIN member, do this:

1. Use a text editor to open your RUNSYSIN member.
2. Insert the line PPREF='INIFILE=*membername*' into your RUNSYSIN member.
3. Save the member.

Chapter

9

Setting Up Security

This chapter describes how to set up security in the NetMaster for TCP/IP region.

This chapter contains the following topics:

- About Security
- Security Considerations for Existing Users
- Defining NetMaster Users
- Managing Users in a Multisystem Environment
- External Security Packages

About Security

Access to a NetMaster for TCP/IP region is controlled by the User ID Access Maintenance Subsystem (UAMS).

NetMaster for TCP/IP supplies five sample group definitions that are generated during installation. These groups and their characteristics are described below:

- $RMADMIN—administrator—this group of users has access to all NetMaster administrative functions, such as adding users. An administrator has access to all the menu options and is authorized to delete database records.
- $RMOPER—operator—this group of users has access to a restricted subset of NetMaster functions. An operator does not have access to all the menu options and is not authorized to delete database records.
- $RMNOPER—network operator—this group of users has similar access to NetMaster functions as an operator. Network operators can manage network operations but are not authorized to manage system operations.
- $RMMON—monitor—this group of users has access to a restricted subset of NetMaster functions. A monitor user does not have access to all the menu options and can browse but not update or delete database records.
- $RMBUSER—background user—this group of users has region or engine component authorization.

Important: *Do not modify the supplied $RMBUSER group definition, because this could impede the operation of the NetMaster product.*

Security Considerations for Existing Users

If you are using a pre-existing UAMS database, perform the following tasks to ensure that users are properly authorized to operate in the region:

Ensure that the group definitions are authorized for TCP/IP.

Ensure that the background users are defined by using the $RMBUSER group definition.

Checking Existing User Group Definitions

Ensure that the group definitions are authorized for TCP/IP as follows:

1. Enter the **/UAMS** shortcut to access the UAMS maintenance function.
2. Enter **L** at the ===> prompt and **$RM** in the User field. The group definitions are listed.
3. Update each of the $RMADMIN, $RMOPER, $RMNOPER, $RMMON, and $RMBUSER definitions to ensure that the following fields are specified correctly:

Field	Value
Network Management field on the Access Authorities panel (3rd panel)	Y
TCP/IP field on the Network Management Details panel (8th panel)	2

Customizing Existing Background User Definitions

The $RMBUSER group definition for background regions is defined when a NetMaster for TCP/IP region starts the first time. The following UAMS background user definitions (where *xxxx* is the domain ID) are also generated and linked to the UAMS group:

User ID	Description
xxxxAOMP	AOM procedure
xxxxBLOG	Logger
xxxxBMON	Monitor
xxxxBSVR	Server
xxxxBSYS	System
xxxxLOGP	Log procedure

Note: The domain ID of the region is specified in the RUNSYSIN member during setup.

Updating Background User Definitions

If you set up your region by using a pre-existing UAMS database in which the background users are already defined for your region, those background user definitions are not replaced. To enable the new region to work correctly, you must update those background user definitions by associating the definitions to the $RMBUSER group ID. You can do this by completing the following steps:

1. Enter the **/UAMS** shortcut. The UAMS maintenance function is displayed.
2. Enter **L** at the ===> prompt and *xxxx* in the User field. A list of the background user definitions is displayed.
3. Update a required background user ID by entering $RMBUSER in the Group ID field.
4. Press F3 (File) to file the change.
5. Repeat steps 3 and 4 for each of the background users.

6. After you finish updating the user definitions, enter **CMD** at the ===> prompt to access the Command Entry panel.

7. Enter the following command for each of the background users to invoke the changes.

   `SUBMIT USER=`*`background-user-id`* `SIGNON`

   The following example invokes the changes for the background logger:

   `SUBMIT USER=`*`xxxx`*`BLOG SIGNON`

Defining NetMaster Users

Define users in a NetMaster for TCP/IP region through the user profiles.

To add a user profile, do this:

1. Enter the **/ASADMIN.UP** path to display the User Profile List panel.

2. Press F4 (Add) to add a new user profile. The action presents you with the first panel in the user profile definition, the User Description panel.

3. Enter the user ID, password, and name, and assign the user to a security group. See the section, About Security, for a list of the supplied groups.

   If you assign a user to a security group, then when you save the profile, a UAMS user ID definition is automatically created.

4. Press F3 (File) or F4 (Save) to save the new record. (Pressing F3 saves the profile and exits the profile definition. Pressing F4 saves the profile but leaves the definition open so that another profile can be added.)

Tip: Once you have defined one user profile, you can use the **C** (Copy) action to duplicate an existing user profile and change the values for another user in the copied record as required.

Managing Users in a Multisystem Environment

If you have implemented a multisystem environment, where several NetMaster for TCP/IP regions are linked together, you can define a user in one region and have the user definition copied to the other linked regions.

To enable this feature, you have to customize the SECSHIPPING parameter group.

Implementing User Definition Synchronization Across Linked Regions

To synchronize UAMS records across all active linked regions, you need to enable the automatic propagation facility in all those regions. Do not use this facility if the linked regions share a single UAMS database or a common security exit.

If automatic propagation is enabled, it only occurs when a user ID is added or updated by using the method described in the section, Defining Users.

Synchronization depends on whether you make an update from a focal point region or a subordinate region as follows:

If the update is in a ...	The update is synchronized across ...
Focal point region	All active linked regions that are enabled for this feature.
Subordinate region	Only those active linked focal point regions that are enabled for this feature.

Updating the SECSHIPPING Parameter Group

Enabling or disabling the synchronization of UAMS records across multiple regions is the function of the parameter group SECSHIPPING. Synchronization occurs only between the linked regions in which synchronization is enabled.

To update the parameter group, do this:

1. Enter the **/ICS** shortcut. The Customization Parameters panel is displayed.
2. Enter **U** (Update) beside the SECSHIPPING parameter group, which is located under the SECURITY category.

3. With the SECSHIPPING - Ship UAMS Maintenance panel displayed, the following settings can be made:

   – Respond **YES** to both questions to allow all add, update, delete, *and* password change operations for UAMS records to be propagated to linked regions.

   – Respond **NO** to both questions to disable user definition synchronization. If an update is requested from a remote region through this facility, it is refused.

***Important**: If you respond YES to the question Ship to Linked Systems? and NO to the question Including Password Changes?, password changes are not synchronized. (This setting is for regions that do not share a UAMS database but use the RACF distributed password update facility.)*

Synchronization Report

The UAMS database is updated when you update the User Description panel of a user profile and save it. If user definition synchronization is enabled, UAMS updates are sent to the linked regions immediately. A UAMS update report is displayed immediately, indicating the success or failure of those updates.

Troubleshooting

Possible reasons for a remote region update not working include:

- The region is not profiled for remote updates. To profile it for remote updates, specify **YES** in the Ship to Linked Systems? field of its SECSHIPPING parameter group.
- The link or remote region is not active. (Because UAMS update records are not written to a staging file, no record of the update is retained, and the UAMS record in the remote region is not updated.)
- The user does not have UAMS administration authority on the remote region.
- The record or database is locked.
- The UAMS record does not exist. This condition occurs when you update a user profile record, without providing a new initial password, and there is no associated UAMS record in a remote region. Remedy this by supplying an initial password. This results in the automatic generation of a UAMS record for the remote region, and resets the users password across the other regions.
- You based the user access on a customized user group that does not exist in the remote region.

External Security Packages

External security packages, such as RACF or CA-ACF2, can provide minimal or full security checking.

If your organization has an external security package, access to that package is provided through one of the following types of exit:

- *Partial security exit* – password and logon access maintenance is controlled by the external security package while UAMS stores the user definitions.
- *Full security exit* – all security functions are maintained and stored by your external security package.

The following sample security exits are provided with NetMaster and can be found in the following libraries:

- *The SMP target zone library,* dsnpref.*INSTAL*
- *The SMP distribution zone library,* dsnpref.*BASE.INSTALL*

Sample Exit	Description
CCRACFFX	Full security exit for RACF
CCACF2FX	Full security exit for CA-ACF2
NMSAFPX	SAF partial security exit

The NMSAFPX partial security exit is a SAF based security exit that supports UTOKENS. SAF is the IBM System Authorization Facility and is the agreed standard for the encoding of requests that require security checking.

SAF is documented in the IBM *External Security Interface (RACROUTE) Macro Reference for MVS and VM* manual. You should refer to the documentation for your security package to see whether the package supports SAF-formatted calls.

For OS/390 systems, it is recommended that you use the NMSAFPX partial security exit as it supports RACF and CA-ACF2 security packages.

When the exit has been written, link it to create a load module. Place the module in the load library, or another library concatenated to the load library through the STEPLIB DD statement in the started task JCL. Change the JCL parameters to include the SEC={ load-module-name } parameter.

For further information, about using SAF samples and NPF, see the *Automation Services Administrator Guide.*

Chapter

10 Troubleshooting

This chapter describes solutions to common problems that might be encountered when using NetMaster for TCP/IP.

This chapter contains the following topics:

- About Troubleshooting
- Performing NetMaster for TCP/IP Self Test
- Displaying the NetMaster for TCP/IP Initialization Log
- Troubleshooting OS/390 SNMP Problems
- Commonly Encountered Errors
- Interpreting Socket Error Codes

About Troubleshooting

NetMaster for TCP/IP provides two features to help you diagnose and locate the source of problems within your product:

- Self test
- Initialization log

These two features are described in the following two sections.

The last section in this chapter provides a list of common errors, their cause, and the actions required to rectify the problems.

Performing NetMaster for TCP/IP Self Test

The NetMaster for TCP/IP self test can be used to help diagnose problems with your product. It provides checks for:

- The sockets interface of your TCP/IP product
- The management interface of your TCP/IP product
- Logging and data collection parameters
- Reporting parameters
- Multi-system support
- Miscellaneous parameters
- The UNIX shell SSI interface
- OSAs
- The data space
- The web interface

Messages are displayed for the following:

- Each test
- Confirmation of the successful completion of each test

If errors are found, appropriate messages are displayed. All errors are highlighted in the display for easy recognition. For help on error messages, place the cursor on the error message and press F1 (Help). These messages can help you locate the source of your error so that more specific action can be taken.

Accessing Self Test

To access the self test, enter:

- **/IPTEST** at the ===> prompt
- The **SELFTEST** command on the OCS panel

For online help on the SELFTEST command, enter **SELFTEST ?** on the OCS panel.

```
PROD------------------------ NetMaster : Command Entry -----------Line 1 of 81
Command ===>
        ===>

System PROD          Limit 1000   Wrap OFF   Edit OFF   Scroll OFF   Async ON
1---+----10---+----20---+----30---+----40---+----50---+----60---+----70---+----
SELFTEST
IPDI5200 TCP/IP Self Test on PROD, TCP type is IBM
IPDI5210 Checking Sockets Interface
IPDI5212    TCP/IP is active on host PROD
IPDI5212    Host full name is PROD.SYD.NMD.SOLV.COM
IPDI5212    Host IP address is 123.0.12.34   IP port is 2604
IPDI5212    CONNECTED TO TCPIP38 USING HPNS, NODE: PROD
IPDI5212    DSN: TCPIP.DATA.PO(DATAD1) VERSION: 3.8
IPDI5220 Checking IBM TCP/IP Management Interface
IPDI5221    Dataset SYSTCPD is OK, DSN=TCPIP.DATA.PO(DATAD1)
IPDI5221    Dataset PROFILE is OK, DSN=TCPIP.DATA.PO(PROFTR)
IPDI5221    Dataset FTP.DATA is OK, DSN=TCPIP.DATA.PO(FTPDATA)
IPDI5221    Dataset OBEYFILE is OK, DSN=AUDE0.PROD.OBEYFILE
IPDI5221    Dataset IPFILE is OK, DSN=AUDE0.PROD.IPFILE
IPDI5221    Dataset SYSPRINT is OK, DSN=AUDE0.PROD.SYSPRINT
IPDI5225    Netstat interface is MVS TCP/IP ONETSTAT CS V2R8 Jobname=TCPIP38
IPDI5225    TCP/IP was started on 30-FEB-2002 08.33.52
IPDI5212    PROD was started on 31-FEB-2002 07.00.27
```

Displaying the NetMaster for TCP/IP Initialization Log

The NetMaster for TCP/IP initialization log can be used to help diagnose problems which occurred during the following:

- Initial initialization of your product
- When you change any parameters using the administration menu options

To access the initialization log enter **/INIT.L** at the ===> prompt. The ICS : Initialization and Customization Log panel is displayed.

The location of the error messages allows you to identify the feature causing the error and to take appropriate action. Enter:

- **S** next to a message for more information about the message.
- **L** next to a message to access the activity log. Use this to view other activity in the region at the time of the error.

Troubleshooting OS/390 SNMP Problems

If you receive any of the following errors, access to SNMP data may not be enabled:

- No entries in OS/390 routing table
- No entries in interface
- Listener list detectors always trigger

To enable access to SNMP data, do this:

1. Check that the subagent has been configured in profile TCP.

2. The TCP/IP subagent must be able to connect to the SNMP agent. To do this, it uses a UNIX Stream socket represented by the file name in the dpiPathNameForUnixStream MIB object.

   To set the file name for the Unix Stream socket, do one of the following:

   - Use the dpiPathNameForUnixStream statement in the OSNMPD.DATA file
   - Use the -s OSNMPD initialization parameter

   The Unix Stream socket must be a UNIX System Services (USS) Special File. If the file exists, ensure that:

   - It has the correct file attributes
   - It has permission bits set to allow both read and write

   If the file does not exist, but the file name is specified as described above, the SNMP agent will automatically create it.

3. To determine the file name, issue the USS command:

   ```
   osnmp  get dpiPathNameForUnixStream.0
   ```

   The file name is displayed—the default file name is /tmp/dpi_socket.

4. To determine the attributes and permissions, issue the USS command:

   ```
   ls -1 /tmp/dpi_socket
   ```

   The file attributes and permissions are displayed. For example:

   ```
   crw-r-----             1 OMVSKERN SYS1       6,  0 Jan 27 03:35 dpi_socket
   ```

   The first character (c) indicates this is a Special File. The following nine characters show permissions. The values 6 and 0 preceding the date indicate the major and minor node type. The Unix Stream socket must be major node type 6.

5. To create the file use the MKNOD command located in the /usr/sbin directory. For example:

   ```
   cd /usr/sbin
   mknod /tmp/dpi_socket c 6 0
   ```

   A character special file is created; major type 6, minor type 0.

For more information see:

- IBM's *Communications Server IP Configuration* for further information about configuring the SNMP Agent and configuring community names
- IBM's *UNIX System Services Planning Manual* for information about creating Special Files
- IBM's *UNIX System Services Command Reference* for information on how to use and interpret USS commands

Commonly Encountered Errors

The following table describes common errors or conditions you might encounter, their probable cause, and the recommended action to take to resolve them:

Error Type and/or Symptom	Probable Cause	What to do...
NETSTAT or Connections List Errors:		
General note:	The majority of errors received from NETSTAT or List Connection requests are caused by incorrect or incomplete configuration.	See the *NetMaster Network Performance for TCP/IP Implementation Guide*. Ensure that all steps in the tasks for your environment have been successfully completed.
IPNS0417 NETSTAT NOT AVAILABLE ON PORT 23	The definition of the TELNET service in the TCPaccess ACPCFG*xx* member uses USSTAB, AUTO or does not enable the use of the NETSTAT application.	Follow the instructions in the section Enabling the System Command Console Interface in the *NetMaster Network Performance for TCP/IP Implementation Guide*.
IPNS0807 UNIX SHELL OPEN REQUEST HAS FAILED, CALL=0 RC=0 RSN=0000-0000	SSI software is out of date.	Ensure that the SOLVE SSI run-time libraries contain Management Services V5.0 or later.
IPNS0806 UNIX SHELL REQUEST OPEN HAS FAILED DUE TO service not available	SSI is not configured to act as the USS interface.	Update your SOLVE SSI configuration to include the start parameter UNIX=YES.
IPNS0806 UNIX SHELL REQUEST OPEN HAS FAILED DUE TO UNIX interface not available	The user ID associated with the SOLVE SSI started task does not have an OMVS segment.	Set up an OMVS segment in the RACF, CA-ACF2, or CA-Top Secret profile of the user ID associated with the SOLVE SSI started task (see the *Unicenter Mainframe Installation and Setup Instructions*).
IPNS0809 UNIX SHELL INITIALIZATION FAILED, CALL=760 RC=129 RSN=053B-006C	The user ID associated with the SOLVE SSI started task has defined in its OMVS segment the name of a shell program that could not be invoked.	Ensure that this user ID has a valid shell program specified in the PROGRAM section of its OMVS segment (see the *NetMaster Network Performance for TCP/IP Implementation Guide)*.

Error Type and/or Symptom	Probable Cause	What to do...
Cannot change to HOME directory	The user ID associated with the SOLVE SSI started task does not have at least read access to the home directory defined in its OMVS segment.	Ensure that this user ID has at least read access to the specified home directory (see the *NetMaster Network Performance for TCP/IP Implementation Guide*).
User ID not displayed in list connections or IP address not being displayed in EASINET.	The default application for Telnet is not performing user ID registration. This may be because:	
	* The Telnet default application is a region using SOLVE:Access (EASINET or MAI) with a management services level below 4.0.	* Use the EASINET logon request to pass session control to directly to a region running management services level 4.0 or higher. See the chapter, "Implementing Connection Awareness". * Upgrade the default application region to management services level 4.0 or higher.
	* SYSPARM IPCHECK may be set to NONE.	Set IPCHECK to REGISTER or VERIFY.
	* The default application using RACF is not SOLVE:Access.	See the chapter "Setting Up Connection Awareness".
Errors Starting Sockets Interface:		
N3B290 WARNING - UNABLE TO OBTAIN TCP/IP HOST NAME, USING 'LOCALHOST' N3B291 WARNING - UNABLE TO OBTAIN TCP/IP HOST ADDRESS USING 127.0.0.1 (LOOPBACK)	The DNRALC*xx* and DNRHST*xx* members have not been configured as specified. For example DNRALC*xx* does not contain an entry for the TCPaccess subsystem name.	Follow the instructions for setting up DNR in the *NetMaster Network Performance for TCP/IP Implementation Guide.*

Error Type and/or Symptom	Probable Cause	What to do...
TVG101 INTERLINK TCPAXS INTERFACE INITIALIZATION FAILURE 31 - BAD RET HOSTNAME LEN: 0 N3B220 TCPIP START (TYPE=TCPAXS) FAILED, 7 - TCPAXS MODULE NM053030 INITIALIZATION FAILURE, RC: 8	The DNRALCxx member contains an entry for the TCPaccess subsystem name, which translates the name directly to an IP address, not to a host name.	Follow the instructions for setting up DNR in the *NetMaster Network Performance for TCP/IP Implementation Guide.*
Name lookup is not working	The host defined as the DNS is down or incorrectly defined	Ping the name server host to make sure it is accessible from the system. **IBM TCP/IP:** Check the NSINTERADDR parameter specified (up to three occurrences allowable) in the TCPIP.DATA dataset. This parameter should contain the IP address of a Domain Name Server (DNS) system. **TCPaccess:** Check the contents of the DNRNSC*xx* member in the TCPaccess PARM dataset
Name lookup is not working (cont)	The host's files are incorrect or out of date.	**IBM TCP/IP:** Check the contents of the *prefix*.HOSTS.LOCAL dataset. If the *prefix*.HOSTS.LOCAL dataset has been updated since NetMaster for TCP/IP was started and you wish to refresh it, go /ICS SOCKETS parameter group.
Name lookup, ping, or traceroute is slow	The Name Server is unreachable, inoperative, or slow.	**TCPaccess:** Update the TCPaccess Sockets parameter group to specify DNR mode 'Local'.

Error Type and/or Symptom	Probable Cause	What to do...
IPSP1304 NO RESPONSE FROM &P1 - SNMP MAY NOT BE AUTHORIZED OR SUPPORTED IPSP1804 NO RESPONSE FROM &P1 - SNMP MAY NOT BE AUTHORIZED OR SUPPORTED	The device may not support SNMP or you may not be authorized.	See the chapter, "Setting Up Connection Awareness".
Mails not being sent for Email trouble ticket	The job name or destination on the trouble ticket interface definition is not running.	Check the trouble ticket job name. See the chapter "Setting Up Proactive Monitoring".
UNIX SHELL SSI Interface Self Test errors: **General note:**	The majority of errors received from the UNIX shell SSI Interface section of the NetMaster for TCP/IP self test are caused by incorrect or incomplete configuration.	See the *Unicenter NetMaster for TCP/IP Implementation Guide*. Ensure that all steps in all tasks in this section have been successfully completed.
IPDI5227 Warning: Unexpected response from NETSTAT, see log for details.	Netstat errors may indicate a variety of conditions, which may or may not affect your system.	Look in the activity log for additional error messages. These may be prefixed by IPNS or EZA.

Interpreting Socket Error Codes

For information about socket error codes, see the appendix "&SOCKET Verbs".

Appendix

A

UNIX Configuration Issues

NetMaster for TCP/IP can interact with remote, foreign network managers such as NetView for AIX and HP OpenView via a two-way TCP/IP connection. Before full use can be made of the NetMaster for TCP/IP SNMP components that interact with remote foreign network managers, the system administrator must ensure that the remote network management applications are installed and operational.

There is no code to install on your UNIX system.

Defining SNMP Managers to NetMaster for TCP/IP

NetMaster for TCP/IP can be used to provide access to SNMP MIB data and to perform TCP/IP management commands (such as PING and TRACEROUTE) by using an SNMP management platform such as OpenView. This functionality is available on the SNMP : Function Menu.

The facilities on the NetMaster for TCP/IP SNMP : Function Menu allows SNMP commands and TCP/IP management commands to be routed to supported SNMP network management stations.

A supported SNMP network management station (or SNMP manager) can be one of the following:

- TCP/IP on MVS
- A remote SNMP manager that operates under UNIX

SNMP Manager Support

Before you can use any of the functions on the SNMP : Function Menu with a UNIX manager, you need to define it to NetMaster for TCP/IP. You do this by using the Manager Administration List.

1. From the Primary Menu, enter **/IPDIAG.SF.A**. The SNMP: Defined Managers List is displayed.
2. Press F4 (Add). The SNMP: Manager Definition panel is displayed.
3. Complete the fields on the panel. Pres F1 (Help) for information on the fields.
4. Press F3 (File) to save the definition.

Prerequisites - NetView for AIX

The following is required for NetMaster for TCP/IP:

- NetView for AIX Version 3.2 or later
- TCP/IP communications link to the host

Prerequisites - OpenView

The following is required for NetMaster for TCP/IP:

- HP-UX Version 9 or later
- H-P OpenView Version 3.3 or later
- TCP/IP communications link to the host

Recording an SNMP Manager Definition

From the SNMP : Manager Definition panel, you can record information for the following kinds of managers:

- TCP/IP for MVS managers
- UNIX-based SNMP managers that are linked to the mainframe by a REXEC connection

```
PROD-------------------- SNMP: Manager Definition ---------------$SPSMDISPLAY
Command ===>                                                   Function=Add

Manager Name ............... ________
        Description ........ ________________________________
        Type .............+ ________________
             Description ...
        Category ...........

For Communications using REXEC
     IP Address ............ ________________
     Port Number ........... 512_____
     User ID ............... ____________________________________________________
     Password .............
     Password (Confirm) ....

Command Paths
     SNMP .................. /usr/OV/bin_________________________________________
     OVTopodump ............ /usr/OV/bin_________________________________________
     Ping .................. /etc________________________________________________
     Traceroute ............ /usr/local/etc______________________________________
     Findroute ............. /usr/OV/bin_________________________________________

 F1=Help      F2=Split     F3=File      F4=Save
                           F9=Swap                                  F12=Cancel
```

Definitions of the fields on the SNMP : Manager Definition panel are:

Manager Name—the name you want to assign to the particular manager you are adding.

Description—a brief description of the particular manager you are adding, such as:

`Network Control RS6000.`

Type—a code identifying the type of SNMP manager. The following values are supported:

H-P/OV	H-P Open view
MVS_TCP/IP_A	TCP/IP for MVS
NV/AIX	NetView for AIX V3.2 or above

Description—a brief description of the manager type, including the type and version number of the SNMP manager. When you enter a valid value in the Manager Type field, this description is supplied automatically. It cannot be overwritten.

Category—how the manager is contacted:

- Local (native)
- REXEC

For Communications Using REXEC

If you are using the REXEC interface to communicate with this manager, you need to complete these fields:

IP Address—the IP address of this manager.

Port Number—the port number of the REXEC protocol for this manager. The default is 512.

User ID—the user ID (up to 16 characters) which will issue commands on the UNIX system.

Note: This user ID needs to be defined on the UNIX system with authority to connect via REXEC and to issue the commands listed below

Password—the password (up to 16 characters) of the UNIX user ID.

Password (Confirm)—a field to confirm the password.

Command Paths

The command paths tell NetMaster for TCP/IP where various commands can be found on the UNIX system.

The command paths needed for a REXEC connection are:

- snmpwalk, snmpget, snmpset, snmpnext
- ovtopodump
- ping
- traceroute
- findroute

Copying a Similar Manager Definition

To add a manager definition to the NetMaster for TCP/IP configuration file that is similar to an existing manager definition, you can copy the existing record and change field values as appropriate. The procedure for copying an existing record is documented in Maintaining SNMP Manager Definitions

Maintaining SNMP Manager Definitions

To browse, update, copy, or delete an SNMP manager definition (assuming that you have the authority), you first need to list all recorded managers. The procedure is as follows:

1. Select option **A** - Manager Administration List from the SNMP : Function Menu. The SNMP : Defined Managers List panel is displayed with a list of all recorded SNMP managers.
2. Press F8 (Right) to see further manager details.
3. Find the manager you require and apply one of the available actions to that listed item:
   - **S**, **/**, or **B** to select or browse the manager definition
   - **C** to copy the manager definition
   - **D** to delete the manager definition
   - **U** to update the manager definition

Appendix

B Enhanced Automation

This appendix provides details of various programming procedures that you can use within NetMaster for TCP/IP. These procedures provide you with enhanced automation of your TCP/IP network management.

This appendix contains the following information:

- $IPCALL and $TNCALL Procedures
- Setting Up Application Connection Names
- Reacting to IP Node Monitor State Changes
- Trapping FTP, Telnet, Connection, and Message Events

$IPCALL and $TNCALL Procedures

The $IPCALL and $TNCALL programming interfaces can be used by NCL procedures to invoke the facilities of NetMaster for TCP/IP.

$IPCALL Procedure

$IPCALL is a programming interface that can be used to query information using NetMaster for TCP/IP.

Some NetMaster for TCP/IP commands, such as TELNET, are implemented by command equates to $IPCALL. These commands are described in the *Command Reference*.

This appendix describes the following $IPCALL functions, which are useful for programming:

- Using $IPCALL to obtain device and links information about the local TCP/IP stack
- Using $IPCALL to obtain system and interface information from a host that supports MIB-II
- Using $IPCALL to control the receipt of messages issued by TCPaccess

$TNCALL Procedure

$TNCALL provides access to the Telnet client functions that can be used to start, stop, and send data on a Telnet connection.

The Telnet functions are described in the *NetMaster for TCP/IP User's Guide*, and the syntax is described in the *Command Reference*.

Example

The TELNET command is actually a command equate to $TNCALL COMMAND=TELNET.

$IPCALL ATTR=DEVLINKS

Function

Obtains information about devices and links on the local TCP/IP stack.

```
&CONTROL SHRVARS=$IPDATA
  - EXEC $IPCALL ACTION=GET  +
  CLASS=TCPIPMVS  +
  ATTR=DEVLINKS  +
  [SYSTEM=linkname]
```

Use

To obtain the device and links information as shown on the device links display for a stack resource. The data is returned in a Mapped Data Object (MDO) named $IPDATA mapped by $IPMPDEV.

Operands

ACTION=GET CLASS=TCPIPMVS ATTR=DEVLINKS

Requests the device links information.

SYSTEM=*linkname*

Specifies the NetMaster for TCP/IP link from which you want to obtain information. Leave this blank for the local system.

Examples

```
&CONTROL SHRVARS=$IPDATA
-EXEC $IPCALL ACTION=GET CLASS=TCPIPMVS ATTR=DEVLINKS
&IF &RETCODE NE 0 &THEN +
   &DO
      &WRITE &SYSMSG
      &END
   &DOEND
&ASSIGN VARS=NAME* FROM MDO=$IPDATA.DEVICE.{*}.NAME
&MAXDEV = &ZVARCNT
&ASSIGN VARS=STATUS* FROM MDO=$IPDATA.DEVICE.{*}.STATUS
&I = 1
&DOWHILE &I LE &MAXDEV
   &WRITE DATA=NAME=&NAME&I STATUS=&STATUS&I
   &I = &I + 1
&DOEND
```

Notes

You can view the structure of the $IPMPDEV map from the Mapping Services menu to obtain more information about fields available in the map.

$IPCALL ATTR=SYSINFO

Function

Obtains system information about an IP host.

```
&CONTROL SHRVARS=$IPDATA
   - EXEC $IPCALL ACTION=GET +
   CLASS=IPHOST +
   ATTR=SYSINFO +
   HOST=hostaddr [COMMNAME=snmpcommunityname] +
   [SYSTEM=linkname]
```

Use

Obtains information about an IP device such as a router by querying MIB variables in the device. This is the same information as displayed on the system information display on the network diagnosis menu.

The information available includes the device type, up time, and contact and location details. For Cisco routers, it also includes the percentage of CPU usage. The data is returned in an MDO named $IPDATA mapped by $IPMPSYS.

Operands

ACTION=GET CLASS=IPHOST ATTR=SYSINFO

Requests the system information.

HOST=*hostaddr*

Specifies the IP address or host name of the device.

COMMNAME=*snmpcommunityname*

Is the SNMP community name of the device.

Note: As community names are case-sensitive, your NCL should run &CONTROL NOUCASE. The community name default is supplied by the SNMP community name utility.

SYSTEM=*linkname*

Lets you nominate the NetMaster for TCP/IP system that you want to issue the SNMP GET request.

Examples

```
&CONTROL NOUCASE
&CONTROL SHRVARS=$IPDATA
EXEC $IPCALL ACTION=GET CLASS=IPHOST ATTR=SYSINFO +
              HOST=199.0.128.1 SYSTEM= COMMNAME=public
&IF &RETCODE NE 0 &THEN +
   &DO
      &WRITE &SYSMSG
      &END
   &DOEND
&ASSIGN VARS=INF* FROM MDO=$IPDATA.* GENERIC
&WRITE DATA=&INFHOSTNAME(&INFADDRESS) is a &INFOBJECTID +
            CPU: &INFCPBUSYPC
```

Notes

You can view the structure of the $IPMPSYS map from the Mapping Services menu to obtain more information about fields available in the map.

$IPCALL ATTR=IFS

Function

Obtains interface information from a device.

```
&CONTROL SHRVARS=$IPDATA
  - EXEC $IPCALL ACTION=GET +
  CLASS=IPHOST +
  ATTR=IFS +
  HOST=hostaddr [COMMNAME=snmpcommunityname] +
  [SYSTEM=linkname]
```

Use

To obtain interface information from an IP host such as a router by querying MIB variables in the device. This is the same information as displayed on the interface display on the network diagnosis menu.

The information available is a list of interfaces, including interface name, type, desired status, and actual status. The data is returned in an MDO named $IPDATA mapped by $IPMPIFS.

Operands

ACTION=GET CLASS=IPHOST ATTR=IFS

Requests the interface information.

HOST=*hostaddr*

Specifies the IP address or host name of the device.

COMMNAME=*snmpcommunityname*

The SNMP community name of the device.

Note: As community names are case sensitive, your NCL should run &CONTROL NOUCASE. The community name default is supplied by the SNMP community name utility.

SYSTEM=*linkname*

Lets you nominate the NetMaster for TCP/IP system you want to issue the SNMP GET request.

Examples

```
&CONTROL NOUCASE
&CONTROL SHRVARS=$IPDATA
EXEC $IPCALL ACTION=GET CLASS=IPHOST ATTR=IFS +
             HOST=199.0.128.1 SYSTEM= COMMNAME=public
&IF &RETCODE NE 0 &THEN +
   &DO
      &WRITE &SYSMSG
      &END
   &DOEND
&ASSIGN VARS=NAME* FROM MDO=$IPDATA.IFS.{*}.DESCR
&MAXDEV = &ZVARCNT
&ASSIGN VARS=IFTYPE* FROM MDO=$IPDATA.IFS.{*}.TYPE
&ASSIGN VARS=DSTAT* FROM MDO=$IPDATA.IFS.{*}.ADMSTAT
&ASSIGN VARS=ASTAT* FROM MDO=$IPDATA.IFS.{*}.OPERSTAT
&I = 1
&DOWHILE &i LE &MAX
   &WRITE DATA=&NAME&i TYPE=&IFTYPE&I +
          Desired state=&dstat&i actual=&Astat&i
   &I = &I + 1
&DOEND
```

Notes

You can view the structure of the $IPMPIFS map from the Mapping Services menu to obtain more information about fields available in the map.

Setting Up Application Connection Names

NetMaster for TCP/IP assigns an application name to the connection start and connection end events it receives from the TCP/IP stack. The application name makes it easier to recognize and summarize events in reports. The application name is derived from the:

- Local port number
- Remote port number
- OS/390 job name involved in the connection

If the application names automatically assigned by NetMaster for TCP/IP are not suitable you can change them with the application name mapping facility. This facility controls how connections are associated with an application.

There are three tables you can set up to recognize a connection:

- Job name
- Local port
- Remote port

You can specify the order the tables are searched to match the connection. Blank lines and lines beginning with a hash (#) are ignored in the tables.

To set up the connection matching facility, do this:

1. If you want the job name to identify connections create a job name table by adding a member to the TESTEXEC library. For example a member called JOBTAB may contain:

```
# Comments start with '#'
# Each line is a job name followed by the app name
# The job FTPSRV32 is 'FTP'
FTPSRV32 FTP
# Jobs starting with CICS are 'CICS'
CICS* CICS
# Jobs starting with PR use the job name as the APPL name
PR*  *
```

2. Set up the port name tables. A default name table with standard port numbers is provided as the member $IPPR00T in the *hlq*.IP600.IPTEXEC library. If you want to modify this, copy it to TESTEXEC first. You can use:
   - The same table for local and remote ports
   - Individual tables for the local and remote ports
   - Determine the order the tables are searched in. For example, if you set up a job name table your search order may be JOBNAME, LCLPORT, and then RMTPORT.
3. Determine a default application name to use if no matches are found.

4. Register the table names and search order by creating and running an NCL procedure that contains a call to the registration process. Specify the parameters as follows:

Application Connection Parameters

Parameter	Value	Meaning
1	$IPPKG, RegisterAppMap	Request registration
2	(*first*,*second*,*third*)	Set the order to check the tables
3	*Jobtable*	Name of the member containing the job table
4	Lclporttable	Name of the member containing the local port table
5	Rmtporttable	Name of the member containing the remote port table
6	Default	The default application name if not found in the tables

```
For example:
&CALL PROC=$CAPKCAL PARMS=($IPPKG, RegisterAppMap, +
       (JOBNAME,LCLPORT,RMTPORT), -* set order
        JOBTAB,
        $IPPR00T,
        $IPPR00T,
        UNKNOWN)
&WRITE &SYSMSG
```

5. Test your system to ensure that the correct application names are being used.
6. Make the registration permanent by starting it from the READY procedure at system initialization.

Reacting to IP Node Monitor State Changes

A change in the state of an IP node monitored by the IP node monitor is advertised by an Event Distribution Services (EDS) event, which triggers a display update for anyone watching the device status display. The EDS event also can be picked up by any other process which requires this information.

The EDS event created for a state change has the following attributes:

Name

$IPNMON.STATE.UPD

Resource

The IP address of the node

Ref

oldstate - newstate

Possible status values are:

Unknown

Ping has not completed.

OK

Ping completed successfully.

Timeout

Ping timed out.

SNMP Error

Ping completed successfully but an SNMP request has returned an error.

Trapping FTP, Telnet, Connection, and Message Events

NetMaster for TCP/IP allows you to use the proactive monitor to trap:

- FTP, Telnet, and connection message events from SMF exits
- Console and syslog messages from TCPaccess

See the chapter, "Setting Up Connection Awareness", for further information about using SMF exits and TCPaccess messages.

See the chapter, "Setting Up Proactive Monitoring", for further information about the proactive monitor.

To trap FTP failures, use an FTPFAIL monitor. To trap other EDS events, use a CUSTOM monitor.

Sample Code

Shown below is sample NCL code to receive and process FTP and Telnet events.

1. Issue the PROFILE EDS command to register an interest in selected EDS events:

   ENABLE provides a profile name. Choose a value which is meaningful to you.

   NAME provides the (full or generic) event names that you wish to receive. See the following table for a list of event names and their meanings.

   ```
   PROFILE EDS ENABLE=xxxxxxxx   +        -*  Choose a profile name.
               NAME=$IP.xxxxxxxx           -*  Choose event name/prefix
   ```

2. Wait for each event to arrive.

   The event includes an MDO, mapped by the $NCL map, which contains various variables according to the specific event. For example, FTP events include a LASTREPL field that shows the last FTP reply code.

   ```
   &INTREAD MDO=IPMDO.       -*wait for first event
   &DOWHILE &ZFDBK = 0
     &ASSIGN MDO=IPMDO. MAP=$NCL
     &ASSIGN VARS=$IP$* GENERIC FROM MDO=IPMDO.+
          VARS=xxxx* -*is any variable prefix
                     -*max 4 characters
     &WRITE DATA=Message=&$IP$MSGTEXT
     &WRITE DATA=Local IP Address=&$IP$LCLADDR
     &WRITE DATA=Remote IP Address=&$IP$RMTADDR
     &WRITE DATA=Last Reply=&$IP$LASTREPL
     &INTREAD MDO=IPMDO.  -*wait for next event
     &DOEND
   ```

The EDS events that are issued when an FTP, Telnet, or connection event takes place are listed in this table.

FTP, Telnet, Connection, and Message EDS Events

Event	Event Name	Object	Resource	Reference	Message ID	Client/ Server
FTP Retrieve	$IP.FTPLOG.RETR	Remote IP address	Dataset name	FTP server name	IPFM2103 IPCM2303	S C
FTP Store	$IP.FTPLOG.STOR	Remote IP address	Dataset name	FTP server name	IPFM2103 IPCM2303	S C
FTP Store Unique	$IP.FTPLOG.STOU	Remote IP address	Dataset name	FTP server name	IPFM2103 IPCM2303	S C
FTP Append	$IP.FTPLOG.APPE	Remote IP address	Dataset name	FTP server name	IPFM2103 IPCM2303	S C
FTP Logon Failed	$IP.FTPLOG.LOGONF	Remote IP address		FTP server name	IPFM2102	S
FTP Delete	$IP.FTPLOG.DELETE	Remote IP address	Dataset name	FTP server name	IPFM2104	S
FTP Rename	$IP.FTPLOG.RENAME	Remote IP address		FTP server name	IPFM2105	S
FTP Retrieve Failure	$IP.FTPFAIL.RETR	Remote IP address	Dataset name	FTP server name	IPFM2113 IPCM2309	S C
FTP Store Failure	$IP.FTPFAIL.STOR	Remote IP address	Dataset name	FTP server name	IPFM2113 IPCM2309	S C
FTP Store Failure Unique	$IP.FTPFAIL.STOU	Remote IP address	Dataset name	FTP server name	IPFM2113 IPCM2309	S C
FTP Append Failure	$IP.FTPFAIL.APPE	Remote IP address	Dataset name	FTFTP server name	IPFM2113 IPCM2309	S C

Event	Event Name	Object	Resource	Reference	Message ID	Client/ Server
FTP Delete Failure	$IP.FTPFAIL.DELETE	Remote IP address	Dataset name	FTP server name	IPFM2114	S
FTP Rename Failure	$IP.FTPFAIL.RENAME	Remote IP address	Old dataset name, new dataset name	FTP server name	IPFM2115	S
Telnet connection started	$IP.TNLOG.START	Remote IP address	Telnet LU name	Application name	IPCM2002	S
Telnet connection started	$IP.TNLOG.START	Remote IP address	Telnet client job name	Telnet client node name	IPCM2313	C
Telnet connection stopped	$IP.TNLOG.STOP	Remote IP address	Telnet LU name	Application name	IPCM2003	S
Telnet connection stopped	$IP.TNLOG.STOP	Remote IP address	Telnet job name	Telnet client node name	IPCM2314	S
Connection started	$IP.CONNECT.START	Remote IP address	Client job name	Client job name	IPCM2311	-
Connection stopped	$IP.CONNECT.STOP	Remote IP address	Client job name	Client job name	IPCM2312	-
TCPaccess message received	$IP.AXSLOG.MESSAGE	Severity code	TCPaccess SSID	TCPaccess message number	TCPaccess message number	

References

For further information about the format of FTP records, and about how SMF exits are invoked, see IBM's *TCP/IP for MVS Customization and Administration Guide* or *Communications Server IP Configuration Guide*.

For further information about FTP reply codes, see *RFC 959*, *File Transfer Protocol*.

Trappable Events

Event	Event Name	Object	Resource	Reference	MDO
IPTREND dataset created	$IP.REPORTING.IPTREND	Dataset	IPTREND	Created	Mapped by $NCL with variables: DSN—Name of IPTREND dataset RECCOUNT—Number of records written to IPTREN
IPDETAIL dataset created	$IP.REPORTING.IPDETAIL	Dataset	IPDETAIL	Created	Mapped by $NCL with variables: DSN—Name of IPDETAIL dataset RECCOUNT—Number of records written to IPDETAIL

Appendix

C

&SOCKET Verbs

This appendix describes the &SOCKET verb set which allows NCL processes to use the Communications Server and TCPaccess sockets interfaces.

It provides you with the following information:

- About the Socket Interfaces
- &SOCKET
- System Variables
- Sample Code for TCP and UDP &SOCKET Verbs
- TCP/IP Vendor Interface Restrictions and Limitations
- Interpreting Socket Error Codes

About the Socket Interfaces

A socket is an end point for inter-process communication over a network running TCP/IP. The OS/390 and TCPaccess socket interfaces support a number of underlying transport mechanisms. Sockets can simultaneously transmit and receive data from another process, using methods that depend on the type of socket being used. Sockets can be of the following types, each representing a different type of communications service:

- TCP sockets
- UDP sockets

TCP Sockets

TCP (Transmission Control Protocol) sockets provide reliable, connection-based communications. In the case of a sockets interface, the two processes must establish a logical connection with each other. The data is a stream of bytes that is sent without errors or duplication, and is received in the same order in which it was sent.

The following sections describe the various types of stream socket applications and the &SOCKET verbs you would use for each type.

NCL Verb Set for a Server

&SOCKET REGISTER	to register the server
&SOCKET ACCEPT	to accept connection from clients
&SOCKET TRANSFER_REQUEST	to transfer a socket ID from one NCL process to another
&SOCKET TRANSFER_ACCEPT	to accept a socket ID from a donor NCL process
&SOCKET SEND	to send data to clients
&SOCKET RECEIVE	to receive data from clients
&SOCKET CLOSE	to close the client connection

NCL Verb Set for a Client

&SOCKET CONNECT	to connect to the server
&SOCKET SEND	to send data to the server
&SOCKET RECEIVE	to receive data from the server
&SOCKET CLOSE	to close the server connection

UDP Sockets

UDP (User Datagram Protocol) sockets communicate by way of discrete messages called datagrams, which are sent as packets. UDP sockets are connectionless. Communication processes do not have a logical connection with each other and therefore the delivery of their data is unreliable. The datagrams can be lost or duplicated, or they might not arrive in the same order in which they were sent.

The following sections describe the UDP socket application and the &SOCKET verbs you would use.

NCL Verb Set for UDP Sockets

&SOCKET OPEN	to open the communication socket and port
&SOCKET SEND_TO	to send datagrams
&SOCKET RECEIVE_FROM	to receive datagrams
&SOCKET CLOSE	to close the communication socket

NCL Verb Set for PING and TRACEROUTE

The PING and TRACEROUTE verbs directly access the lower layer protocols such as Internet Protocol (IP) and Internet Control Message Protocol (ICMP).

&SOCKET PING	to ping the host
&SOCKET TRACEROUTE	to trace the route to the host

Socket Built-in Functions

NetMaster for TCP/IP contains built-in functions which you can use to obtain information about socket processes. The types of information available are:

- If a function is supported
- Information about the local host
- Information about a specific socket

Determining if a Function Is Supported

You can use the following built-in function to determine if a function is supported by the current vendor stack:

&ZTCPSUPP *function*

Returns NO if the *function* is unknown or not supported, or if the TCPIP START command has never been issued.

Returns YES if the *function* is supported.

You can use this built-in function for the following functions:

- PING
- TRACEROUTE
- GETHOSTBYADDR
- GETHOSTBYNAME

Obtaining Information About the Local Host

You can use the following built-in function to obtain information about the local host or vendor stack:

&ZTCPINFO ***infotype***

The data returned depends on the value of *infotype*, as shown in the following table:

TYPE	Returns the vendor stack type as specified on TCPIP START command TYPE=
STATUS	Returns ACTIVE, STARTING, STOPPING, QUIESCING, or INACTIVE.
HOSTADDR	Returns the local host IP address.
HOSTNAME	Returns the local host alias name.
HOSTFULLNAME	Returns the local host full name.
INTERFACE	Returns descriptive interface information returned by the vendor interface.
CONNECTION	Returns connection information from the vendor interface.

Obtaining Information About a Specific Socket

Use the following built-in function to obtain information about a specific socket owned by the process:

&ZSOCINFO ***socket_id infotype***

socket_id is used to identify the socket.

The data returned depends on the value of *infotype*, as shown in the following table (default **EXISTS**):

EXISTS	Returns YES or NO to indicate whether the socket exists.
TYPE	Returns a character string representing the type of socket: TCP, TCPLISTEN, UDP.
PORT	Returns the port number assigned to the socket by the local host.
ADDR	Returns the IP address assigned to the socket by the local host.
PEERPORT	Returns the port number of the peer host on a TCP connection. For a UDP socket this is the port number last referenced by a SEND_TO or RECEIVE_FROM verb.
PEERADDR	Returns an IP address of the peer host on a TCP connection. For a UDP socket this is the IP address last referenced by a SEND_TO or RECEIVE_FROM verb.
PEERNAME	Returns the name of the peer host on a TCP connection (where HOSTNAME was used to establish the connection).
RETCODE	Returns the return code from the last operation on this socket.
FDBK	Returns the reason code from the last operation on this socket.
VERRIN	Returns the vendor error information from the last operation on this socket.
ERRNO	Returns the error number value from the last operation on this socket.
BYTESIN	Returns the number of bytes of data received by this socket.
BYTESOUT	Returns the number of bytes of data sent by this socket.

Obtaining Information About Error Codes

Use the following built-in functions to obtain information about error codes:

&ZTCPERNM #

Returns the logical name of a TCP/IP error code; for example, &A = &ZTCPERNM 40 would set &A to the value ENETUNREACH.

If the number is not a recognized error code, &A is set to EUNKNOWN.

&ZTCPERDS #

Returns a short message for a TCP/IP error code; for example, &B = &ZTCPERDS 40 would set &B to the value:

N3AD01 40 - ENETUNREACH - DESTINATION NETWORK UNREACHABLE

If the number is not a recognized error code, &B is set to:

N3AD01 - *** - EUNKNOWN - UNKNOWN ERRNO

(where *** is the error number supplied).

&SOCKET

Function

The &SOCKET verbs provide NCL control over the allocation and management of communications using TCP/IP. The syntax is shown below. For further information about each of the &SOCKET verbs, see the *Network Control Language Reference*.

```
&SOCKET                 ACCEPT
                        ID=socket_id
                        [ TYPE={ SYNC | ASYNC } ]

&SOCKET                 CLOSE
                        ID=socket_id

&SOCKET                 CONNECT
                        {ADDRESS=ip_address | HOSTNAME=host_name }
                        PORT=port_id
                        [ WAIT=time | 30 ]
                        [ TYPE={ SYNC | ASYNC } ]

&SOCKET                 GETHOSTBYADDR
                        ADDRESS=ip_address
                        [ MDO=mdo_name ]
                        [ WAIT=time ]
                        [ TYPE={ SYNC | ASYNC } ]

&SOCKET                 GETHOSTBYNAME
                        HOSTNAME=host_name
                        [ MDO=mdo_name ]
                        [ WAIT=time ]
                        [ TYPE={ SYNC | ASYNC } ]

&SOCKET                 OPEN
                        [ PORT=port_id ]

&SOCKET                 PING
                        { ADDRESS=ip_address | HOSTNAME=host_name }
                        [ PACKETSIZE=nn ]
                        [ COUNT=nn ]
                        [ MDO=mdo_name ]
                        [ GETNAME={ YES | NO } ]
                        [ WAIT=time ]
                        [ TYPE={ SYNC | ASYNC } ]

&SOCKET                 RECEIVE
                        ID=socket_id
                        { MDO=mdo_name |
                        VARS={ name | (name,name,... name) [SEGMENT] } |
                        { ARGS | VARS=prefix* } [ RANGE=(start,end) ] [SEGMENT] }
                        [ LENGTH=0..4 ]
                        [ WAIT=time ]
                        [ TYPE={ SYNC | ASYNC } ]
```

```
&SOCKET          RECEIVE_FROM
                 ID=socket_id
                 { MDO=mdo_name |
                 VARS={ name | (name,name,... name) [SEGMENT] } |
                 { ARGS | VARS=prefix* } [ RANGE=(start,end) ] [SEGMENT] }
                 [ WAIT=time ]
                 [ TYPE={ SYNC | ASYNC } ]

&SOCKET          REGISTER
                 PORT=port_id
                 [ CONVLIM=nn ]

&SOCKET          SEND
                 ID=socket_id
                 { MDO=mdo_name | VARS=prefix* [ RANGE=(start,end) ] |
                 VARS={ name | name,name,... name) } |
                 ARGS [ RANGE=(start,end) ] | DATA=data }
                 [ LENGTH=0..4 ]

&SOCKET          SEND_TO
                 ID=socket_id
                 { MDO=mdo_name | VARS=prefix* [ RANGE=(start,end) ]
                 VARS={ name | name,name,... name) } |
                 ARGS [ RANGE=(start,end) ] | DATA=data }
                 { ADDRESS=ip_address |  HOSTNAME=host_name }
                 PORT=port_id

&SOCKET          TRACEROUTE
                 { ADDRESS=ip_address | HOSTNAME=host_name }
                 [ PACKETSIZE=nn | 64 ]
                 [ COUNT=nn | 3 ]
                 [ HOPS=nn | 10 ]
                 [ FROMHOP=nn | 1 ]
                 [ WAIT=time | 3 ]
                 [ TYPE={ SYNC | ASYNC } ]
                 [ GETNAME={ YES | NO } ]
                 MDO=mdo_name

&SOCKET          TRANSFER_ACCEPT
                 ID=socket_id

&SOCKET          TRANSFER_REQUEST
                 ID=socket_id
                 NCLID=ncl_id
```

System Variables

The following pages describe the system variables available within the NetMaster for TCP/IP &SOCKET verb set.

&ZTCP	Indicates the status of the TCP/IP API
&ZTCPHSTA	Contains the value of the local host's IP address
&ZTCPHSTF	Contains the value of the local host's full name
&ZTCPHSTN	Contains the value of the local host's short name
&ZSOCID	Contains the socket ID of the last referenced socket
&ZSOCHNM	Contains the host name of the host referenced by some requests, such as &SOCKET GETHOSTBYADDR
&ZSOCFHNM	Contains the full host name of the host referenced by some requests, such as &SOCKET GETHOSTBYADDR
&ZSOCHADR	Contains the IP address of the host referenced by some requests, such as &SOCKET GETHOSTBYNAME
&ZSOCCID	Contains the socket ID used by the interface; for example, for IBM TCP/IP the internal socket number (a small number). Can be used to identify an NCL socket or display produced by TCP/IP; for example, the NETSTAT command.
&ZSOCERRN	Contains the last ERRNO value returned from an &SOCKET request
&ZSOCPRT	Contains the port number of the last referenced socket
&ZSOCTYPE	Indicates the socket type of the last referenced socket
&ZSOCVERR	Returns vendor error information—format is specific to the TCP/IP interface in use

For further information about system variables, see the *Network Control Language Reference*.

Sample Code for TCP and UDP &SOCKET Verbs

Sample code for TCP and UDP &SOCKET verbs is available online. It is located in the IPTEXEC file under the names given for the examples below.

Examples of Using TCP &SOCKET Verbs

Following are examples of using TCP &SOCKET verbs.

$IPSATC1—TCP Socket Server

This is a sample server procedure that accepts connections and transfers those connections to new NCL processes that are started by the server to service the connections.

To invoke the server, all that you need to do is specify a port number.

Synopsis

```
$IPSATC1 PORT=port_number
```

This procedure starts the procedure $IPSATC2 as a new process to service each new connection.

This sample also demonstrates the use of an asynchronous &SOCKET verb.

This sample works in conjunction with the following sample procedures:

- $IPSATC2 – Command Processor
- $IPSATC3 – Command Client

$IPSATC2 – Command Processor

This sample procedure is started by $IPSATC1 (the server) and has a connection transferred to it. This procedure accepts the connection, receives a command, and issues the command. It then reads the responses from the command and sends them back to the requester. The requests and responses are received and sent in ASCII to demonstrate how to do ASCII/EBCDIC translation.

Any errors that this procedure encounters are written to the log.

This sample works in conjunction with the following sample procedures:

- $IPSATC1 – TCP Socket Server
- $IPSATC3 – Command Client

You can also use Telnet instead of $IPSATC3 as the client program.

$IPSATC3 – Command Client

This is a sample client procedure that sends a command and receives responses using the &SOCKET verb. It works in conjunction with the following sample server application procedures:

- $IPSATC1
- $IPSATC2

This procedure demonstrates ASCII/EBCDIC conversion, because all data sent and received is in ASCII.

Synopsis

```
$IPSATC3 ipAddr=
hostName=
port=
command="command to be executed"
```

Note: The command must be in quotes.

You can use a Telnet client instead of this program to send commands to the server (procedures $IPSATC1 and $IPSATC2).

$IPSATC4 – SMTP Client

This is a sample SMTP client procedure that sends mail to a user. The contents of the mail are hard coded in this procedure, but the procedure could easily be modified so that the text to be sent is passed to it.

Synopsis

```
$IPSATC4 sender=userid@company.com
 recipient=userid@company.com
 smtpSvr=smtp_server_name_or_addr
```

It is suggested that you read RFC821 for an understanding of the SMTP protocol. This RFC is available on the following Web address:

http://www.ietf.org

Example of Using UDP &SOCKET Verbs

Following is an example of using UDP &SOCKET verbs. The sample shows how to send an enterprise-specific trap to HP OpenView.

$IPSAUD1 – Usage

```
$IPSAUD1 IPADDR=xxx.xxx.xxx TEXT="This is a test trap"
$IPSAUD1 HOSTNAME=a.b.c      TEXT="This is a test trap"
```

You need to change Object Identifier values to suit your requirements.

For an understanding of the SNMP protocol, read RFC1157. This RFC is available on the following Web address:

http://www.ietf.org

TCP/IP Vendor Interface Restrictions and Limitations

This section describes the restrictions and limitations of each TCP/IP vendor interface. It covers the following vendor interfaces:

- Communications Server
- TCPaccess

Vendor interface restrictions and limitations are particularly relevant to NCL &SOCKET programming.

Communications Server

The system interfaces to IBM's Communications Server using the TCP/IP macro-level interface, which uses the HPNS (High Performance Native Sockets) facility.

This interface has the following restriction and limitation: To use the PING and TRACEROUTE functions, the region RACF user ID must be in the TCP/IP OBEY list. On Communications Server systems the user ID must be set to superuser.

TCPaccess

The TCPaccess interface uses the assembler macro TLI interface to connect to TCPaccess.

This interface has the following restrictions and limitations:

- The maximum size of a UDP datagram that can be sent or received is limited by the TCPaccess configuration parameters. As distributed, this is 9000 bytes.
- By default, the system uses global DNS, except for obtaining the local host name, when the system requests local DNS.

Interpreting Socket Error Codes

The error codes that relate to problems encountered when attempting to use the &SOCKET verbs are:

- Feedback codes (&ZFDBK)
- Socket error codes (&ZSOCERRN)
- Vendor-specific codes (&ZSOCVERR)

For information about feedback codes and socket error codes, see the *Network Control Language Reference*.

Vendor-specific errors have an error number of 999 and an additional VERRIN (vendor-specific error) code. The interpretation of this error code is different depending on the vendor of the TCP/IP software. These error codes appear in messages and in the NCL system variable &ZSOCVERR.

Interpreting IBM Systems Error Codes

The &ZSOCVERR (VERRIN) system variable contains the IBM TCP/IP socket ERRNO value. This is translated into the &ZSOCERRN value documented in the *Network Control Language Reference*.

The IBM TCP/IP socket ERRNO value is displayed as a decimal number. For the meaning of the value see *IBM Communication Server IP and SNA Codes* (SC31-8571).

Interpreting TCPaccess Systems Error Codes

For TCPaccess, the &ZSOCVERR (VERRIN) system variable is in one of two formats:

- Format A: 04/*aa-bb-cccc*
- Format B: *nn-xxxx*

Decoding Format A

The A-format 04/*aa-bb-cccc* is decoded as follows:

04

A TPL function received a return code of 4

aa

Recovery action code (that is, the value of TPLACTCD in hex notation)

bb

Specific error code (that is, the value of TPLERRCD in hex notation)

cccc

Diagnostic code (that is, the value of TPLDGNCD in hex notation)

For further information about these codes, see the *TCPaccess Messages and Codes Volume II (Unprefixed Messages and Codes)* manual and look up the codes as follows:

1. Locate the *API Return Codes* chapter (page numbers are prefixed API/RET-).
2. Among the RTNCD *aabb* page titles, locate the first two sets of digits (*aa-bb*).
3. For each of these titles, locate the table that contains the various diagnostic code values (*cccc*).

Decoding Format B

The B-format *nn-xxxx* is decoded as follows:

nn

General return code (that is, the R15 value from a TPL function request)

xxxx

Diagnostic code (that is, the low half of R0 from the TPL function request, hex expanded)

For further information about these codes, see the *TCPaccess Messages and Codes* manual, and look up the codes as follows:

1. Locate the *API Return Codes* chapter (page numbers are prefixed API/RET-).
2. Locate the *TPL-Based General Return Codes* section just after the AOPEN/ACLOSE Error Codes section.
3. For each return code value (*nn*), locate the table that contains the diagnostic code values (*xxxx*).

Appendix

D SMF Record Structure

This appendix describes the structure of the SMF user records optionally created by NetMaster for TCP/IP.

General SMF Record Format

SMF Record Format

Position	Length in Bytes	Description
1 to 18	18	SMF record header
19 and 20	2	Net Master category ID Always X'5000'
21 to 32	12 blank padded	NMID Example: NETMASTR*bbbb* (where *b* is a blank)
32 to 34	2	Record type DS or DH where DS is a one-off sample record and DH is the hourly roll up
35 and 36	2	Length of data Example: X'0007' indicates that the data is 7 bytes long
37 and 38	2	Field identifier (see table below)
39 to *n*	Varying	Data

Field Identifier

Field Identifiers

Field Identifier	Description
X'0001'	Resource application ID
X'0002'	Resource class
X'0003'	Resource group name
X'0004'	Resource ID
X'0005'	Attribute ID
X'0006'	Attribute qualifier
X'0007'	Attribute type (counter, gauge, enumerated, or total)
X'0008'	Attribute value if numeric
X'0009'	Attribute character value if not numeric
X'000A'	Minimum value (for gauge only)
X'000B'	Maximum value (for gauge only)
X'000C'	Period covered by the sample (only for enumerated attribute types in DH record types)

For details of the attributes that can be monitored see the *NetMaster for TCP/IP Graphical Performance Manager User's Guide*.

Access Control Audit Log SMF Record Format

SOLVE:Access Control writes an SMF record whenever a security definition is added, updated, or deleted.

The SMF record ID is chosen from the:

- SYSPARMS SMFID, if not a zero.
- SMFID set for NetMaster for TCP/IP reporting, if not a zero.
- 246 if the above are not used. If this conflicts with other SMF records, specify an SMFID with one of the methods above.

Action=Delete Class=Stack, Port, or Host

Position	Length in Bytes	Description
1 to 18	18	SMF record header
19 and 20	2	NetMaster category ID Always X'0700'
21 to 32	12 blank padded	NMID
33 to 40	8	USERID of the user who performed the action
41 to 48	8	TERMINAL on which the action was performed
49 to 56	8	UPDATE DATE in the format *ccyymmdd*
57 to 64	8	UPDATE TIME in the format HH.MM.SS
65 to 69	5	CLASS either STACK, HOST, or PORT
70 to 75	6	ACTION=DELETE
76 to 87	12	NAME of the resource being changed

Action=Add or Update Class=Stack

Position	Length in Bytes	Description
1 to 18	18	SMF record header
19 and 20	2	NetMaster category ID Always X'0700'
21 to 32	12 blank padded	NMID
33 to 40	8	USERID of the user who performed the action
41 to 48	8	TERMINAL on which the action was performed
49 to 56	8	UPDATE DATE in the format *ccyymmdd*
57 to 64	8	UPDATE TIME in the format HH.MM.SS
65 to 69	5	CLASS = STACK
70 to 75	6	ACTION=ADD or UPDATE
76 to 87	12	NAME of the resource being changed

Position	Length in Bytes	Description
88 to 92	5	TYPE and version of the stack, either IBMCS, IPMVS, AXS52, or AXS41
93 to 100	8	JOB NAME
101 to 108	8	STEP NAME
109 to 117	9	SECURITY MODE, either, TERMINATE or WARNING
118 to 121	4	SSID (TCPaccess stacks only)
122 to 125	4	SYN Limit (TCPaccess stacks only)
126 to 129	4	SYN WARNING Limit (TCPaccess stacks only)

Action=Update Class=Port

Position	Length in Bytes	Description
1 to 18	18	SMF record header
19 and 20	2	NetMaster category ID Always X'0700'
21 to 32	12 blank padded	NMID
33 to 40	8	USERID of the user who performed the action
41 to 48	8	TERMINAL on which the action was performed
49 to 56	8	UPDATE DATE in the format *ccyymmdd*
57 to 64	8	UPDATE TIME in the format HH.MM.SS
65 to 69	5	CLASS = PORT
70 to 75	6	ACTION=ADD or UPDATE
76 to 87	12	NAME of the resource being changed
88 to 92	5	PORT NUMBER (0 to 65355)
93 to 99	7	ALERT either NO, ADDRESS, UERID, or BOTH
100 to 103	4	UNIVERSAL ACCESS either NONE or READ
104	1	CHECK IP ADDRESS either Y or N
105	1	SAF CHECK OF USER ID? either Y or N
106 to 112	7	DAYS in seven character string
113 to 116	4	STARTING TIME in format HHMM
117 to 120	4	FINISHING TIME in format HHMM
121 to 128	8	DEFLOGON NAME (TN3270 only)
129 to end	variable	REMOTE PORTS list of remote ports and/or ranges (for example, 20, 30, 80-81)

Action=Update Class=Host

Position	Length in Bytes	Description
1 to 18	18	SMF record header
19 and 20	2	NetMaster category ID Always X'0700'
21 to 32	12 blank padded	SOLVE NMID
33 to 40	8	USERID of the user who performed the action
41 to 48	8	TERMINAL on which the action was performed
49 to 56	8	UPDATE DATE in the format *ccyymmdd*
57 to 64	8	UPDATE TIME in the format HH.MM.SS
65 to 69	5	CLASS = HOST
70 to 75	6	ACTION=ADD or UPDATE
76 to 87	12	NAME of the resource being changed
88 to 102	15	IP ADDRESS (LOW)
103 to 117	15	IP ADDRESS (HIGH)
118 to 124	7	ALERT either NO, ADDRESS, USERID, or BOTH
125 to 128	4	UNIVERSAL ACCESS either NONE or READ
129	1	Check IP ADDRESS? either Y or N
130	1	SAF Check User ID? either Y or N
131 to 137	7	DAYS seven character string
138 to 141	4	STARTING TIME in the format HHMM
142 to 145	4	FINISH TIME in the format HHMM
146 to 153	8	DEFLOGON NAME (TN3270 only)
154 to end	variable	REMOTE PORTS List of remote ports and/or ranges (For example, 20, 30, 80-81)

Glossary

Abstract Syntax Notation

See *ASN.1*.

Address Space

An OS/390 region running a started task, batch job, or TSO session.

ASN.1 (Abstract Syntax Notation One)

An OSI data description language, used by SNMP to define objects in Management Information Bases (MIBs). ASN.1 defines a number of data types and these are used in specifying objects.

Agent

See Network Management Agent.

BER (Basic Encoding Rules)

An OSI standard for encoding ASN.1 data structures, used by SNMP for encoding its messages.

Bridge

A communications device that connects two or more LANs and passes traffic between them.

Channel Card

Channel attachment interface for Cisco routers. The channel card is used to connect a host mainframe to a router to provide functions such as SNA and TCP/IP connectivity and offloading of TN3270 processing. An example of a channel card is the Channel Interface Processor (CIP).

CIP (Cisco Channel Interface Processor)

See Channel Card.

CMCC (Cisco Mainframe Channel Connection)

Also known as channel card. See *Channel Card*.

CNM (Communications Network Management)

IBM term for its SNA management facilities.

Community

An administrative definition of a set of SNMP agents and managers (management stations), used to grant common access rights.

Community Name

A password used by SNMP network management stations to access remote network management agents.

Communications Server for OS/390

IBM's TCP/IP stack released with OS/390 version 2.5. It combines IBM SNA/APPN and TCP/IP expertise to manage heterogeneous networking environments in a consolidated manner.

Communications Storage Manager (CSM)

Buffer management technology that enables application programs to put large amounts of data in buffers that can be accessed by other applications without the need to copy the data.

Datagram

The message unit transmitted by the Internet Protocol layer. It can be regarded as the TCP/IP counterpart of the Basic Information Unit in SNA networks.

Echo

A special ICMP signal sent to another node, which generates a reply. This is done as a test of connectivity.

Enterprise Extender (EE)

A data link control that allows high performance SNA access over IP networks.

Fragment

A subunit of a datagram. Datagrams that are too large for a particular subnetwork to transport are split into fragments. These fragments are recombined by the IP layer at the destination node.

Gateway

A device that connects two or more networks with different network architectures and routes traffic between them. A gateway also acts as a protocol translator. (The term *router* is more commonly used currently.)

Host

A network node that also performs application processing, and has an associated Internet address.

IBM TCP/IP

Any IBM TCP/IP stack, including Communications Server and TCP/IP for MVS. See also Communications Server for OS/390.

ICMP (Internet Control Message Protocol)

A protocol defined within the Internet Protocol (IP) for the purpose of performing various control functions, including: detecting and reporting network problems, testing connectivity, and tracing network routes.

ICS (Initialization and Customization Services)

The Initialization and Customization Services is an AutoAssist facility that helps you set up your region parameters.

INMC (Inter-Management Services Connection)

Links multiple systems.

Internet

A collection of linked networks running TCP/IP and related protocols. By convention, this word, when used with a capital I, denotes the (one) world-wide internetwork that has evolved from the ARPANET. The word *internet* with a lower case *i* denotes any particular implementation that uses the same protocol suite.

Internet Address

The logical address assigned to a node's interface to a network, for use by the Internet Protocol (IP). This address is a 32-bit number that is conventionally written as a sequence of the four decimal values of the bytes, starting from the highest value and separated by dots (periods); for example, 149.124.176.6.

Internet Protocol (IP)

The protocol that routes packets across multiple subnetworks (such as LANs and WANs). It is analogous to the protocol for the network layer in the OSI Reference Model.

Internet Suite of Protocols

The set of protocols in common usage on the Internet. These include IP, TCP, ICMP, SNMP, SMTP, Telnet, and FTP.

ISO

The International Organisation for Standardization.

LAN

Local Area Network

Management Information Base (MIB)

A definition of a set of data items used for management purposes. The items are known as objects, and represent features of network resources that can be managed. See also MIB-II.

MIB-II

The standard set of object definitions that all SNMP agents are required to support.

MIB Walk

The process of proceeding sequentially through a MIB, to examine object values. The sequence of objects is defined by the MIB hierarchy.

Multicast

Sent to more than one destination. Usually refers to a network message.

MVS (Multiple Virtual Storage)

The major IBM operating system.

NCL (Network Control Language)

The interpretive language of the Management Services. This language allows logical procedures (programs) to be developed externally and executed by the Management Services, on command. NCL contains a wide range of logic, built-in functions, and arithmetic facilities which can be used to provide powerful monitoring and automatic control functions.

NCL Procedure

A member of the Management Services procedures dataset, which comprises NCL statements and SOLVE or VTAM commands. The NCL statements and other commands are executed using an &CALL statement (or an EXEC or START command) that specifies the name of the procedure.

NCS (Network Control Services)

A facility of NetMaster for SNA that allows display and control of SNA network resources.

Neighbor

A logically adjacent node. This concept is used in routing protocols such as EGP.

Network Error Warning System

See NEWS.

NEWS (Network Error Warning System)

A feature of NetMaster that provides network error and traffic statistics and error alert messages.

Network Management Agent

A management program that executes at a managed network node, for the purpose of managing resources at that node under the direction of one or more network managers. The agent responds to SNMP commands from the network management station and sends SNMP traps to the station.

Network Manager

A management program that executes at a network node, for the purpose of allowing a network operator to send management commands to remote nodes and to receive event reports from these remote nodes. The network manager sends SNMP commands to network management agents at nodes, requesting the agents to act on Management Information Base (MIB) objects. It receives SNMP responses and traps from the agents, the latter indicating significant events at the nodes.

NMVT (Network Management Vector Transport)

A type of SNA request/response Record Unit (RU), used for transmitting management commands and data through an SNA network.

Node

A connection point in a communications network.

NTS (Network Tracking System)

A feature of NetMaster, NTS is an integrated network management and problem determination system that operates in multi-domain networks. It accumulates traffic statistics on a session and resource basis to allow network performance monitoring.

Object

An item of managed data maintained within a Management Information Base. Individual instances (values) of objects are also referred to as MIB variables.

Object Identifier

An ASN.1 data type. An object identifier is a sequence of integers separated by dots (periods). Each integer represents a level in a registration tree. The sequence of integers is used to uniquely identify an object in a Management Information Base. For example, the object identifier 1.3.6.1.2.1.1.1.0 identifies the sysDescr object in MIB-II.

Open Systems Adapter (OSA)

An S/390 hardware feature that combines S/390 I/O channel and network port functions. An OSA provides direct connectivity between S/390 applications and their clients.

Octet

An eight-bit value, synonymous with the term *byte*.

OCS (Operator Console Services)

A Management Services feature that provides general operational control and an advanced operator interface to VTAM for network management.

Open Systems Interconnection

See OSI.

OSI (Open Systems Interconnection)

A set of ISO standards for communication between computer systems.

Ping

A program that enables a test message to be sent to another node, requesting a reply, for the purpose of testing connectivity. This program uses ICMP Echo messages. Formally defined as Packet InterNet Grope, hence the acronym.

Protocol

A set of rules for achieving communication across a network.

RFC 1213

An Internet standard (Request For Comment) that defines MIB-II.

Router

A network node that receives packets, examines their destination addresses and makes decisions about the communication paths on which to forward them. Today, most routers support multiple network protocols, so the term router is replacing the term *gateway*, which was previously used to denote protocol translation and packet routing.

Server

A computer or a process that responds to a request for service from one or more clients.

Simple Network Management Protocol

See SNMP.

SNA (Systems Network Architecture)

This term describes the logical structure, formats, protocols, and operational sequences for transmitting communications data through the communication system in an IBM network. Intended by IBM as a set of standards that allows the integration of all the different IBM hardware/software products into a universal network. Introduced in 1974.

SNMP (Simple Network Management Protocol)

A protocol for management of internetworks, originally designed for management of devices running TCP/IP. The term is also commonly used for the architecture on which this protocol is based.

SNMP Manager

A management system that uses the Simple Network Management Protocol. Also known within SNMP as a network management station.

Subnet Mask

A 32-bit number used to partition the IP address space of an internetwork, to allow traffic to be routed to multiple subnetworks.

TCP (Transmission Control Protocol)

A transport level protocol that runs on top of IP and provides guaranteed delivery in sequence of data across an internetwork.

TCP/IP (Transmission Control Protocol/Internet Protocol)

Commonly used to signify the Internet protocol suite.

Transmission Control Protocol/Internet Protocol

See TCP/IP.

Trap

An SNMP event report.

UAMS (Userid Access Maintenance Subsystem)

The security component of Management Services that supports the definition of authorized users and their associated function and privilege levels.

UDP (User Datagram Protocol)

A transport level protocol that allows multiplexing of IP datagrams between different application processes but does not guarantee delivery. It is used by SNMP.

Unicast

Sent to a single destination. Usually refers to a network message.

VTAM(Virtual Telecommunications Access Method)

A suite of programs that controls communication between terminals and application programs.

WAN—Wide Area Network

Index

$

$AMEVFWD command, 7-25

$IPCALL, B-2

$TNCALL, B-2

&

&SOCKET verb set
- for a client, C-3
- for a server, C-2
- for PING and TRACEROUTE, C-3
- for UDP, C-3

&SOCKET verbs
- built-in functions, C-4
- error codes, C-14
- system variables, C-10

&ZSOCVERR format, C-14

2

2216 routers, defining, 2-8

A

access checking, connections, 5-5

access control, 5-2
- applications, 5-7
- enabling, 5-3
- how it works, 5-2
- users, 5-6

access lists, SNMP, 6-13

administration functions, controlling access, 5-10

alert history, implementing, 7-22

Alert Monitor panels
- Custom Trouble Ticket, 7-20
- Email a Trouble Ticket, 7-19
- History Logging Parameters Definition, 7-22
- Interface Definition, 7-19
- Trouble Ticket Data Entry Definition, 7-21

ALERTHIST parameter group, 7-22

alerts
- actions, 7-7
- defining, 7-6
- filtering, 7-23
- forwarding, 7-24
- using variables with, 7-6

applications
- access control, 5-7
- connection awareness, 1-8

ASCII and EBCDIC, translating between, 6-8

B

built-in functions, C-4

C

CICS commands, issuing, 6-17

Cisco channel cards
- defining, 2-8
- monitoring messages, 7-14

Cisco routers
- collecting data from, 2-8

specifying SNMP access, 2-8

commands, specific
$AMEVFWD, 7-25
IPMON, 3-5
LINK START, 7-11

communications server, enabling, 6-2

community names, 6-13
address ranges, 6-15
masks, 6-15
predefining, 6-14
specifying for Cisco routers, 2-8

connection awareness, 1-2
application, 1-8
how it works, 1-2
stacks, 1-5
user ID, 1-6
user ID with EASINET, 1-6
user ID with RACF, 1-7

connection detectors, defining, 7-10

connections
criteria for detectors, 7-9
monitoring, 7-9

console messages, monitoring, 7-17

custom events
detecting, 7-11
monitoring, 7-11

D

data space
exporting, 6-12
listing, 6-12
working with, 6-11

defining
2216 routers, 2-8
Cisco channel cards, 2-8
host to SAF, 5-9
NT server, 4-4
port to SAF, 5-9
resources, 2-5
stack, 2-5

detectors, event, 7-2
sample, 7-3
types, 7-3

E

EASINET, 5-7

EBCDIC and ASCII, translating between, 6-8

EDS events, B-11

enabling
access control, 5-3
Communication Server packet tracing, 6-2
management of SNA/VTAM resources and sessions, 6-7
performance monitoring, 4-2
TCPaccess packet tracing, 6-4

errors, socket, C-14

event detectors, 7-2
defining, 7-4
sample, 7-3
types, 7-3

events
passing to NetMaster, 1-3
processing and reporting, 1-9
reporting, 1-4

external writer, creating, 6-2

F

feedback and error codes, &SOCKET verbs, C-14

field identifiers, D-2

filtering alerts, 7-23

FTP
activity, logging, 1-1
failures
monitoring, 7-11

H

HP OpenView, interaction, A-1

I

IBM TCP/IP sockets interface

error codes, C-14
restrictions, C-13

ICMP, monitoring messages, 7-15

ICS parameter groups
ALERTHIST, 7-22
IPFILES, 1-10, 4-2
IPMONITOR, 1-10
IPMONITOR, 4-3, 7-12
REPORTER, 4-5
SOCKETMGMT, 6-16
TELNETTRT, 6-8
TN3270ACCESS, 5-6

implementing
Report Center, 4-5
reporting, 1-10
system events receivers and log options, 1-9

initialization files, configuring, 8-2

Internet Explorer
disabling security prompts, 6-10
setting up, 6-9

IP node monitor, setting up, 3-2

IP resources, monitoring, 2-2

IPFILES parameter group, 1-10, 4-2

IPLOG dataset, reorganizing, 1-11

IPMON command, 3-5

IPMONITOR parameter group, 1-10, 4-3, 7-12

L

LINK START command, 7-11

listeners, monitoring, 7-12

M

messages, passing to NetMaster, 1-3

monitor groups
attributes, 3-2
maintaining, 3-5

monitoring
alerts, 7-2
custom events, 7-11
FTP failures, 7-11
IP resources, 2-2
listeners, 7-12
performance, 4-1

multisystem support, enabling, 6-6

N

NCL &SOCKET verb set
for a client, C-3
for a server, C-2
for PING and TRACEROUTE, C-3
for UDP, C-3

NCS, 6-7

Netscape
disabling security prompts, 6-10
setting up, 6-9

NetView for AIX, interaction, A-1

NT server, defining, 4-4

NTS, 6-7

O

OSA, setting up, 2-8

P

packet tracing
Communication Server enabling, 6-2
TCPaccess enabling, 6-4

panels
Alert Monitor : Custom Trouble Ticket, 7-20
Alert Monitor : Email a Trouble Ticket, 7-19
Alert Monitor : History Logging Parameters Definition, 7-22
Alert Monitor : Interface Definition, 7-19
Alert Monitor : Trouble Ticket Data Entry Definition, 7-21
CAS : Valid Value List, 7-4
SNMP Defined Managers List, A-5
SocketMgmt Command Entry, 6-17
TCP/IP : Alert Automated Actions, 7-8
TCP/IP : Alert Definition, 7-6

TCP/IP : Connection Monitor Definition, 7-4
TCP/IP : Event Detector Controls List, 7-3
TCP/IP : IP Node Monitor Details, 3-3
TCP/IP : IP Node Monitor Group List, 3-5
TCP/IP : Monitor Group Details, 3-3
TCP/IP : SNMP Community Definition, 6-14
TCP/IP : Start CTRACE, 6-5

performance monitoring
enabling, 4-2
levels, 4-2

port, defining to SAF, 5-9

R

Report Center, implementing, 4-5

REPORTER parameter group, 4-5

reporting
configuration, 1-10
events, 1-4
implementing, 1-10

resources
defining, 2-5
definitions, 2-3
in multisystem, 2-3
working with, 2-3

restrictions on sockets interface
IBM TCP/IP, C-13
TCPaccess, C-14

REXEC connection, A-3

RFC 959, File Transfer Protocol, B-12

S

SAF
defining host to, 5-9
defining port to, 5-9
user ID, 5-8

security, Web applications, 6-10

sign on panel, 5-6

SMF record format, D-1

SNA/VTAM resources and sessions, 6-7

SNMP, 6-13
managers
defining, A-2, A-3
listing, A-5
maintaining, A-5
predefined community names, 6-14
security, 6-13
access lists, 6-13
community names, 6-13

socket error codes, C-14

Socket Management, 6-15
configuring, 6-15
customizing, 6-16
issuing CICS commands, 6-17

SOCKETMGMT parameter group, 6-16

sockets interface, C-2
datagram (UDP) sockets, C-3
restrictions
IBM TCP/IP, C-13
TCPaccess, C-14
stream (TCP) sockets, C-2

stacks
connection awareness, 1-5
defining, 2-5

state change, B-9

system images, 2-2
defining, 2-4

system variables, &SOCKET verb, C-10

T

TCP/IP
controlling access, 5-3
logging, setting up, 4-3

TCPaccess sockets interface
error codes, C-15
restrictions, C-14

Telnet
activity, logging, 1-1
translation tables, 6-8

TELNETTRT parameter group, 6-8

templates, using, 2-9

time zones, synchronizing, 6-11

TN3270ACCESS parameter group, 5-6

translation tables, Telnet, 6-8

trouble ticket interface
custom, 7-20
defining, 7-19
email, 7-19
implementing, 7-18

U

user
access control, 5-6
sign on panel, 5-6

user ID
connection awareness, 1-6
SAF, 5-8

V

verb set
for a client, C-3
for a server, C-2
for PING and TRACEROUTE, C-3
for UDP, C-3

W

Web applications, security, 6-10

Web interface, setting up, 6-9