

Model 2604
**T1/E1 Digital Access and
Cross-Connect System (DACs)**

Administrator's Reference Guide



Sales Office: +1 (301) 975-1000
Technical Support: +1 (301) 975-1007
E-mail: support@patton.com
WWW: www.patton.com

Document Number: 110051UA Rev. A
Part Number: 07MD2604DACs-ARG-A
Revised: February 20, 2002

Patton Electronics Company, Inc.
7622 Rickenbacker Drive
Gaithersburg, MD 20879 USA
Voice: +1 (301) 975-1000
Fax: +1 (301) 869-9293
Technical Support: +1 (301) 975-1007
Technical Support e-mail: support@patton.com
WWW: www.patton.com

Copyright © 2001, Patton Electronics Company. All rights reserved.

The information in this document is subject to change without notice. Patton Electronics assumes no liability for errors that may appear in this document.

The software described in this document is furnished under a license and may be used or copied only in accordance with the terms of such license.

Contents

About this guide	13
Audience.....	13
Structure.....	13
Typographical conventions used in this document.....	14
General conventions	14
Mouse conventions	15
1 Introduction	17
Introduction	18
Logging into the HTTP/HTML Administration Pages	18
HTTP/HTML and SNMP Object Format	18
Saving HTTP/HTML Object Changes	19
2 Home	21
Introduction	22
Operating Status Variables	23
% CPU Idle (boxIdleTime)	23
Running Since Last Boot (sysUpTime)	23
Total System Alarms (alarmTotal)	23
Immediate Actions	23
Record Current Configuration (storeConfig(1))	23
Hard Reset (hardReset(2))	24
Set Factory Default Configuration (forceDefaultConfig(3))	24
3 Import/Export	25
Introduction	26
Export Configuration	26
Import Configuration.....	28
4 Alarms	29
Introduction	30
Displaying the alarms window	30
Alarm Response Outputs	32
Relay Response	32
Minor Alarm SYSLOG Priority (minSyslogPriority)	32
Major Alarm SYSLOG Priority (majorSyslogPriority)	32
Minor Alarm SNMP Trap IP [address] (minorTrapIp)	32
Major Alarm SNMP Trap IP [address] (majorTrapIp)	32
Temperature Threshold	32
Current Box Temperature	32
Clear All Alarms	32
Alarms	32
Alarm ID	32

Alarm Name	32
Alarm Time	33
Alarm Count	33
Generate Alarm	33
Clear Alarm	33
Alarm Parameters	33
Types of Alarms	34
Modify Response—Configuring the alarm response system	34
Relay Response	34
Minor Alarm Syslog Priority & Major Alarm Syslog Priority	34
Minor Alarm SNMP Trap IP [address] (minSyslogPriority)	34
Major Alarm SNMP Trap IP [address] (majorSyslogPriority)	34
Temperature Threshold	35
Modify Alarms—Configuring alarm severity levels	35
5 DS0 Mapping.....	37
Introduction	38
Displaying the DS0 Mapping window.....	38
DACS Display Type	38
Help (DACS help information)	39
Static Connection	39
ID	39
Device Type	39
Device Number	39
Device Slots	40
Configuration	40
6 Clocking.....	41
Introduction	42
Configuring the System Clock Settings.....	42
Main Reference (daxClockMainRef)	42
Fallback Reference (daxClockFallbackRef)	43
Clock Status (daxClockFailure)	43
7 Ethernet.....	45
Introduction	46
Ethernet statistics.....	46
Alignment Errors (dot3StatsAlignmentErrors)	46
FCS Errors (dot3StatsFCSErrors)	46
Single Collision Frames (dot3StatsSingleCollisionFrames)	46
Multiple Collision Frames (dot3StatsMultipleCollisionFrames)	47
SQE Test Errors (dot3StatsSQETestErrors)	47
Deferred Transmissions (dot3StatsDeferredTransmissions)	47
Late Collisions (dot3StatsLateCollisions)	47
Excessive Collisions (dot3StatsExcessiveCollisions)	47
Other Errors (dot3StatsInternalMacTransmitErrors)	47

Carrier Sense Errors (dot3StatsCarrierSenseErrors)	47
Received Frames Too Long (dot3StatsFrameTooLongs)	47
Other Received Errors (dot3StatsInternalMacReceiveErrors)	48
Chip Set ID (dot3StatsEtherChipSet)	48
8 Filter IP	49
Introduction	50
Defining a filter	50
Name (filterIpName)	52
Direction (filterIpDirection)	52
Action (filterIpAction)	52
Source IP (filterIpSourceIp)	52
Source IP Mask (filterIpSourceMask)	52
Destination IP (filterIpDestinationIp)	52
Destination Mask (filterIpDestinationMask)	53
Source Port (FilterIpSourcePort)	53
Action (filterIpSourcePortCmp)	53
Destination Port (filterIpDestinationPort)	53
Action (filterIpDestinationPortCmp)	53
Protocol (filterIpProtocol)	53
TCP Established (filterIpTcpEstablished)	53
9 ICMP	55
Introduction	56
Block ICMP redirects (boxBLockIcmpRedirects)	56
ICMP Receive/Send Messages window	56
Total Received (icmpInMsgs)	56
Total Sent [icmpOutMsgs]	57
w/Errors (icmpInErrors, icmpOutErrors)	57
wo/Errors [icmpOutErrors]	57
Destinations Unreachable (IcmpInDestUnreachs, IcmpOutDestUnreachs)	57
Times Exceeded (icmpInTimeExcds, icmpOutTimeExcds)	57
Parameter Problems (icmpInParmProbs, icmpOutParmProbs)	57
Source Quenches (icmpInSrcQuenchs, icmpOutSrcQuenchs)	57
Redirects (icmpInRedirects, icmpOutRedirects)	58
Echos (icmpInEchos, icmpOutEchos)	58
Echo Repls (icmpInReps, icmpOutReps)	58
Time Stamps (icmpInTimestamps, icmpInTimestamps)	58
Time Stamp Repls (icmpInTimestampsReps) (icmpOutTimestampsReps)	58
Address Mask Requests (icmpInAddrMasks) (icmpOutAddrMasks)	58
Address Mask Repls (icmpInAddrMasksReps) (icmpOutAddrMasksReps)	58
10 IP.....	59
Introduction	61
IP main window	61
Forwarding (ipForwarding)	62

Default Time-To-Live (ipDefaultTTL)	62
Total Datagrams Received (ipInReceives)	62
Discarded for Header Errors (ipInHdrErrors)	62
Discarded for Address Errors (ipInAddrErrors)	62
Forwarded Datagrams (ipForwDatagrams)	62
Discarded for Unknown Protos (ipInUnknownProtos)	62
Discarded w/No Errors (ipInDiscards)	62
Total Deliveries (ipInDelivers)	63
Out Requests (ipOutRequests)	63
Out Discards (ipOutDiscards)	63
Discarded for No Routes (ipOutNoRoutes)	63
Reassembly Timeout (ipReasmTimeout)	63
# of Reassembled Fragments (ipReasmReqds)	63
# Successfully Reassembled (ipReasmOKs)	63
Reassembly Failures (ipReasmFails)	63
# Fragmented OK (ipFragOKs)	64
# Fragmented Failed (ipFragFails)	64
# Fragments Created (ipFragCreates)	64
# Valid but Discarded (ipRoutingDiscards)	64
Modify	64
Forwarding (ipForwarding)	64
Default Time-To-Live (ipDefaultTTL)	64
Addressing Information	65
IP addressing Information Details	65
Entry Interface Index (ipAdEntIfIndex)	65
Entry Subnet Mask (ipAdEntNetMask)	65
Entry Broadcast Address (ipAdEntBcastAddr)	65
Entry Reassembly Maximum Size (ipAdEntReasmMaxSize)	65
Routing Information	66
Destination (ipRouteDest)	66
Mask (ipRouteMask)	67
Gateway (RouteGateway)	67
Cost (RouteCost)	67
Interface (ipRouteIfIndex)	67
State (RouteState)	67
Add a route:	67
Advanced...	67
O/S forwarding table window.....	68
Destination (ipRouteDest)	68
Mask (ipRouteMask)	68
Next Hop (ipRouteNextHop)	68
Interface (ipRouteIfIndex)	68
Type (ipRouteType)	68
Protocol (ipRouteProto)	69

Info (ipRouteInfo)	69
IP Routing Destination window	70
Route Destination (ipRouteDest)	70
Mask (ipRouteMask)	70
Interface (ipRouteIfIndex)	70
Protocol (ipRouteProto)	70
Seconds Since Updated (ipRouteAge)	71
Tag (RouteTag)	71
Gateway (RouteGateway)	71
Cost (RouteCost)	71
State (RouteState)	71
Address Translation Information	71
Interface (ipNetToMediaEntry)	71
Net Address (ipNetToMediaNetAddress)	72
Physical (ipNetToMediaPhysAddress)	72
Type (ipNetToMediaType)	72
11 TCP	73
Introduction	74
TCP main window	74
Retransmit-Timeout Algorithm (tcpRtoAlgorithm)	74
Retransmit-Timeout Minimum (tcpRtoMin)	74
Retransmit-Timeout Maximum (tcpRtoMax)	74
Maximum Connections (tcpMaxConn)	75
Active Opens (tcpActiveOpens)	75
Passive Opens (tcpPassiveOpens)	75
Attempt/Fails (tcpAttemptFails)	75
ESTABLISHED Resets (tcpEstabResets)	75
Current ESTABLISHED (tcpCurrEstab)	75
Total Received (tcpInSegs)	75
Total Sent (tcpOutSegs)	75
Total Retransmitted (tcpRetransSegs)	75
Total Received in Error (tcpInErrs)	75
Total Sent w/RST Flag (tcpOutRsts)	75
TCP (Details)	76
Local Port (tcpConnLocalPort)	76
Remote Address (tcpConnRemAddress)	76
Remote Port (tcpConnRemPort)	76
State (tcpConnState)	76
12 UDP	79
Introduction	80
Handling of NETBIOS UDP Broadcasts (boxNetbiosUdpBridging)	80
Received (udpInDatagrams)	80
Received With No Ports (udpNoPorts)	80

Others Received with No Delivery (udpInErrors)	80
Sent (udpOutDatagrams)	80
Listener Table (udpTable)	81
Local Address (udpLocalAddress)	81
Local Port (udpLocalPort)	81
13 RIP Version 2	83
Introduction	84
RIP Version 2 main window.....	84
Route Changes Made (rip2GlobalRouteChanges)	84
Responses Sent (rip2GlobalQueries)	84
Adding a RIP address	84
RIP Version 2—Configuration.....	85
Address (rip2IfConfAddress)	85
Domain (rip2IfConfDomain)	86
Authentication Type (rip2IfConfAuthType)	86
Authentication Key (rip2IfConfAuthKey)	86
Send (rip2IfConfSend)	86
Receive (rip2IfConfReceive)	86
Metric (rip2IfConfDefaultMetric)	86
Status (rip2IfConfStatus)	87
RIP Version 2 (Statistics).....	87
Subnet IP Address (rip2IfStatAddress)	87
Bad Packets (rip2IfStatRcvBadPackets)	87
Bad Routes (rip2IfStatRcvBadRoutes)	87
Sent Updates (rip2IfStatSentUpdates)	87
Status (rip2IfStatStatus)	87
14 SNMP	89
Introduction	90
SNMP window.....	90
In	90
Packets (snmpInPkts)	90
Bad Version (snmpInBadVersions)	90
Bad Community Names (snmpInBadCommunityNames)	91
Bad Community Uses (snmpInBadCommunity)	91
ASN ParseErrors (snmpInASNParseErrs)	91
Error Status “Too Big” (snmpInTooBigs)	91
No Such Names (snmpInNoSuchNames)	91
Bad Values (snmpInBadValues)	91
Error Status “Read Only” (snmpInReadOnly)	91
Generated Errors (snmpInGenErrs)	91
Get/Get Next Variables (snmpInTotalReqVars)	91
Set Variables (snmpInTotalSetVars)	91
Get Requests (snmpInGetRequests)	91

- Get Next Requests (snmpInGetNexts)92
- Set Requests (snmpInSetRequests)92
- Get Responses (snmpInGetResponses)92
- Traps (snmpInTraps)92
- Out92
 - Out Packets (snmpOutPkts)92
 - Error Status "Too Big" (snmpOutTooBigs)92
 - No Such Names (snmpOutNoSuchNames)92
 - Bad Values (snmpOutBadValues)92
 - Generated Errors (snmpOutGenErrs)92
 - Get Requests (snmpOutGetRequests)92
 - Get Next Requests (snmpOutGetNexts)92
 - Set Requests (snmpOutSetRequests)92
 - Get Responses (snmpOutGetResponses)93
 - Traps (snmpOutTraps)93
 - Authentication Failure Traps (snmpEnableAuthenTraps)93
- 15 System 95**
 - Introduction97
 - System main window.....98
 - CPU98
 - Percentage CPU Idle (boxidletime)98
 - Time Slices Fully Utilized (boxCPUcritical)98
 - Time Slices 90% Utilized (boxCPUWarning)98
 - SNMP and HTTP98
 - Version (boxSnmpVersion)98
 - Super User Password (boxSnmpMasterPassword)98
 - User Password (boxSnmpMonitorPassword)98
 - LAN IP98
 - How to Obtain Address (boxIPAddressTechnique)99
 - Address(boxIPAddress)99
 - Mask(boxIPMask)99
 - Manufacturer99
 - Serial Number (boxManufactureDatecode)99
 - PCB Revision (boxManufacturePcbRevision)99
 - General Information (boxManufactureGeneralInfo)99
 - Message Blocks99
 - Packet Holding Message Blocks...99
 - Total (boxMsgBlksConfigured)99
 - Free (boxMsgBlksFree)99
 - Total Time Waited (boxCountMsgBlkTaskWait)99
 - Total Times Unavailable (boxCountMsgBlkUnavailable)100
 - Operating System Heap Memory100
 - Total Size (boxHeapSize)100

Free (boxHeapFreeSpace)	100
Largest (boxHeapLargestSpace)	100
Enclosure System	100
Internal Temperature (boxTemperature)	100
Highest Temperature (boxMaxTemperature)	100
Installation	100
Country (installCountry)	100
Other	100
Total DRAM Detected (boxDetectedMemory)	100
SystemID (sysObjectID)	100
Running Since Last Boot (sysUpTime)	101
System Manager (sysContact)	101
Box Name (sysName)	101
Physical Location (sysLocation)	101
Web Settings (boxBackgroundFlag)	101
Monitor Privilege (boxMonitorPrivilege)	101
System—Modify window	102
SNMP and HTTP	102
Version (boxSnmpVersion)	102
Super User Password (boxSnmpMasterPassword)	102
User Password (boxSnmpMonitorPassword)	103
LAN IP	103
Method to Obtain Address (boxIPAddressTechnique)	103
Address (boxIPAddress)	103
Mask (boxIPMask)	103
Installation	103
Country (installCountry)	103
Other	104
System Manager (sysContact)	104
Box Name (sysName)	104
Physical Location (sysLocation)	104
Web Settings (boxBackgroundFlag)	104
Monitor Privilege (boxMonitorPrivilege)	104
System—Packet Holding Message Blocks.....	105
Buffer Size (boxbuffersize)	105
No. of Buffers (boxbuffercount)	105
No. Free (boxbuffersfree)	105
No. of Tasks Waited (boxCountBufferTaskWait)	105
No. of Times Unavailable(boxCountBufferUnavailable)	105
16 System Log	107
Introduction	108
System Log Main Window	108
System Log—Modify	109

Daemons	109
SysLog Daemon IP Address(syslogDaemonIP)	109
SNMP Trap Daemon IP Address (syslogTrapIP)	109
Priority	109
Min Priority for SysLog Daemon (syslogDaemonPriority)	110
Min Priority for Console RS-232 (syslogConsolePriority)	110
Min Priority for Flash Storage (syslogFlashPriority)	110
Min Priority for SNMP Trap Daemon (syslogTrapPriority)	110
Min Priority for RAM (SyslogTablePriority)	111
Unix Facility (syslogUnixFacility)	111
Call Trace (syslogCallTrace)	112
Maintenance	112
Maintain Flash Storage (syslogFlashClear)	112
System Log—Volatile Memory.....	113
Time (slTick)	113
Message (slMessage)	113
System Log—Non-Volatile Memory	114
Time (slfTick)	114
Message (slfMessage)	114
17 T1/E1 Link.....	115
Introduction.....	118
T1/E1 Link Activity main window	119
Link (dsx1LineIndex)	119
Type (dsx1LineType)	119
Circuit ID (dsx1CircuitIdentifier)	119
Line Status (dsx1LineStatus).....	120
Failure States	120
Far End Alarm Failure	120
Alarm Indication Signal (AIS) Failure	121
Loss Of Frame Failure	121
Loss Of Signal Failure	121
Loopback Pseudo-Failure	121
TS16 Alarm Indication Signal Failure	121
Loss Of MultiFrame Failure	121
Far End Loss Of Multiframe Failure	121
Line Status—Configuration.....	122
Time Elapsed (dsx1TimeElapsed)	122
Valid Intervals (dsx1ValidIntervals)	122
WAN Circuit Configuration—Modify.....	123
Line Interface Settings	123
Circuit ID (dsx1CircuitIdentifier)	123
Line Type (dsx1LineType) Type (dsx1LineType)	123
Line Coding (dsx1LineCoding)	124

Receive Equalizer (linkRxEqualizer)	124
Line Build Out (linkLineBuildOut)	124
Yellow Alarm Format (linkYellowFormat)	124
FDL (dsx1FDL)	125
Test Settings	125
Force Yellow Alarm (linkYellowForce)	125
Loopback Config (dsx1LoopbackConfig)	125
Send Code (dsx1SendCode)	125
Error Injection (linkInjectError)	126
Yellow Alarm Severity ()	126
Red Alarm Severity ()	126
Near End Line Statistics—Current	127
Errored Seconds (dsx1CurrentESs)	127
Severely Errored Seconds (dsx1CurrentSESs)	127
Severely Errored Frame Seconds (dsx1CurrentSEFSs)	127
Unavailable Seconds (dsx1CurrentUASs)	127
Controlled Slip Seconds (dsx1CurrentCSSs)	127
Path Code Violations (dsx1CurrentPCVs)	127
Line Errored Seconds (dsx1CurrentLESs)	127
Bursty ErroredSeconds (dsx1CurrentBESs)	127
Degraded Minutes (dsx1CurrentDMs)	128
Line Code Violations (dsx1CurrentLCVs)	128
Near End Line Statistics—History	128
Interval (dsx1IntervalNumber)	128
Errored Seconds (dsx1intervaless)	128
Severely Errored Seconds (dsx1IntervalSESs)	128
Severely Errored Frame Seconds (dsx1IntervalSEFSs)	129
Unavailable Seconds (dsx1IntervalUASs)	129
Controlled Slip Seconds (dsx1IntervalCSSs)	129
Path Code Violations (dsx1IntervalPCVs)	129
Line Errored Seconds (dsx1IntervalLESs)	129
Bursty ErroredSeconds (dsx1IntervalBESs)	129
Degraded Minutes (dsx1IntervalDMs)	129
Line Code Violations (dsx1IntervalLCVs)	129
Near End Line Statistics—Totals	130
Errored Seconds (dsx1TotalESs)	130
Severely Errored Seconds (dsx1TotalSESs)	130
Severely Errored Frame Seconds (dsx1TotalSEFSs)	130
Unavailable Seconds (dsx1TotalUASs)	130
Controlled Slip Seconds (dsx1TotalCSSs)	130
Path Code Violations (dsx1TotalPCVs)	130
Line Errored Seconds (dsx1TotalLESs)	130
Bursty ErroredSeconds (dsx1TotalBESs)	130
Degraded Minutes (dsx1TotalDMs)	131

Line Code Violations (dsx1TotalLCVs)	131
Far End Line Statistics—Current.....	131
Time Elapsed (dsx1FarEndTimeElapsed)	131
Errored Seconds (dsx1FarEndCurrentESs)	131
Severely Errored Seconds (dsx1FarEndCurrentSESs)	131
Severely Errored Frame Seconds (dsx1FarEndCurrentSEFSs)	131
Unavailable Seconds (dsx1FarEndCurrentUASs)	131
Controlled Slip Seconds (dsx1FarEndCurrentCSSs)	132
Line Errored Seconds (dsx1FarEndCurrentLESs)	132
Path Code Violations (dsx1FarEndCurrentPCVs)	132
Bursty Errored Seconds (dsx1FarEndCurrentBESs)	132
Degraded Minutes (dsx1FarEndCurrentDMs)	132
Far End Line Statistics—History	132
Interval (dsx1FarEndIntervalNumber)	133
Errored Seconds (dsx1FarEndIntervalESs)	133
Severely Errored Seconds (dsx1FarEndIntervalSESs)	133
Severely Errored Frame Seconds (dsx1FarEndIntervalSEFSs)	133
Unavailable Seconds (dsx1FarEndIntervalUASs)	133
Controlled Slip Seconds (dsx1FarEndIntervalCSSs)	133
Line Errored Seconds (dsx1FarEndIntervalLESs)	133
Path Code Violations (dsx1FarEndIntervalPCVs)	133
Bursty Errored Seconds (dsx1FarEndIntervalBESs)	133
Degraded Minutes (dsx1FarEndIntervalDMs)	133
Far End Line Statistics—Totals	134
Errored Seconds (dsx1FarEndTotalESs)	134
Severely Errored Seconds (dsx1FarEndTotalSESs)	134
Severely Errored Frame Seconds (dsx1FarEndTotalSEFSs)	134
Unavailable Seconds (dsx1FarEndTotalUASs)	134
Controlled Slip Seconds (dsx1FarEndTotalCSSs)	134
Line Errored Seconds (dsx1FarEndTotalLESs)	134
Path Code Violations (dsx1FarEndTotalPCVs)	134
Bursty Errored Seconds (dsx1FarEndTotalBESs)	135
Degraded Minutes (dsx1FarEndTotalDMs)	135
18 T1/E1 Assignment.....	137
Introduction	138
Displaying the T1/E1 Assignment window.....	138
Slot	139
Device	139
Port #	139
Slot #	139
19 About.....	141
Introduction	142
Patton Electronics Company contact information	142

20 License..... 143

 Introduction144

 End User License Agreement144

 1. Definitions:144

 2. Title:145

 3. Term:145

 4. Grant of License:145

 5. Warranty:145

 6. Termination:145

About this guide

This guide describes configuring a Patton Electronics digital cross connect (DACs). This section describes the following:

- Who should use this guide (see “Audience”)
- How this document is organized (see “Structure”)
- Typographical conventions and terms used in this guide (see “Typographical conventions used in this document” on page 14)

Audience

This guide is intended for the following users:

- System administrators
- Operators
- Installers
- Maintenance technicians

Structure

This guide contains the following chapters:

- Chapter 1 describes configuring the Administration Page window
- Chapter 2 describes configuring the Home window
- Chapter 3 describes configuring the Import/Export window
- Chapter 4 describes configuring the Alarms window
- Chapter 5 describes configuring the DS0 Mapping window
- Chapter 6 describes configuring the Clocking window
- Chapter 7 describes configuring the Ethernet window
- Chapter 8 describes configuring the Filter IP window
- Chapter 9 describes configuring the ICMP window
- Chapter 10 describes configuring the IP window
- Chapter 11 describes configuring the TCP window
- Chapter 12 describes configuring the UDP window
- Chapter 13 describes configuring the RIP Version 2 window
- Chapter 14 describes configuring the SNMP window
- Chapter 15 describes configuring the System window
- Chapter 16 describes configuring the System Log window

- Chapter 17 describes configuring the T1/E1Link window
- Chapter 18 describes configuring the T1/E1 Assignment window
- Chapter 19 describes the contents of the About window
- Chapter 20 describes the contents of the License window
- Appendix A contains a table with the color code for the RJ-21X connector

Typographical conventions used in this document

This section describes the typographical conventions and terms used in this guide.

General conventions

The procedures described in this manual use the following text conventions:

Table 1. Text conventions

Convention	Meaning
Futura bold type	Indicates the names of menu bar options.
<i>Italicized Futura type</i>	Indicates the names of options on pull-down menus.
Futura type	Indicates the names of fields or windows.
Garamond bold type	Indicates the names of command buttons that execute an action.
< >	Angle brackets indicate function and keyboard keys, such as <SHIFT>, <CTRL>, <C>, and so on.
Are you ready?	All system messages and prompts appear in the Courier font as the system would display them.
% dir *.*	Bold Courier font indicates where the operator must type a response or command

Mouse conventions

The following conventions are used when describing mouse actions:

Table 2. Mouse conventions

Convention	Meaning
Left mouse button	This button refers to the primary or leftmost mouse button (unless you have changed the default configuration).
Right mouse button	This button refers the secondary or rightmost mouse button (unless you have changed the default configuration)
Point	This word means to move the mouse in such a way that the tip of the pointing arrow on the screen ends up resting at the desired location.
Click	Means to quickly press and release the left or right mouse button (as instructed in the procedure). Make sure you do not move the mouse pointer while clicking a mouse button. Double-click means to press and release the same mouse button two times quickly
Drag	This word means to point the arrow and then hold down the left or right mouse button (as instructed in the procedure) as you move the mouse to a new location. When you have moved the mouse pointer to the desired location, you can release the mouse button.

Chapter 1 **Introduction**

Chapter contents

Introduction	18
Logging into the HTTP/HTML Administration Pages	18
HTTP/HTML and SNMP Object Format	18
Saving HTTP/HTML Object Changes	19

Introduction

You may configure the digital cross connect (DACS) by using its internal HTTP/HTML Administration Pages. However, to enter into the HTTP/HTML pages, you must first define the LAN Address Technique, LAN IP Address, and LAN Subnet Mask for the DACS. If you have not done so, please refer to the Getting Started Guide that came with your DACS.

Logging into the HTTP/HTML Administration Pages

To log into the HTTP/HTML Administration pages, you must enter the 4-octet Internet Protocol (IP) (for example, *http://your.server.ip.address*) address as the Universal Resource Locator (URL) into a World-Wide Web (WWW) browser. After you enter the IP address, the DACS will ask for your user name and password as shown in figure 1.

Figure 1. DACS login window

Your DACS will accept the following default administrative passwords:

- superuser—this password carries full permission to change and view any parameters in the DACS
- monitor—this password allows full viewing of any non-password oriented variables.

Note For security reasons, we recommend that you change these passwords immediately after initial configuration.

HTTP/HTML and SNMP Object Format

In this document, we shall describe the variables found on each of the internal HTTP/HTML pages. This description will include brief definitions of the Patton Enterprise MIB or SNMP MIB II object identifiers wherever applicable. The format of the variables will resemble figure 2.



Figure 2. HTTP/HTML and SNMP object format

Saving HTTP/HTML Object Changes

Sometimes you will need to save changes that you have made in the HTTP/HTML pages. Do the following to make changes to read/write variables:

1. Select the appropriate **Modify** screen.
2. Make changes to the desired parameter.
3. Click on the **Submit Query** button.
4. Return to the **HOME** screen.
5. Click on the **Record Current Configuration** button.

Note Make sure you follow steps 1 through 5 when modifying the HTTP/HTML pages. Otherwise, your changes will be lost when the DACS is power-cycled.

Chapter 2 **Home**

Chapter contents

Introduction	22
Operating Status Variables	23
% CPU Idle (boxIdleTime)	23
Running Since Last Boot (sysUpTime)	23
Total System Alarms (alarmTotal)	23
Immediate Actions	23
Record Current Configuration (storeConfig(1))	23
Hard Reset (hardReset(2))	24
Set Factory Default Configuration (forceDefaultConfig(3))	24

Introduction

This chapter describes the HOME window—the first Administration Page that you see after logging into the DACS (see figure 3). From HOME, you can monitor current systems status, save any system configuration changes, or reset the system without power-cycling the DACS.

Note Clicking on the HOME link in the Configuration Menu pane will return you to the HOME page from any other page.

The HOME window is divided into two *panes*: the Configuration Menu pane on the left-hand side and the configuration/information pane (see figure 3). The Configuration Menu contains the links to the various DACS subsystems, while the configuration/information pane is where you can view status and other information, or make changes to the system configuration. Unlike the Configuration Menu pane, which looks the same no matter which subsystem page you are viewing, the configuration/information pane contents will change as you move from one subsystem page to another.

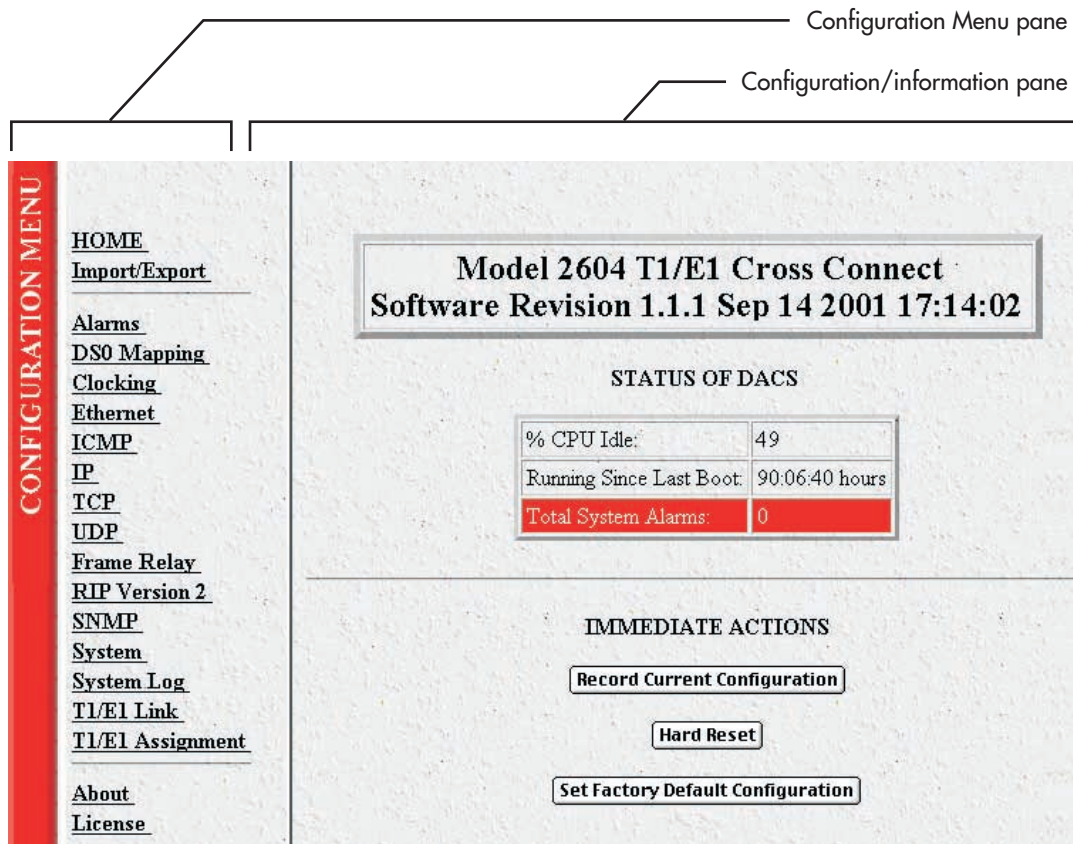
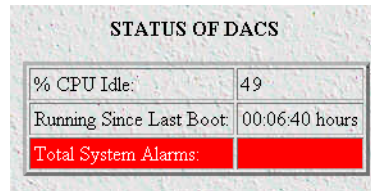


Figure 3. HOME page

Operating Status Variables

There are seven system variables which describe the immediate operating status of the DACS. These variables are shown in figure 4 and are described in the following sections.



STATUS OF DACS	
% CPU Idle:	49
Running Since Last Boot:	00:06:40 hours
Total System Alarms:	

Figure 4. STATUS menu

% CPU Idle (*boxIdleTime*)

This is an indication of the amount of system CPU power which is not being utilized by the Model 2604. The return value is a percentage of free CPU cycles since the last time the variable was read.

Running Since Last Boot (*sysUpTime*)

The time (in hundredths of a second) since the DACS was last power-cycled.

Total System Alarms (*alarmTotal*)

Total number of alarms currently active in the system.

Immediate Actions

There are several immediate actions (see figure 5) in superuser mode which will cause the DACS to operate according to the descriptions in the following sections.

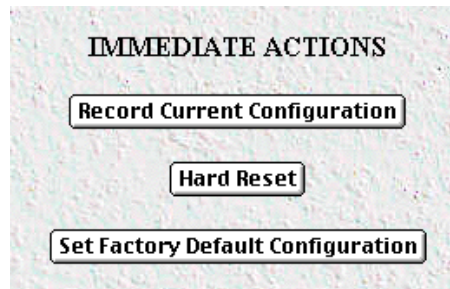


Figure 5. Immediate Actions buttons

Record Current Configuration (*storeConfig(1)*)

This feature saves the current configuration in permanent FLASH memory. In other words, configuration changes made in the subsystem web pages become permanent when you select **Record Current Configuration**.

1. Configuration changes in the DACS are made in the subsystem web pages by clicking **Submit Query**. This stores the configuration in volatile DRAM (Dynamic RAM) only. Since the **Submit Query** changes take immediate effect, the administrator can test different configuration parameters without needing to change the FLASH configuration at this moment.

- Without clicking on **Record Current Configuration**, all configuration changes will be lost if the power is recycled. After doing the **Record Current Configuration** save, the current configuration of the DACS will not be lost with power cycling.

Note The most important step after completing the configuration is to save it in permanent memory by clicking on **Record Current Configuration**.

Hard Reset (*hardReset(2)*)

This button causes the DACS to perform a cold restart. When you select **Hard Reset**, the DACS requests confirmation for the execution of this command. Then, the DACS will disconnect all current sessions, re-initialize the interfaces, and re-load configuration parameters from FLASH.

Set Factory Default Configuration (*forceDefaultConfig(3)*)

This button clears out the configuration in FLASH and loads the factory default parameters into FLASH memory. The factory default settings will not execute on the DACS until it is re-booted, for example by doing a **Hard Reset**.

Note **Set Factory Default Configuration** will delete the DACS's Ethernet IP address and any other site specific settings made for your particular installation. You will have to re-enter the DACS's Ethernet IP address and netmask using the front panel control port in order to use the HTTP/HTML Management pages.

Chapter 3 **Import/Export**

Chapter contents

- Introduction26
- Export Configuration26
- Import Configuration.....28

Introduction

The Import/Export function enables you to make a backup (or *export*) copy of your DACS's configuration parameters. By exporting the configurations, the saved files can quickly be loaded, or *imported*, into a replacement DACS—greatly speeding up the installation process should a DACS need replacing.

Note All actions for Import/Export require superuser access privileges.

To import or export a configuration, click on Import/Export under the Configuration Menu to display the Import/Export main window (see figure 6).

IMPORT / EXPORT DACS

EXPORT CURRENT FLASH CONFIGURATION

The current power up settings as stored in the system flash will be dumped to your screen. You may then save them in a file using the "save as" function in your web browser for later import back into the system.

Note that the information which is exported is the current hard storage settings, **NOT** the currently running settings. You may want to issue a "Record Current Configuration" on the home page before dumping the configuration.

[Export Flash...](#)

IMPORT FLASH CONFIGURATION FROM FILE

If you have previously exported the system configuration to a file then you can submit that file below and the system will update its flash configuration from the data saved in the file.

After this operation the system should be rebooted to activate the new settings. The configuration is loaded directly into the flash and so does **NOT** immediately modify any settings.

WARNING: This operation will erase whatever settings you currently have in the system.

Browse...

Submit Query

Figure 6. Import/Export main window

Export Configuration

Note The exported configuration file is a text-format file. Do not try, however to edit the operating characteristics contained in the file.

Note The parameters that will be exported are the power-up settings as they are stored in flash memory and *may not* be the current operating parameters. To ensure that you export the most current parameters, go to HOME, then click on the **Record Current Configuration** button under Immediate Actions.

To export the flash configuration, click on the Export Flash link on the Import/Export main page. The DACS will display text configuration information resembling that shown in figure 7.

```

*****
Flash configuration data for: Server

The data below is the current hexadecimal representation
of your configurable data in the system. Select the
File/Save As option to save the data to a file. This
file can be reloaded into your system at a later date.

You may edit and comment the top portion of this file
but do not modify any data after the "@" symbol. Also,
do not put an "@" symbol in the comment area.

START CONFIGURATION DATA
@

fconfigData.5 = "0x01:00:00:00:04:04:04:04:04:04:04:04:08:08:08:08:08:08:04:04:04:04
:04:04:04:04:08:08:08:08:08:08:08:08:04:04:04:04:04:04:04:08:08:08
:08:08:08:08:04:04:04:04:04:08:08:08:08:08:08:08:00:00:00:00

fconfigData.6 = "0x01:00:00:00:04:04:04:04:04:04:04:04:08:08:08:08:08:08:04:04:04:04
:04:04:04:04:08:08:08:08:08:08:08:08:04:04:04:04:04:04:04:08:08:08
:08:08:08:08:04:04:04:04:04:08:08:08:08:08:08:08:08:00:00:00:00

```

Figure 7. Typical DACS flash memory configuration data

To save the displayed data as a text file, select the **Save** option on your browser (see figure 8). For example, under Netscape, select **File > Save As**. A dialog box will display enabling you to save the contents of the export parameters to a text file. Select the location where you want the file stored, type a file name, and click **Save**.

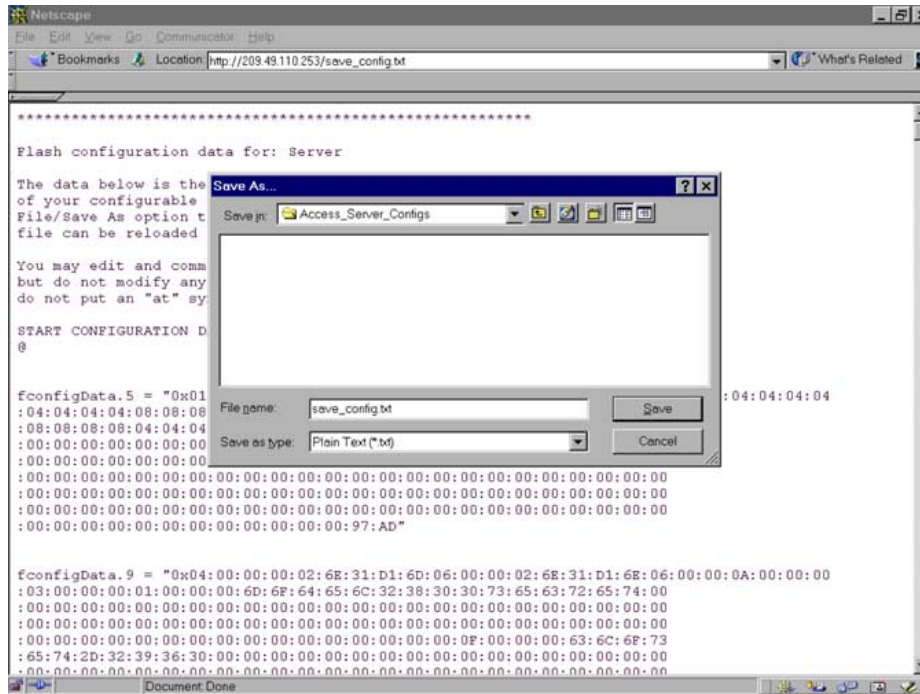


Figure 8. Saving the DACS flash memory configuration data as a text file

Import Configuration

To import a configuration file into the DACS, type the complete path and filename for the configuration file you wish to load or click on the **Browse...** button to select the desired file, then click on the **Submit Query** button (see figure 6 on page 26).

Upon successfully importing the file, the DACS will display *Configuration Load Complete*, indicating that the new operating parameters have been loaded into flash memory.

Click on **HOME** under the Configuration Menu, then click on the **Hard Reset** button under Immediate Actions.

Note Do not select **Record Current Configuration** after importing configuration parameters because the configuration is imported directly into non-volatile FLASH memory. Upon doing a Hard Reset the imported configuration is now the operational software is RAM.

Chapter 4 Alarms

Chapter contents

Introduction	30
Displaying the alarms window	30
Alarm Response Outputs	32
Relay Response	32
Minor Alarm SYSLOG Priority (minSyslogPriority)	32
Major Alarm SYSLOG Priority (majorSyslogPriority)	32
Minor Alarm SNMP Trap IP [address] (minorTrapIp)	32
Major Alarm SNMP Trap IP [address] (majorTrapIp)	32
Temperature Threshold	32
Current Box Temperature	32
Clear All Alarms	32
Alarms	32
Alarm ID	32
Alarm Name	32
Alarm Time	33
Alarm Count	33
Generate Alarm	33
Clear Alarm	33
Alarm Parameters	33
Types of Alarms	34
Modify Response—Configuring the alarm response system.....	34
Relay Response	34
Minor Alarm Syslog Priority & Major Alarm Syslog Priority	34
Minor Alarm SNMP Trap IP [address] (minSyslogPriority)	34
Major Alarm SNMP Trap IP [address] (majorSyslogPriority)	34
Temperature Threshold	35
Modify Alarms—Configuring alarm severity levels	35

Introduction

The DACS has an extensive alarm reporting system which enables users to configure, monitor, and test major and minor alarms. The alarm system can be set to notify of equipment failures (for example, a power supply failure) or T1/E1 port malfunctions. There are 83 alarms that can be configured by the system administrator to generate alerts based on the condition of the DACS.

Displaying the alarms window

Click on Alarms under the Configuration Menu to display the Alarm System main window (figure 9).

Note The system administrator can manually generate a specific alarm for testing purposes or clear the alarm counters from the main window.

Alarm System: Total System Alarms 29 DACS

[Modify Response...](#) [Modify Alarms...](#)

Alarm Response Outputs

Relay Response: major(2)
 Minor Alarm Syslog Priority: priorityInfo(20)
 Major Alarm Syslog Priority: prioritySystem(80)
 Minor Alarm SNMP Trap IP: 0.0.0.0
 Major Alarm SNMP Trap IP: 0.0.0.0
 Temperature Threshold: 65 celsius
 Current Box Temperature: 49 celsius
 Clear All Alarms:

Alarms

ID	Alarm Name	Alarm Severity	Time Since Alarm	Alarm Count	Generate Alarm	Clear Alarm
1	Box: Over Temperature	majorSelfClearing(4)	0.00 sec	0	<input type="button" value="Generate Alarm"/>	<input type="button" value="Clear Alarm"/>
2	Box: Power Supply 1 Fail	minorSelfClearing(3)	0.00 sec	0	<input type="button" value="Generate Alarm"/>	<input type="button" value="Clear Alarm"/>
3	Box: Power Supply 2 Fail	majorSelfClearing(4)	4 days 00:34:40 hours	1	<input type="button" value="Generate Alarm"/>	<input type="button" value="Clear Alarm"/>
4	Box: Main Clock Fail	minorSelfClearing(3)	0.00 sec	0	<input type="button" value="Generate Alarm"/>	<input type="button" value="Clear Alarm"/>

Figure 9. Alarms main window

The T1/E1 DACS Multiplexer has four methods to notify an alarm condition:

1. Front panel LED—The front panel ALARM LED has three states that indicate the presence and severity of an alarm. The states are:
 - Off—No alarm present
 - Solid—Minor alarm
 - Flashing—Major alarm.

Note The POWER LED will flash if a power supply failure alarm is present.

2. Administration web page indication—The Alarms window of the administration page uses red highlighting to indicate which items are in an alarm state (see figure 10).

ID	Alarm Name	Alarm Severity	Alarm Time	Alarm Count	Generate Alarm	Clear Alarm
1	Box Over Temperature	major(2)	0.01 sec	1	Generate Alarm	Clear Alarm
2	Box Power Supply 1 Fail	major(2)	0.00 sec	0	Generate Alarm	Clear Alarm
3	Box Power Supply 2 Fail	major(2)	0.00 sec	0	Generate Alarm	Clear Alarm

Figure 10. Sample alarm indication

3. SYSLOG/SNMP—For external notification, the DACS can be configured to send a SYSLOG message or an SNMP TRAP to an external management host. To configure the alarm response for either SNMP Traps or SYSLOG messages, click on the Alarm Response link (go to “Alarm Parameters” on page 33).
4. Alarm Relay—Located on the rear of the chassis, the Alarm Relay is a 3-position terminal block. The Alarm Relay may be configured to indicate when a major alarm, a minor alarm or both major and minor alarms occur. It may also be disabled. The 3-position terminal block numbers its pins from 1 to 3 from left to right. When no alarms are indicated, pins 1 and 2 are normally closed and pins 2 and 3 are normally open. Upon the occurrence of an alarm—the type is configured as major, minor, or both—pins 1 and 2 are open and pins 2 and 3 are closed.

Besides enabling a user to view current alarm status, you may manually generate an alarm as a test and clear the alarm time and count variables. The Alarms main window also contains links to the following:

- Modify Response—for configuring how the Alarm Response Outputs for notifying administrators of an alarm (see “Alarm Parameters” on page 33)
- Modify Alarms—Clicking on this link takes you to a window where you can configure the importance or severity of each individual alarm. The severity of the alarm type may generate a minor, major, minor self-clearing, or major self-clearing alarm. Any alarm type may be disabled. (“Modify Alarms—Configuring alarm severity levels” on page 35)

Alarm Response Outputs

Alarm Response Outputs display the current setting for handling alarm notification via the different Alarm Response Outputs. To change the Alarm Response Outputs parameters, refer to “Alarm Parameters” on page 33.

Relay Response

The relay of the Alarm Port on the rear of the chassis will be activated when a major, minor, or both major and minor alarm is generated. The Alarm Port may also be disabled.

Minor Alarm SYSLOG Priority (minSyslogPriority)

Sets the priority of the minor alarm SYSLOG message that will be generated upon the occurrence of a minor alarm.

Major Alarm SYSLOG Priority (majorSyslogPriority)

Sets the priority of the major alarm SYSLOG message that will be generated upon the occurrence of a major alarm.

Minor Alarm SNMP Trap IP [address] (minorTrapIp)

Displays the IP address of a SNMP management station for receiving the SNMP trap messages upon the occurrence of an active minor alarm. The SNMP trap messages are sent in UDP datagrams. When the IP address is set to 0.0.0.0, no trap messages will ever be sent.

Major Alarm SNMP Trap IP [address] (majorTrapIp)

The same function as the Minor Alarm Trap IP except for only the occurrence of active major alarms.

Temperature Threshold

An alarm will be generated when the box temperature exceeds this temperature value in degrees Celsius.

Current Box Temperature

The internal temperature in the box in degrees Celsius.

Clear All Alarms

Click on this button to clear all the alarms (that is, to reset all the alarms). This clearing action will, for all the alarms, reset the alarm, reset Time Since Alarm to 0.00 seconds, and reset the Alarm Count to 0 (zero).

Alarms

This portion of the Alarms main window displays the alarm status table, where you can view current alarm status, manually generate an alarm as a test, and clear the alarm time and alarm count variables.

Alarm ID

This number identifies the alarm item.

Alarm Name

The alarm items are grouped into two categories: system and WAN trunk alarms. The system group category lists DACS temperature and power supply status. The WAN category monitors the T1/E1/PRI ports for yellow and red alarms.

Alarm Time

The Alarm Time column displays the number of seconds the alarm has been activated.

Alarm Count

The Alarm Count column indicates how many times the alarm has occurred and is useful for monitoring self-clearing alarms.

Generate Alarm

For testing purposes, clicking the **Generate Alarm** button next to each alarm name will cause that alarm condition to be activated.

Clear Alarm

Clicking the **Clear Alarm** button resets the alarm to a non-alarm condition.

Alarm Response System

Alarm Response Outputs

Relay Response:	major(2) ▾	Submit
Minor Alarm Syslog Priority:	priorityInfo(20) ▾	Submit
Major Alarm Syslog Priority:	prioritySystem(80) ▾	Submit
Minor Alarm Trap IP:	0.0.0.0	Submit
Major Alarm Trap IP:	0.0.0.0	Submit
Temperature Threshold:	65	Submit

Figure 11. Alarm Response System window

Alarm Parameters

The Alarm Status Table on the Alarm System main page displays the current alarm status. You may also manually generate an alarm as a test and clear the alarm, the alarm time and the alarm count variables.

- Alarm ID—The Alarm ID identifies the alarm numerically. E.g., Alarm ID #2 identifies the alarm named “Box: Power Supply I Failed.”
- Alarm Name—The alarm items are grouped into two categories: Box and WAN alarms. The Box alarm group contains the alarms “Over Temperature” and “Power Supply Fail” for each of the two power supplies. The WAN alarm group includes yellow and red alarms.
- Alarm Severity—For each alarm, it shows whether the alarm is disabled or configured to generate a major, minor, major self-clearing, or minor self-clearing alarm.
- Time Since Alarm—Elapsed time since the alarm occurred.
- Alarm Count—The number of times this alarm has occurred since it has been cleared. It is also for monitoring self-clearing alarms.
- Generate Alarm—For testing a particular alarm, click on Generate Alarm. This activates the alarm as if the actual trigger event had occurred.

- Clear Alarm—Clearing the alarm resets the alarm, resets Time Since Alarm to 0.00 seconds and resets Alarm Count to 0 (zero).

Types of Alarms

- Box Alarm Group
 - Box: Over Temperature—When the internal box temperature exceeds the temperature threshold under Modify Response..., an alarm will be generated.
 - Box: Power Supply I – II Fail—An alarm will be generated when a power supply fails.
- WAN Alarm Group
 - WAN 1 – 4: Yellow Alarm—When a WAN port sees a yellow alarm, the specific WAN alarm will be sent.
 - WAN 1 – 4: Red Alarm—When a WAN port sends a red alarm, the specific WAN alarm will be sent.

Modify Response—Configuring the alarm response system

The alarm response outputs refer to points of external notification. Note that the front panel Alarm LED and the web administration pages will always indicate an occurrence of an active alarm. To configure each alarm response output, click on **Modify Response**. The Alarm Response System page appears (see figure 11). Choose the alarm response output that you want to configure with the pull down menu. After configuring a specific alarm response output, remember to click on **Submit Query** before going to the next alarm response output; otherwise the change will not occur.

Relay Response

The relay may be set to go active for minor alarms, major alarms, or both. It may also be disabled with the parameter “none.”

Minor Alarm Syslog Priority & Major Alarm Syslog Priority

When a minor/major alarm occurs, a message of the selected priority is sent to the Syslog engine. The Priority levels are priorityDisable(100), prioritySystem(80), priorityService(60), priorityOddity(40), priorityInfo(20), priorityDebug(10), and priorityVerbose(5). For more information on Syslog messages, refer to Chapter 16, “System Log”.

Minor Alarm SNMP Trap IP [address] (minSyslogPriority)

Upon the occurrence of a minor alarm, an SNMP Trap message is sent to a host system (or a management station). This parameter is the IP address of the host running the SNMP Trap daemon. When the IP address is set to 0.0.0.0 no SNMP Trap message will be sent.

Major Alarm SNMP Trap IP [address] (majorSyslogPriority)

This parameter functions in the same manner as the Minor Alarm SNMP Trap IP [address] except it applies to major alarms. Upon the occurrence of a major alarm, an SNMP Trap message is sent to a host system (or a management station). This parameter is the IP address of the host running the SNMP Trap daemon. When the IP address is set to 0.0.0.0 no SNMP Trap message will be sent.

Temperature Threshold

An alarm message is generated when the internal box temperature exceeds this threshold value (degrees Celsius). You can change the threshold temperature, but we recommend that you use the factory default.

Modify Alarms—Configuring alarm severity levels

Clicking on Modify Alarms window (see figure 12) displays a table listing each individual alarm. From this page you can configure the severity for each alarm (such as major, minor, major self-clearing, and minor self-clearing). Each alarm can be disabled as appropriate for your application.

The screenshot shows a window titled "Alarm System" with a sub-section "Alarms". It contains a table with four columns: "ID", "Alarm Name", "Alarm Severity", and "Alarm Options". There are 11 rows of data. The "Alarm Severity" column contains dropdown menus, and the "Alarm Options" column contains "Submit Query" buttons. A dropdown menu is open for the second row, showing the following options: "ignore(0)", "minor(1)", "major(2)", "minorSelfClearing(3)", and "majorSelfClearing(4)".

ID	Alarm Name	Alarm Severity	Alarm Options
1	Box:Over Temperature	major(2)	Submit Query
2	Box:Power Supply 1 Fail	ignore(0) minor(1) major(2) minorSelfClearing(3) majorSelfClearing(4)	Submit Query
3	Box:Power Supply 2 Fail	major(2)	Submit Query
4	WAN1:Yellow Alarm	minorSelfClearing(3)	Submit Query
5	WAN2:Yellow Alarm	minor(1)	Submit Query
6	WAN3:Yellow Alarm	minor(1)	Submit Query
7	WAN4:Yellow Alarm	minor(1)	Submit Query
8	WAN1:Red Alarm	major(2)	Submit Query
9	WAN2:Red Alarm	major(2)	Submit Query
10	WAN3:Red Alarm	major(2)	Submit Query
11	WAN4:Red Alarm	major(2)	Submit Query

Figure 12. Modify Alarms settings window

There are 83 alarms that can be independently configured to generate alarm messages. Each alarm item can be set for one of the following severity levels:

- Ignore(0)—Do not generate an alarm.
- Minor(1)—Generate a minor alarm that will not reset until the administrator manually clears it.
- Major(2)—Generate a major alarm that will not reset until the administrator manually clears it.

- **MinorSelfClearing(3)**—Generate a minor alarm that automatically clears after a fixed period of time. If the alarm condition has not ceased, the alarm will be automatically cleared, but another alarm will be immediately generated. If the alarm condition has ceased, the alarm will be automatically cleared after the same fixed period of time.
- **MajorSelfClearing(4)**—Same as **MinorSelfClearing(3)** except that it is a Major alarm instead of Minor.

Note For maximum application flexibility, the administrator shall choose which constitute major or minor alarm. Some examples of typical major and minor alarms include:

- Box: Over Temperature—Major Alarm
- WAN 1: Red Alarm—MajorSelfClearing
- WAN 1: Yellow Alarm—MinorSelfClearing

To set an alarm, click on the drop-down menu for the desired alarm item, choose the new setting followed by clicking on **Submit Query**.

Chapter 5 **DSO Mapping**

Chapter contents

- Introduction38
- Displaying the DSO Mapping window.....38
 - DACS Display Type38
 - Help (DACs help information)39
- Static Connection39
 - ID39
 - Device Type39
 - Device Number39
 - Device Slots40
- Configuration40

Introduction

One of the remote locations is the CPE's DACS. The second remote location is typically connected through some WAN port's time slots. For communication between these remote locations they shall be connected together within the DACS. These connections are configured in the DS0 mapping window.

Displaying the DS0 Mapping window

Do the following:

1. Click on DS0 Mapping under the Configuration Menu. The DS0 Mapping Configuration window displays (see figure 13).

DS0 Mapping Configuration DACS

DACS Display Type

Configure Static Connections

Dev Type A: Dev Num A: Dev Slots A: Dev Type B: Dev Num B: Dev Slots B:

t1-e1(1) t1-e1(1)

Static Connections Table

ID	Device Type A	Device Number A	Device Slots A	Device Type B	Device Number B	Device Slots B
1	t1-e1(2)	port2(2)	1, 4	t1-e1(3)	port3(3)	1, 4
2	t1-e1(1)	port1(1)	10, 15	t1-e1(3)	port3(3)	10, 15

Figure 13. DS0 Mapping Configuration window

The following sections describe the contents of the DS0 Mapping Configuration window.

DACS Display Type

You can configure or *map* the static connections by using the Long Format or the Command Line Format.

- displayLongForm(0)—This is the easiest to use by selecting the options from the pull-down menus.
- displayCliForm(1)—If you prefer the command line format, select displayCliForm(1) and click on the Submit Query button. Consult the following sections for the format of the command line.

Help (DACS help information)

Clicking on the **Help** button displays the DACS Help Information window (see figure 14). The purpose of this window is to help the user learn how to add DSO connections using the DACS HTML pages. This window define all of the parameters available within this web page. If you are using the Command Line Format to make connections, scroll down the window to the heading **Command Line Format**. The information contained in the Help window is also covered in this chapter.

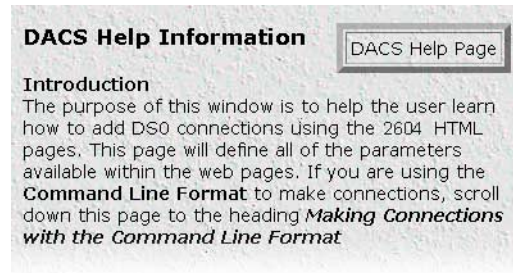


Figure 14. Example DACS Help Information window

Static Connection

Pull Down Menus

Each pull down window signifies part of a connection. Each connection is made up of an "A" side and a "B" side. These names have been arbitrarily chosen and do not signify the direction that data will travel. All data will be bi-directional. There are three parameters that need to be defined for each connection, they are:

- Device Type
- Device Number
- Device Slots

ID

Then ID number identifies each mapping with a unique number. The number is automatically assigned sequentially when a static connection is entered. The ID number begins at "1."

Device Type

The Device Type specifies the physical interface that you will be connecting. Within the 2604 the user has the option of selecting either a T1/E1 WAN line or a DACS. While the one side may be a T1/E1 WAN line and the other a DACS, note that both sides of the connection can be T1/E1 or both sides can be iDSL. The two Device Types are t1-e1(1) and t1-e1(2).

Device Number

The next step in creating a connection through the DACs is to select the port that you would like to use. This corresponds to the Port Number for the devices selected in the previous step. For example, if you would like to make a connection to port 3 (referring to DACS #3), then select "Port 3" in the "Device Number" field. Note that there are only four t1-e1 ports so you may not select t1-e1 ports 5 - 24. This will generate an error in the system. Since there are 24 DACSs within the 2604, you may choose any of the 24 ports.

Device Slots

The "slots" input identifies the DS0 channels—each DS0 channel is 64 kbps—that you would like to connect. Each time slot in a T1 or E1 WAN port has 24 or 31 DS0 channels, respectively. When selecting the slots you must select the same number of slots on the "A" and "B" side of the connection. The slots are selected by entering a string that represents the slots. For a WAN port configured as a T1, the available slots are numbered from 1 - 24. For a WAN port configured as an E1, the available slots are 1 - 31. The following notation should be used for entering the slots. Several examples are given below.

- dash: (-) 1 - 4
- comma: (,) 1,4,9
- combo: 1 - 2, 3,6 - 7

For example, to connect a T1 Port using timeslots 1,2, 5, 6, 7, and 15, you can input any of the following strings:

1,2, 5-7, 15

1 - 2, 5,6,7,15

1 - 2, 5 - 6, 7, 15

Configuration

The user can make connections in the box using two different methods. The easiest way is by using the pull down windows provided. But the user can also add connection using the command line format by entering a text string. To input a static connection into the box using the text string. Use the following convention:

- DeviceA:PortA:SlotsA/DeviceB:PortB:SlotsB

Device Options - The interface that you would like to select

t1-e1

Port Options - The Port Number (starting at 1) may be one of the four WAN ports. To configure Slots (DS0 channels), choose the slots that you would like to use. The following notations are allowed:

1) dash (-): 1 - 4

2) comma (,): 1,4,9

3) combination of dashes and commas: 1 - 2, 3,6 - 7

Example: To connect a T1 line, Port 1, timeslots 1 and 2 to a T1 line, Port 2, timeslots 5 and 6, input the following string:

t1-e1:1:1-2/t1-e1:2:5-6

Chapter 6 **Clocking**

Chapter contents

- Introduction42
- Configuring the System Clock Settings.....42
 - Main Reference (daxClockMainRef)42
 - Fallback Reference (daxClockFallbackRef)43
 - Clock Status (daxClockFailure)43

Introduction

Click on Clocking in the Configuration Menu to display the System Clocking Configuration main window (see figure 15).

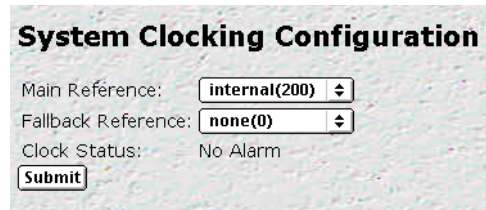


Figure 15. System Clocking Configuration window

The Clocking window is where the primary and secondary clocking sources that synchronize all DS0 channels are configured. The DACS uses a single clock source for all DS0 channels. You select the clock for the DS0 channels from the following clock sources:

- An internal oscillator
- Any of the WAN ports
- An external clock (accessed via the *Ext. Clock* 3-position terminal block located on the rear panel of the DACS.”

The *Main Reference* setting determines the clock source if this source is operational. If the *Main Reference* clock source fails, the *Fallback Reference* becomes the clock source to synchronize all DS0 channels. The clock source is the system clock for the entire DACS.

Configuring the System Clock Settings

The following sections describe configuring the clock settings.

Main Reference (*daxClockMainRef*)

The *Main Reference* and *Fallback Reference* parameters have the same selections for system clock. Make sure you choose different clock sources for the Main Reference and Fallback Reference. The following settings are available:

- none(0)—No clock selection.
- wan-1(1)—WAN port #1 is the clock source
- wan-2(2)—WAN port #2 is the clock source
- wan-3(3)—WAN port #3 is the clock source
- wan-4(4)—WAN port #4 is the clock source
- wan-5(5)—N/A
- wan-6(6)—N/A
- wan-7(7)—N/A
- wan-8(8)—N/A

- netref-1(101)—N/A
- netref-2(102)—N/A
- internal(200)— The internal free-running oscillator is the clock source.
- external(300)—The external clock source connected to the 3-position terminal block on the rear of the 2604 DACS is the clock source

Fallback Reference (*daxClockFallbackRef*)

The fallback reference enables the configuration of a back-up clock reference should the main reference fail. The *Main Reference* and *Fallback Reference* parameters have the same selections for system clock. Make sure you choose different clock sources for the Main Reference and Fallback Reference. The fallback reference settings are the same as those described in section “Main Reference (*daxClockMainRef*)” on page 42.

Clock Status (*daxClockFailure*)

The clock status indicates alarm conditions relating to the system clock. If there are no alarms, the Clocking page will indicate *No Alarm* (see figure 15 on page 42). If an alarm condition exists, an *Alarms Present* message will be displayed along with one of the following failure descriptions.

- no-failures(0)—No alarms present
- main-ref-fail(1)—The main clock reference has failed
- fallback-ref-fail(2)—The fall back clock reference has failed

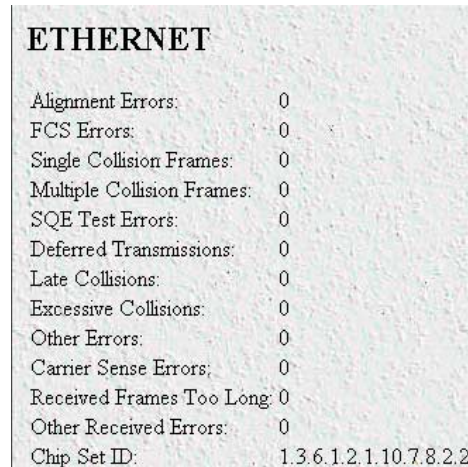
Chapter 7 Ethernet

Chapter contents

Introduction	46
Ethernet statistics.....	46
Alignment Errors (dot3StatsAlignmentErrors)	46
FCS Errors (dot3StatsFCSErrors)	46
Single Collision Frames (dot3StatsSingleCollisionFrames)	46
Multiple Collision Frames (dot3StatsMultipleCollisionFrames)	47
SQE Test Errors (dot3StatsSQETestErrors)	47
Deferred Transmissions (dot3StatsDeferredTransmissions)	47
Late Collisions (dot3StatsLateCollisions)	47
Excessive Collisions (dot3StatsExcessiveCollisions)	47
Other Errors (dot3StatsInternalMacTransmitErrors)	47
Carrier Sense Errors (dot3StatsCarrierSenseErrors)	47
Received Frames Too Long (dot3StatsFrameTooLongs)	47
Other Received Errors (dot3StatsInternalMacReceiveErrors)	48
Chip Set ID (dot3StatsEtherChipSet)	48

Introduction

The DACS provides management and statistical information in the Ethernet window (see figure 16). Most of the descriptions for these MIB variables are from RFC 1643. Detailed information regarding the SNMP MIB II variables may be downloaded from *RFC 1643, Definitions of Managed Objects for the Ethernet-like Interface Types*.



ETHERNET	
Alignment Errors:	0
FCS Errors:	0
Single Collision Frames:	0
Multiple Collision Frames:	0
SQE Test Errors:	0
Deferred Transmissions:	0
Late Collisions:	0
Excessive Collisions:	0
Other Errors:	0
Carrier Sense Errors:	0
Received Frames Too Long:	0
Other Received Errors:	0
Chip Set ID:	1.3.6.1.2.1.10.7.8.2.2

Figure 16. Ethernet window

Click on Ethernet under the Configuration Menu to monitor Ethernet statistics.

Ethernet statistics

Alignment Errors (*dot3StatsAlignmentErrors*)

A count of frames received that are not an integral number of octets in length and do not pass the FCS check. The count represented by an instance of this object is incremented when the alignmentError status is returned by the MAC service to the LLC (or other MAC user). Received frames for which multiple error conditions obtain are, according to the conventions of IEEE 802.3 Layer Management, counted exclusively according to the error status presented to the LLC.

FCS Errors (*dot3StatsFCSErrors*)

A count of frames received that are an integral number of octets in length but do not pass the FCS check. The count represented by an instance of this object is incremented when the frameCheckError status is returned by the MAC service to the LLC (or other MAC user). Received frames for which multiple error conditions obtain are, according to the conventions of IEEE 802.3 Layer Management, counted exclusively according to the error status presented to the LLC."

Single Collision Frames (*dot3StatsSingleCollision Frames*)

A count of successfully transmitted frames for which transmission is inhibited by exactly one collision. A frame that is counted by an instance of this object is also counted by the corresponding instance of either the

ifOutUcastPkts, ifOutMulticastPkts, or ifOutBroadcastPkts, and is not counted by the corresponding instance of the dot3StatsMultipleCollisionFrames object."

Multiple Collision Frames (dot3StatsMultipleCollisionFrames)

The number of successfully transmitted frames for which transmission is inhibited by more than one collision. A frame that is counted by an instance of this object is also counted by the corresponding instance of either the ifOutUcastPkts, ifOutMulticastPkts, or ifOutBroadcastPkts, and is not counted by the corresponding instance of the dot3StatsSingleCollisionFrames object."

SQE Test Errors (dot3StatsSQETestErrors)

A count of times that the SQE TEST ERROR message is generated by the PLS sublayer. The SQE TEST ERROR message is defined in section 7.2.2.2.4 of ANSI/IEEE 802.3-1985 and its generation is described in section 7.2.4.6 of the same document."

Deferred Transmissions (dot3StatsDeferredTransmissions)

The number of times for which the first transmission attempt is delayed because the medium is busy. This number does not include frames involved in collisions.

Late Collisions (dot3StatsLateCollisions)

The number of times that a collision is detected later than 512 bit-times into the transmission of a packet. Five hundred and twelve bit-times corresponds to 51.2 microseconds on a 10 Mbps system. A (late) collision included in a count of late collisions is also considered as a (generic) collision for purposes of other collision-related statistics.

Excessive Collisions (dot3StatsExcessiveCollisions)

The number of frames in which transmission failed due to excessive collisions.

Other Errors (dot3StatsInternalMacTransmitErrors)

The number of frames for which transmission fails due to an internal MAC sublayer transmit error. A frame is only counted if it is not counted by the corresponding instance of either the dot3StatsLateCollisions object, the dot3StatsExcessiveCollisions object, or the dot3StatsCarrierSenseErrors object. The precise meaning of the count represented by an instance of this object is implementation-specific. In particular, an instance of this object may represent a count of transmission errors on a particular interface that are not otherwise counted.

Carrier Sense Errors (dot3StatsCarrierSenseErrors)

The number of times that the carrier sense condition was lost or never asserted when attempting to transmit a frame. The is incremented at most once per transmission attempt, even if the carrier sense condition fluctuates during a transmission attempt."

Received Frames Too Long (dot3StatsFrameTooLongs)

The number of frames received that exceed the maximum permitted frame size. The count is incremented when the frameTooLong status is returned by the MAC service to the LLC (or other MAC user). Received frames for which multiple error conditions obtain are, according to the conventions of IEEE 802.3 Layer Management, counted exclusively according to the error status presented to the LLC."

Other Received Errors (*dot3StatsInternalMacReceiveErrors*)

The number of frames in which reception fails due to an internal MAC sublayer receive error. A frame is only counted if it is not counted by either the *dot3StatsFrameTooLongs* object, the *dot3StatsAlignmentErrors* object, or the *dot3StatsFCSErrors* object. The precise meaning of the count represented by an instance of this object is implementation-specific. In particular, an instance of this object may represent a count of receive errors on a particular interface that are not otherwise counted."

Chip Set ID (*dot3StatsEtherChipSet*)

Identifies the chipset to implement the Ethernet interface. The chipset ID identifies the chipset which gathers the transmit and receive statistics and error indications.

Chapter 8 **Filter IP**

Chapter contents

Introduction	50
Defining a filter	50
Name (filterIpName)	52
Direction (filterIpDirection)	52
Action (filterIpAction)	52
Source IP (filterIpSourceIp)	52
Source IP Mask (filterIpSourceMask)	52
Destination IP (filterIpDestinationIp)	52
Destination Mask (filterIpDestinationMask)	53
Source Port (FilterIpSourcePort)	53
Action (filterIpSourcePortCmp)	53
Destination Port (filterIpDestinationPort)	53
Action (filterIpDestinationPortCmp)	53
Protocol (filterIpProtocol)	53
TCP Established (filterIpTcpEstablished)	53

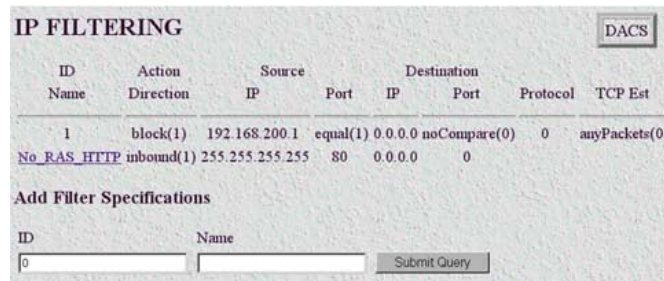
Introduction

The DACS software provides an IP filtering system that enables you to set up security for the internal management system.

Each filter is a defined list of parameters based upon attributes in the IP, TCP, and UDP headers. There are two major steps to filter creation: first defining the filter, then applying it to a user connection. The same filter can be shared by several users.

The DACS enables 20 separate filters to be defined, of which up to 10 can be used during a single user connection. Since the IP connections in the 2604 are only for the superuser and the monitor user, these will be the only two users. The application of the filters is done on the Filter IP main web page.

Click on Filter IP under the Configuration Menu to display the Filter IP main window (see figure 17).



The screenshot shows a web interface titled "IP FILTERING" with a "DACS" button in the top right. Below the title is a table with columns: ID, Action, Source, Destination, Protocol, and TCP Est. The table contains one row with the following data: ID: 1, Action: block(1), Source IP: 192.168.200.1, Source Port: equal(1), Destination IP: 0.0.0.0, Destination Port: noCompare(0), Protocol: 0, TCP Est: anyPackets(0). Below the table is a section titled "Add Filter Specifications" with two input fields labeled "ID" and "Name", and a "Submit Query" button.

ID	Action	Source	Destination	Protocol	TCP Est		
Name	Direction	IP	Port	IP	Port		
1	block(1)	192.168.200.1	equal(1)	0.0.0.0	noCompare(0)	0	anyPackets(0)
No_RAS_HTTP	inbound(1)	255.255.255.255	80	0.0.0.0	0		

Figure 17. Filter IP main window

Defining a filter

This section gives a brief summary on defining a filter. For a complete discussion with filter examples, see the final section.

To define a new filter, select an ID number and a name, then click on the **Submit Query** button to submit the request. The ID number and name must not already exist in the IP FILTER list, and the number must be an integer between 1 and 20. To delete a filter, enter just the ID number without a name and click on the **Submit Query** button.

Note Block filters take priority, therefore any applied and matching block filters will drop the packet. Next, pass filters are examined, if PASS filters have been defined, then at least one of them must match or else the packet will be dropped. After the block and pass filters are examined, the WRAP filter, if it exists, will be applied.

After entering an ID number and name, click on the name of the filter to display the filter parameters window (see figure 18).

Figure 18. Filter IP parameters window

The configurable filter parameters are :

- Name (filterIpName)
- Direction (filterIpDirection)
- Action (filterIpAction)
- Source IP (filterIpSourceIp)
- Source IP Mask (filterIpSourceMask)
- Destination IP (filterIpDestinationIp)
- Destination Mask (filterIpDestinationMask)
- Source Port (FilterIpSourcePort)
- Action (filterIpSourcePortCmp)
- Destination Port (filterIpDestinationPort)
- Action (filterIpDestinationPortCmp)
- Protocol (filterIpProtocol)
- TCP Established (filterIpTcpEstablished)

Note Any changes to a filter take effect immediately upon clicking **Submit Query**. This can aid in troubleshooting a filter profile while the user is online.

The following sections provide detailed descriptions of the configurable filter IP parameters.

Name (*filterIpName*)

This is the name of the filter

Direction (*filterIpDirection*)

Specifies the direction of the filter (that is, whether it applies to data packets inbound or outbound from the DACS). The filter only applies to the *Superuser* and the *Monitor Users* through the Ethernet interface. (Since the DACS and WAN connections function as a transparent pipe, neither of the two users can utilize these interfaces, only the Ethernet interface.) The following options are available:

- inactive(0)—Disables filter operation
- inbound(1)—Relates to packets coming into the DACS
- outbound(2)—Relates to packets leaving the DACS
- both(3)—Specifies both inbound and outbound operation

Action (*filterIpAction*)

Specifies the action to effect the packet. The action decides whether to block or pass the packet. The following options are available:

- pass(0)—If pass is selected, checking will continue on to other filters until either a match occurs, a block occurs, or there are no more filters remaining to check.

Note If there are any applied PASS filters, then at least one of them must match or the packet will be dropped.

- block(1)—If a filter has block set and the filter matches the block, the packet is discarded and no further processing is done.
- wrap(2)—All packets received on the specified link will be encapsulated in an extra IP header as defined in RFC2003. The destination IP address of the wrapper is given by the destination IP setting in the filter. The source IP address of the wrapper is the ethernet address of the DACS.

All wrap filters are inbound only.

Source IP (*filterIpSourceIp*)

This is the Source IP address in the IP header, it is used when comparing a packet's source address.

Source IP Mask (*filterIpSourceMask*)

This is the Source IP Mask (*filterIpSourceMask*) used when comparing a packet's source address. Bit positions that are set to 1 will be compared and 0's will be ignored. Thus, a setting of 0.0.0.0 will have the effect of disabling source IP address comparison.

Destination IP (*filterIpDestinationIp*)

This is the destination IP address in the IP header used when comparing a packet's destination address.

Destination Mask (*filterIpDestinationMask*)

This is the destination mask used when comparing a packet's destination address. Bit positions that are set to 1 will be compared and 0's will be ignored. Thus, a setting of 0.0.0.0 will have the effect of disabling destination IP address comparison.

Source Port (*filterIpSourcePort*)

Specifies the source port number (TCP or UDP) that the access server DACS compares. The source port Action (see Action (*filterIpSourcePortCmp*) next) action will determine how the source port is treated, whether the source port in the IP packet is not compared, equal, less than, or greater than the Source Port designated in the filter.

Action (*filterIpSourcePortCmp*)

Specifies the Action (*filterIpSourcePortCmp*) that the DACS compares. The source port action determines whether the source port in the IP packet is not compared, equal, less than, or greater than the Source Port designated in the filter.

- noCompare(0) – No Comparison to the source port in the IP packet.
- equal(1)—The port in the source IP packet is the same
- lessThan(2)—The port in the source IP packet is less than
- greaterThan(3)—The port in the source IP packet is greater than

Destination Port (*filterIpDestinationPort*)

Specifies the destination port number which the DACS compares. The destination action functions similarly to the Source Port and its Action defined above.

Action (*filterIpDestinationPortCmp*)

Specifies the action (TCP or UDP) which the DACS compares. The destination action will determine how the destination port is treated.

- noCompare(0)—No Comparison to the destination port in the IP packet.
- equal(1)—The port in the destination IP packet is the same
- lessThan(2)—The port in the destination IP packet is less than
- greaterThan(3)—The port in the destination IP packet is greater than

Protocol (*filterIpProtocol*)

Specifies the IP Protocol number to use for filtering. Some examples of protocol numbers are 1 for ICMP; 6 for TCP; and 17 for UDP. A list of protocol numbers can be found in RFC 1340. A setting of 0 disables processing based on protocol number.

TCP Established (*filterIpTcpEstablished*)

Specifies whether the filter should match only those packets which indicate in the TCP header flags that the connection is established. The following choices are available:

- anyPackets(0)—Applies the filter to all packets
- onlyEstablishedConnections(1)—Only applies the filter to established TCP connections

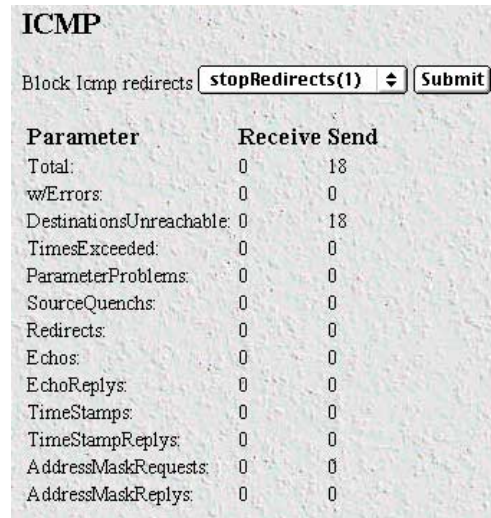
Chapter 9 ICMP

Chapter contents

Introduction	56
Block ICMP redirects (boxBLockIcmpRedirects)	56
ICMP Receive/Send Messages window	56
Total Received (icmpInMsgs)	56
Total Sent [icmpOutMsgs]	57
w/Errors (icmpInErrors, icmpOutErrors)	57
wo/Errors [icmpOutErrors]	57
Destinations Unreachable (IcmpInDestUnreachs, IcmpOutDestUnreachs)	57
Times Exceeded (icmpInTimeExcds, icmpOutTimeExcds)	57
Parameter Problems (icmpInParmProbs, icmpOutParmProbs)	57
Source Quenches (icmpInSrcQuenchs, icmpOutSrcQuenchs)	57
Redirects (icmpInRedirects, icmpOutRedirects)	58
Echos (icmpInEchos, icmpOutEchos)	58
Echo Replys (icmpInReps, icmpOutReps)	58
Time Stamps (icmpInTimestamps, icmpInTimestamps)	58
Time Stamp Replys (icmpInTimestampsReps) (icmpOutTimestampsReps)	58
Address Mask Requests (icmpInAddrMasks) (icmpOutAddrMasks)	58
Address Mask Replys (icmpInAddrMasksReps) (icmpOutAddrMasksReps)	58

Introduction

When networking problems or undesirable conditions occur, the ICMP protocol is used for communicating control or error information plus testing. The statistics listed on the DACS ICMP window (see figure 19) comprise those contained in RFC 792—Internet Control Message Protocol (ICMP). Implementation of the ICMP group is mandatory for all TCP/IP networks. RFC 1312—ICMP Group of MIB-II Variables—provides the definitions of these variables. It is important to remember that any RFC can be superseded by a newer.



The screenshot shows the DACS ICMP configuration window. At the top, there is a dropdown menu for 'Block Icmp redirects' set to 'stopRedirects(1)' and a 'Submit' button. Below this is a table with two columns: 'Parameter' and 'Receive Send'.

Parameter	Receive	Send
Total:	0	18
wErrors:	0	0
DestinationsUnreachable:	0	18
TimesExceeded:	0	0
ParameterProblems:	0	0
SourceQuenches:	0	0
Redirects:	0	0
Echos:	0	0
EchoReplies:	0	0
TimeStamps:	0	0
TimeStampReplies:	0	0
AddressMaskRequests:	0	0
AddressMaskReplies:	0	0

Figure 19. ICMP window

Click on ICMP under the Configuration Menu to monitor DACS ICMP statistics.

Block ICMP redirects (boxBlockIcmpRedirects)

The two options for “Block ICMP Redirects” either allow the reception of ICMP Redirect messages [allowredirects(0)] or block the reception of ICMP Redirect messages [stopredirects(1)]. The recommended configuration is to block the ICMP redirect messages because in some instances they could alter the routing table with undesirable effects, which is considered a breach of security.

ICMP Receive/Send Messages window

The ICMP window displays the ICMP message counters. ICMP messages are displayed in the window as columns comprising two types of messages:

- Messages received by the DACS (InMibVariable)
- Messages sent by the DACS (OutMibVariable)

The numbers following the parameters can be a good source of what is happening on the network to point out potential problems. Both gateways (routers) and hosts can send ICMP messages.

Total Received (icmpInMsgs)

The total number of ICMP messages which the 2604 DACS has received. Note that this counter includes all those counted by icmpInErrors (see “w/Errors (icmpInErrors, icmpOutErrors)” on page 57).

Total Sent [icmpOutMsgs]

Similar to icmpInMsgs, Total Sent represents the total number of ICMP messages which the 2604 has attempted to send. This variable includes all ICMP messages counted by icmpOutErrors (see “wo/Errors [icmpOutErrors]”).

w/Errors (icmplnErrors, icmpOutErrors)

The number of ICMP messages which the Model 2604 received/sent but having ICMP-specific errors (for example, bad ICMP checksums, bad length, or non-routable errors).

wo/Errors [icmpOutErrors]

The number of ICMP messages which the Model 2604 did not send due to problems discovered within ICMP such as a lack of buffers. It does not include errors discovered outside the ICMP layer like the inability of IP to route the resultant datagram.

Destinations Unreachable (IcmlnDestUnreachs, IcmpOutDestUnreachs)

The number of ICMP destination unreachable messages received/sent. For instance, if the information in a gateway's routing table determines that the network specified in a packet is unreachable, the gateway will send back an ICMP message stating that the network is unreachable. The following conditions will send back an unreachable message:

- The network is unreachable
- The host is unreachable
- The protocol is not available to the network
- The port on the host is unavailable. a specified source route failed
- A packet must be fragmented (that is, broken up into two or more packets) but the packet was sent anyway with instructions not to be fragmented.

Times Exceeded (icmplnTimeExcds, icmpOutTimeExcds)

The number of ICMP time exceeded messages received/sent. Each time a packet passes through a gateway, that gateway reduces the time-to-live (TTL) field by one. The default starting number is defined under the IP section. If the gateway processing a packet finds that the TTL field is zero it will discard the packet and send the ICMP time exceeded message. Time exceeded will also be incremented when a host which is reassembling a fragmented packet cannot complete the reassembly due to missing packets within its time limit. In this case, ICMP will discard the packet and send the time exceeded message.

Parameter Problems (icmplnParmProbs, icmpOutParmProbs)

The number of ICMP parameter problem messages received/sent. If while processing a packet, a gateway or host finds a problem with one or more of the IP header parameters which prohibits further processing, the gateway or host will discard the packet and return an ICMP parameter problem message. One potential source of this problem may be with incorrect or invalid arguments in an option. ICMP sends the parameter problems message if the gateway or host has discarded the whole packet.

Source Quenches (icmplnSrcQuenchs, icmpOutSrcQuenchs)

The number of ICMP source quench messages received/sent. A gateway will discard packets if it cannot allocate the resources, such as buffer space, to process the packet. If a gateway discards the packet, it will send an

ICMP source quench message back to the sending device. A host may send this messages if packets arrive too fast to be processed or if there is network congestion. The source quench message is a request to reduce the rate at which the source is sending traffic. If the DACS receives a source quench, it will wait for acknowledgement of all outstanding packets before sending more packets to the remote destination. Then it will begin sending out packets at an increasing rate until the connection is restored to standard operating conditions.

Redirects (icmpInRedirects, icmpOutRedirects)

The number of ICMP redirect messages received/sent. A gateway sends a redirect message to a host if the network gateways find a shorter route to the destination through another gateway.

Echos (icmpInEchos, icmpOutEchos)

The number of ICMP echo request messages received/send. The ICMP echo is used whenever one uses the diagnostic tool ping. Ping is used to test connectivity with a remote host by sending regular ICMP echo request packets and then waiting for a reply. Received echos (icmpInEchos) will increment when the DACS is pinged.

Echo Replies (icmpInReps, icmpOutReps)

The number of ICMP echo reply messages received/sent. An echo reply is a response to an echo request. Send echos (icmpOutEchos) will increment when the DACS sends an echo reply message in response to a ping.

Time Stamps (icmpInTimestamps, icmpInTimestamps)

The number of ICMP time stamp messages received/sent. Time stamp and time stamp replies were originally designed into the ICMP facility to allow network clock synchronization. Subsequently, a new protocol—Network time protocol (NTP) has taken over this function. Normally, this number will be zero.

Time Stamp Replies (icmpInTimestampsReps) (icmpOutTimestampsReps)

The number of ICMP timestamp reply messages received/sent. This message is part of a time stamp (see “Time Stamps (icmpInTimestamps, icmpInTimestamps)”) request. Normally, this number will be zero.

Address Mask Requests (icmpInAddrMasks) (icmpOutAddrMasks)

The number of ICMP address mask request messages received/sent. this message is generally used for diskless workstations which use this request at boot time to obtain their subnet mask. This number will increase if there are hosts on the network which broadcast these requests.

Address Mask Replies (icmpInAddrMasksReps) (icmpOutAddrMasksReps)

The number of ICMP address mask reply messages received/sent. Normally, this number will be zero.

Chapter 10 IP

Chapter contents

Introduction	61
IP main window	61
Forwarding (ipForwarding)	62
Default Time-To-Live (ipDefaultTTL)	62
Total Datagrams Received (ipInReceives)	62
Discarded for Header Errors (ipInHdrErrors)	62
Discarded for Address Errors (ipInAddrErrors)	62
Forwarded Datagrams (ipForwDatagrams)	62
Discarded for Unknown Protos (ipInUnknownProtos)	62
Discarded w/No Errors (ipInDiscards)	62
Total Deliveries (ipInDelivers)	63
Out Requests (ipOutRequests)	63
Out Discards (ipOutDiscards)	63
Discarded for No Routes (ipOutNoRoutes)	63
Reassembly Timeout (ipReasmTimeout)	63
# of Reassembled Fragments (ipReasmReqds)	63
# Successfully Reassembled (ipReasmOKs)	63
Reassembly Failures (ipReasmFails)	63
# Fragmented OK (ipFragOKs)	64
# Fragmented Failed (ipFragFails)	64
# Fragments Created (ipFragCreates)	64
# Valid but Discarded (ipRoutingDiscards)	64
Modify	64
Forwarding (ipForwarding)	64
Default Time-To-Live (ipDefaultTTL)	64
Addressing Information	65
IP addressing Information Details	65
Entry Interface Index (ipAdEntIfIndex)	65
Entry Subnet Mask (ipAdEntNetMask)	65
Entry Broadcast Address (ipAdEntBcastAddr)	65
Entry Reassembly Maximum Size (ipAdEntReasmMaxSize)	65
Routing Information	66
Destination (ipRouteDest)	66
Mask (ipRouteMask)	67
Gateway (RouteGateway)	67
Cost (RouteCost)	67
Interface (ipRouteIfIndex)	67
State (RouteState)	67
Add a route:	67

Advanced...	67
O/S forwarding table window	68
Destination (ipRouteDest)	68
Mask (ipRouteMask)	68
Next Hop (ipRouteNextHop)	68
Interface (ipRouteIfIndex)	68
Type (ipRouteType)	68
Protocol (ipRouteProto)	69
Info (ipRouteInfo)	69
IP Routing Destination window	70
Route Destination (ipRouteDest)	70
Mask (ipRouteMask)	70
Interface (ipRouteIfIndex)	70
Protocol (ipRouteProto)	70
Seconds Since Updated (ipRouteAge)	71
Tag (RouteTag)	71
Gateway (RouteGateway)	71
Cost (RouteCost)	71
State (RouteState)	71
Address Translation Information	71
Interface (ipNetToMediaEntry)	71
Net Address (ipNetToMediaNetAddress)	72
Physical (ipNetToMediaPhysAddress)	72
Type (ipNetToMediaType)	72

Introduction

The IP (Internet Protocol) window lists IP statistics and parameters, and enables you to modify IP settings.

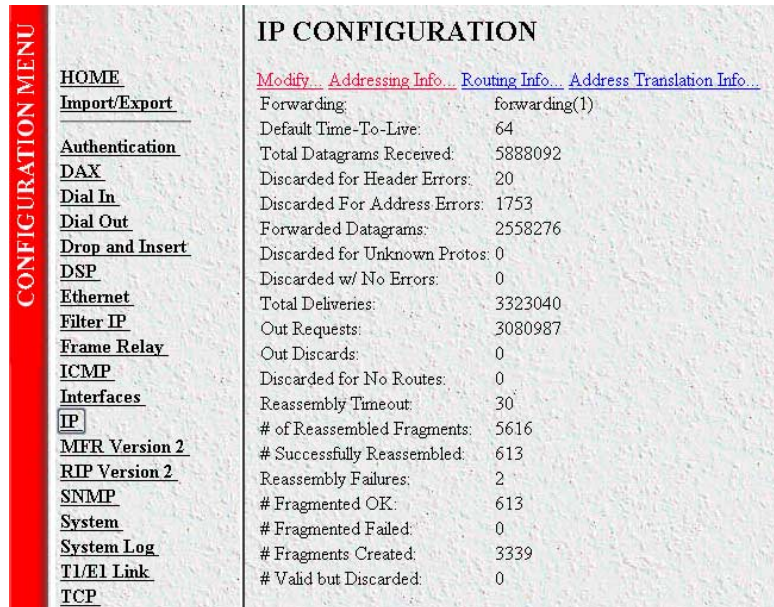


Figure 20. IP main window

Click on IP under the Configuration Menu to display the IP window.

IP main window

All items described in this chapter are defined in *RFC 1213: Management Information Base for Network Management of TCP/IP-based internets: MIB-II*.

The IP main window contains basic IP configuration parameters and statistics, and it has the following links to windows that will enable you to modify IP parameters:

- **Modify**—This window is where you can modify forwarding and time-to-live settings (see “Modify” on page 64).
- **Addressing Info**—This window (see “Addressing Information” on page 65) displays IP addressing details for the default address for outgoing IP datagrams, the local or loopback address of the box and the IP address of the box as defined in Chapter 15, “System”.
- **Routing Info**—This window displays routing information for routing IP datagrams (the IP address, subnet mask, next hop router, and interface for each network interface defined in the box) (see “Routing Information” on page 66).
- **Address Translation Info**—The IP address translation table contains the IP address to physical address equivalences (see “Address Translation Information” on page 71).

Forwarding (*ipForwarding*)

The indication of whether this entity is acting as an IP gateway in respect to the forwarding of datagrams received by, but not addressed to, this entity. IP gateways forward datagrams, IP hosts do not (except those source-routed via the host).

Note For some managed nodes, this object may take on only a subset of the values possible. Accordingly, it is appropriate for an agent to return a “badValue” response if a management station attempts to change this object to an inappropriate value.

The following conditions can be displayed:

- forwarding(1)—acting as a gateway and will forward IP datagrams to other gateways
- not-forwarding(2)—*not* acting as a gateway so it will discard IP datagrams destined for other gateways

Default Time-To-Live (*ipDefaultTTL*)

The default value inserted into the time-to-live field of the IP header of datagrams originated at this entity, whenever a TTL value is not supplied by the transport layer protocol.

Total Datagrams Received (*ipInReceives*)

The total number of input datagrams received from interfaces, including those received in error.

Discarded for Header Errors (*ipInHdrErrors*)

The number of input datagrams discarded due to errors in their IP headers, including bad checksums, version number mismatch, other format errors, time-to-live exceeded, errors discovered in processing their IP options, and so on.

Discarded for Address Errors (*ipInAddrErrors*)

The number of input datagrams discarded because the IP address in their IP header's destination field was not a valid address to be received at this entity. This count includes invalid addresses (e.g., 0.0.0.0) and addresses of unsupported Classes (e.g., Class E). For entities which are not IP Gateways and therefore do not forward datagrams, this counter includes datagrams discarded because the destination address was not a local address.

Forwarded Datagrams (*ipForwDatagrams*)

The number of input datagrams for which this entity was not their final IP destination, as a result of which an attempt was made to find a route to forward them to that final destination. In entities which do not act as IP Gateways, this counter will include only those packets which were source-routed via this entity, and the source-route option processing was successful.

Discarded for Unknown Protos (*ipInUnknownProtos*)

The number of locally-addressed datagrams received successfully but discarded because of an unknown or unsupported protocol.

Discarded w/No Errors (*ipInDiscards*)

The number of input IP datagrams for which no problems were encountered to prevent their continued processing, but which were discarded (for example, due to lack of buffer space).

Note The Discarded w/No Errors counter does not include any datagrams discarded while awaiting re-assembly.

Total Deliveries (*ipInDelivers*)

The total number of input datagrams successfully delivered to IP user-protocols (including ICMP).

Out Requests (*ipOutRequests*)

The total number of IP datagrams which local IP user-protocols (including ICMP) supplied to IP in requests for transmission.

Note The Out Requests counter does not include any datagrams counted in `ipForwDatagrams`.

Out Discards (*ipOutDiscards*)

The number of output IP datagrams for which no problem was encountered to prevent their transmission to their destination, but which were discarded (e.g., for lack of buffer space).

Note The Out Discards counter would include datagrams counted in `ipForwDatagrams` if any such packets met this (discretionary) discard criterion.

Discarded for No Routes (*ipOutNoRoutes*)

The number of IP datagrams discarded because no route could be found to transmit them to their destination.

Note The Discarded for No Routes counter includes any packets counted in `ipForwDatagrams` which meet this “no-route” criterion. This includes any datagrams which a host cannot route because all of its default gateways are down.

Reassembly Timeout (*ipReasmTimeout*)

The maximum number of seconds which received fragments are held while they are awaiting reassembly at this entity.

of Reassembled Fragments (*ipReasmReqds*)

The number of IP fragments received which needed to be reassembled at this entity.

Successfully Reassembled (*ipReasmOKs*)

The number of IP datagrams successfully reassembled.

Reassembly Failures (*ipReasmFails*)

The number of failures detected by the IP reassembly algorithm (for whatever reason: timed out, errors, etc).

Note The Reassembly Failures value is not necessarily a count of discarded IP fragments since some algorithms (notably the algorithm in RFC 815) can lose track of the number of fragments by combining them as they are received.

Fragmented OK (*ipFragOKs*)

The number of IP datagrams that have been successfully fragmented at this entity.

Fragmented Failed (*ipFragFails*)

The number of IP datagrams that have been discarded because they required fragmenting at this entity, but were not fragmented because their *Don't Fragment* option was set.

Fragments Created (*ipFragCreates*)

The number of IP datagram fragments that have been generated at this entity.

Valid but Discarded (*ipRoutingDiscards*)

The number of routing entries which were chosen to be discarded even though they are valid. One possible reason for discarding such an entry could be to make more buffer space available for other routing entries.

Modify

The Modify IP configuration window (see figure 21) is where you can change IP Forwarding and Default Time-to-Live parameters.

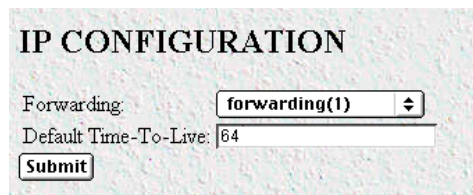


Figure 21. IP configurations modification window

Forwarding (*ipForwarding*)

Determines whether this entity is acting as an IP gateway that will forward datagrams received by—but not addressed to—this entity. IP gateways forward datagrams, IP hosts do not (except those source-routed via the host).

Note For some managed nodes, this object may take on only a subset of the values possible. Accordingly, it is appropriate for an agent to return a "badValue" response if a management station attempts to change this object to an inappropriate value.

The following options are available:

- forwarding(1)—acting as a gateway
- not-forwarding(2)—*not* acting as a gateway

Default Time-To-Live (*ipDefaultTTL*)

The default value inserted into the Time-To-Live (TTL) field in the IP header of datagrams originating from this entity, whenever a TTL value is not already supplied by the transport layer protocol.

Addressing Information

The IP addressing Information window (see figure 22) is where you can view the default address for outgoing IP datagrams, the local or loopback address of the box, and the IP address of the box as defined in Chapter 15, “System”.

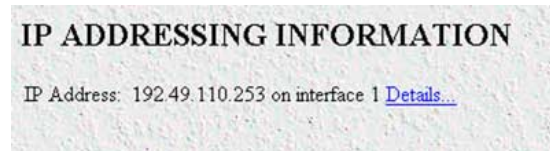


Figure 22. IP addressing Information window

Click on the Details link to display IP address Table entries for each defined network interface (see “IP addressing Information Details”.

IP addressing Information Details

This window (see figure 23) shows IP address Table entries for each defined network interface.

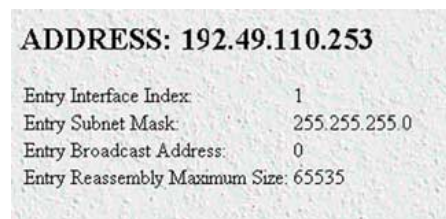


Figure 23. IP addressing Details window

Entry Interface Index (ipAdEntIfIndex)

The index value that identifies the interface to which this entry applies.

Entry Subnet Mask (ipAdEntNetMask)

The subnet mask associated with the IP address of this entry. The value of the mask is an IP address with all the network bits set to 1 and all the hosts bits set to 0.

Entry Broadcast Address (ipAdEntBcastAddr)

The value of the least-significant bit in the IP broadcast address used for sending datagrams on the interface associated with the IP address of this entry. For example, when the Internet standard all-ones broadcast address is used, the value will be 1. This value applies to both the subnet and network broadcast addresses used by the entity on this interface.

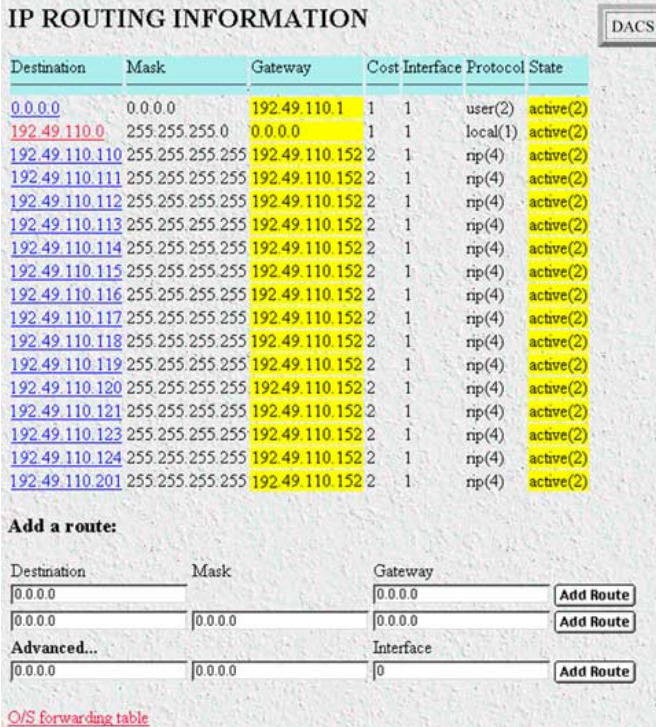
Entry Reassembly Maximum Size (ipAdEntReasmMaxSize)

The size of the largest IP datagram which this entity can re-assemble from incoming IP fragmented datagrams received on this interface.

Routing Information

The IP Routing Information window (see figure 24) displays information required to route IP datagrams, including the IP address, subnet mask, next-hop router, and interface for each network interface defined in the DACS.

The IP Routing Information window also has a link to the O/S forwarding table where the forwarding parameters are displayed (“O/S forwarding table window” on page 68).



The screenshot shows the 'IP ROUTING INFORMATION' window with a 'DACS' button in the top right. The main table lists routes with columns for Destination, Mask, Gateway, Cost, Interface, Protocol, and State. Below the table is an 'Add a route:' form with input fields for Destination, Mask, Gateway, and Interface, each with an 'Add Route' button. A link for 'O/S forwarding table' is at the bottom left.

Destination	Mask	Gateway	Cost	Interface	Protocol	State
0.0.0.0	0.0.0.0	192.49.110.1	1	1	user(2)	active(2)
192.49.110.0	255.255.255.0	0.0.0.0	1	1	local(1)	active(2)
192.49.110.110	255.255.255.255	192.49.110.152	2	1	rip(4)	active(2)
192.49.110.111	255.255.255.255	192.49.110.152	2	1	rip(4)	active(2)
192.49.110.112	255.255.255.255	192.49.110.152	2	1	rip(4)	active(2)
192.49.110.113	255.255.255.255	192.49.110.152	2	1	rip(4)	active(2)
192.49.110.114	255.255.255.255	192.49.110.152	2	1	rip(4)	active(2)
192.49.110.115	255.255.255.255	192.49.110.152	2	1	rip(4)	active(2)
192.49.110.116	255.255.255.255	192.49.110.152	2	1	rip(4)	active(2)
192.49.110.117	255.255.255.255	192.49.110.152	2	1	rip(4)	active(2)
192.49.110.118	255.255.255.255	192.49.110.152	2	1	rip(4)	active(2)
192.49.110.119	255.255.255.255	192.49.110.152	2	1	rip(4)	active(2)
192.49.110.120	255.255.255.255	192.49.110.152	2	1	rip(4)	active(2)
192.49.110.121	255.255.255.255	192.49.110.152	2	1	rip(4)	active(2)
192.49.110.123	255.255.255.255	192.49.110.152	2	1	rip(4)	active(2)
192.49.110.124	255.255.255.255	192.49.110.152	2	1	rip(4)	active(2)
192.49.110.201	255.255.255.255	192.49.110.152	2	1	rip(4)	active(2)

Add a route:

Destination	Mask	Gateway	
<input type="text" value="0.0.0.0"/>		<input type="text" value="0.0.0.0"/>	<input type="button" value="Add Route"/>
<input type="text" value="0.0.0.0"/>	<input type="text" value="0.0.0.0"/>	<input type="text" value="0.0.0.0"/>	<input type="button" value="Add Route"/>
Advanced...		Interface	
<input type="text" value="0.0.0.0"/>	<input type="text" value="0.0.0.0"/>	<input type="text" value="0"/>	<input type="button" value="Add Route"/>

[O/S forwarding table](#)

Figure 24. IP Routing Information window

Destination (*ipRouteDest*)

The destination IP address of this route. An entry with a value of 0.0.0.0 is considered a default route. Multiple routes to a single destination can appear in the table, but access to such multiple entries is dependent on the table-access mechanisms defined by the network management protocol in use.

To view or modify next-hop routing information for each destination, click on a destination link in the Destination column. For more information about modifying next-hop routing information settings, refer to “IP Routing Destination window” on page 70.

Mask (*ipRouteMask*)

Indicates the mask to be logical-ANDed with the destination address before being compared to the value in the *ipRouteDest* field. For those systems that do not support arbitrary subnet masks, an agent constructs the value of the *ipRouteMask* by determining whether the value of the corresponding *ipRouteDest* field belongs to a Class A, B, or C network, and then using the appropriate mask from table 3.

Table 3. Masks

Mask	Network
255.0.0.0	class-A
255.255.0.0	class-B
255.255.255.0	class-C

Gateway (*RouteGateway*)

Specifies the IP address to which the packets should be forwarded.

Cost (*RouteCost*)

This is the cost of the route as defined by RIP standards. Cost is sometimes considered to be number of hops. A cost of 16 is considered to be infinite. A cost can be given to user-entered routes so their preference in relation to learned routes can be calculated.

Interface (*ipRouteIfIndex*)

The index value that identifies the local interface through which the next hop of this route should be reached. The interface identified by a particular value of this index is the same interface as identified by the same value of *ifIndex*.

State (*RouteState*)

- *invalid(1)*—This setting deletes the route.
- *active(2)*—A valid route is in use.
- *nopath(3)*—No route is available to the specified gateway. The gateway is not known to local networks.
- *agedout(4)*—Invalid route (soon to be removed).
- *costly(5)*—A valid route, but not in use because of its higher cost.

Add a route:

This portion of the IP Routing Information window is where you can add a new route to the IP Routing Information table. Fill in the Destination, Mask, and Gateway information, then click **Add Route**.

Advanced...

Enables a route to be attached to an interface. Packets to a network will be routed to that interface, allowing the gateway IP address to be dynamic.

O/S forwarding table window

The O/S forwarding table window lists forwarding information for all routes.

FORWARDING TABLE						
Destination	Mask	Next Hop	Interface	Type	Proto	Info
0.0.0.0	0.0.0.0	192.49.110.1	1	indirect(4)	local(2)	0.0
192.49.110.0	255.255.255.0	0.0.0.0	1	direct(3)	local(2)	0.0
192.49.110.110	255.255.255.255	192.49.110.152	1	indirect(4)	local(2)	0.0
192.49.110.111	255.255.255.255	192.49.110.152	1	indirect(4)	local(2)	0.0
192.49.110.112	255.255.255.255	192.49.110.152	1	indirect(4)	local(2)	0.0
192.49.110.113	255.255.255.255	192.49.110.152	1	indirect(4)	local(2)	0.0
192.49.110.114	255.255.255.255	192.49.110.152	1	indirect(4)	local(2)	0.0
192.49.110.115	255.255.255.255	192.49.110.152	1	indirect(4)	local(2)	0.0
192.49.110.116	255.255.255.255	192.49.110.152	1	indirect(4)	local(2)	0.0
192.49.110.117	255.255.255.255	192.49.110.152	1	indirect(4)	local(2)	0.0
192.49.110.118	255.255.255.255	192.49.110.152	1	indirect(4)	local(2)	0.0
192.49.110.119	255.255.255.255	192.49.110.152	1	indirect(4)	local(2)	0.0
192.49.110.120	255.255.255.255	192.49.110.152	1	indirect(4)	local(2)	0.0
192.49.110.121	255.255.255.255	192.49.110.152	1	indirect(4)	local(2)	0.0
192.49.110.123	255.255.255.255	192.49.110.152	1	indirect(4)	local(2)	0.0
192.49.110.124	255.255.255.255	192.49.110.152	1	indirect(4)	local(2)	0.0
192.49.110.201	255.255.255.255	192.49.110.152	1	indirect(4)	local(2)	0.0

Figure 25. IP Routing Forwarding Table

Destination (*ipRouteDest*)

The destination IP address of this route. An entry with a value of 0.0.0.0 is considered a default route. Multiple routes to a single destination can appear in the table, but access to such multiple entries is dependent on the table-access mechanisms defined by the network management protocol in use.

Mask (*ipRouteMask*)

Indicates the mask to be logical-ANDed with the destination address before being compared to the value in the *ipRouteDest* field. For those systems that do not support arbitrary subnet masks, an agent constructs the value of the *ipRouteMask* by determining whether the value of the correspondent *ipRouteDest* field belongs to a Class A, B, or C network, and then using the appropriate mask from table 3 on page 67.

Next Hop (*ipRouteNextHop*)

The IP address of the next hop of this route. (In the case of a route bound to an interface which is realized via a broadcast media, the value of this field is the agent's IP address on that interface.)

Interface (*ipRouteIfIndex*)

The index value that identifies the local interface through which the next hop of this route should be reached. The interface identified by a particular value of this index is the same interface as identified by the same value of *ifIndex*.

Type (*ipRouteType*)

One of the following route types:

- other(1)—none of the following
- invalid(2)—an invalidated route

- `direct(3)`—route to directly connected (sub-)network
- `indirect(4)`—route to a non-local host/network/sub-network

Note The values `direct(3)` and `indirect(4)` refer to the notion of direct and indirect routing in the IP architecture. Setting this object to the value `invalid(2)` has the effect of invalidating the corresponding entry in the `ipRouteTable` object. That is, it effectively disassociates the destination identified with said entry from the route identified with said entry. It is an implementation-specific matter as to whether the agent removes an invalidated entry from the table. Accordingly, management stations must be prepared to receive tabular information from agents that corresponds to entries not currently in use. Proper interpretation of such entries requires examination of the relevant `ipRouteType` object.

Protocol (`ipRouteProto`)

The routing mechanism via which this route was learned. Inclusion of values for gateway routing protocols is not intended to imply that hosts must support those protocols.

- `unknown(0)`
- `local(1)`—Added by the DACS to support an interface. For example, adding a route for a new dial-in user.
- `user(2)`—Added by an administrator on the IP Routing Information table or via SNMP management tools.
- `dspf(3)`—Not currently implemented.
- `rip(4)`—Learned via reception of RIP packet.
- `icmp(5)`—Learned via reception of ICMP packet.
- `radius(6)`—Provided in RADIUS response packet.

Info (`ipRouteInfo`)

A reference to MIB definitions specific to the particular routing protocol which is responsible for this route, as determined by the value specified in the route's `ipRouteProto` value. If this information is not present, its value should be set to the OBJECT IDENTIFIER { 0 0 }, which is a syntactically valid object identifier, and any conformant implementation of ASN.1 and BER must be able to generate and recognize this value.

IP Routing Destination window

The IP Routing Destination window (see figure 26) shows next-hop routing information.

```

ROUTE DESTINATION: 192.49.110.0
Mask:                255.255.255.0
Interface:           1
Protocol:            local(1)
Seconds Since Updated: 508023
Tag:                 0
Gateway:             0.0.0.0
Cost:                1
State:               active(2)
  
```

Figure 26. Routing Destination window

Route Destination (*ipRouteDest*)

The destination IP address of this route. An entry with a value of 0.0.0.0 is considered a default route. Multiple routes to a single destination can appear in the table, but access to such multiple entries is dependent on the table-access mechanisms defined by the network management protocol in use.

Mask (*ipRouteMask*)

Indicates the mask to be logical-ANDed with the destination address before being compared to the value in the *ipRouteDest* field. For those systems that do not support arbitrary subnet masks, an agent constructs the value of the *ipRouteMask* by determining whether the value of the corresponding *ipRouteDest* field belongs to a Class A, B, or C network, and then using the appropriate mask from table 3 on page 67.

Interface (*ipRouteIfIndex*)

The index value which uniquely identifies the local interface through which the next hop of this route should be reached. The interface identified by a particular value of this index is the same interface as identified by the same value of *ifIndex*.

Protocol (*ipRouteProto*)

The routing mechanism via which this route was learned. Inclusion of values for gateway routing protocols is not intended to imply that hosts must support those protocols.

- unknown(0)
- local(1)—Added by the DACS to support an interface. For example, adding a route for a new dial-in user.
- user(2)—Added by an administrator on the IP Routing Information table or via SNMP management tools.
- dspf(3)—Not currently implemented.
- rip(4)—Learned via reception of RIP packet.
- icmp(5)—Learned via reception of ICMP packet.
- radius(6)—Provided in RADIUS response packet.

Seconds Since Updated (*ipRouteAge*)

The number of seconds since this route was last updated or otherwise determined to be correct.

Tag (*RouteTag*)

An identifier associated with the route. This can have different meanings depending on the protocol. For example, this gives the tag that was passed with a learned RIP route.

Gateway (*RouteGateway*)

Specifies the IP address to which the packets should be forwarded.

Cost (*RouteCost*)

This is the cost of the route as defined by RIP standards. Cost is sometimes considered to be number of hops. A cost of 16 is considered to be infinite. A cost can be given to user-entered routes so their preference in relation to learned routes can be calculated.

State (*RouteState*)

Defines the state which a route may be in during its lifetime.

- invalid(1)—This setting deletes the route.
- active(2)—A valid route is in use.
- nopath(3)—No route is available to the specified gateway. The gateway is not known to local networks.
- agedout(4)—Invalid route (soon to be removed).
- costly(5)—A valid route, but not in use because of its higher cost.

Address Translation Information

The IP address translation table window (see figure 27) contain the IP address to physical address equivalences. Some interfaces do not use translation tables for determining address equivalences (for example, DDN-X.25 uses an algorithmic method)—if all interfaces are of this type, then the Address Translation table is empty (zero entries).

Interface	Net Address	Physical	Type
1	192.49.110.1	0x00:00:0C:33:5D:48	dynamic(3) <input type="button" value="Submit"/>
1	192.49.110.34	0x00:05:02:66:FE:11	dynamic(3) <input type="button" value="Submit"/>
1	192.49.110.57	0x00:60:97:D2:06:F3	dynamic(3) <input type="button" value="Submit"/>

Add entries:

Figure 27. Address Translation Information window

Interface (*ipNetToMediaEntry*)

Each entry contains one IP address to physical address equivalence.

Net Address (*ipNetToMediaNetAddress*)

The IP address corresponding to the media-dependent physical address.

Physical (*ipNetToMediaPhysAddress*)

The media-dependent physical address.

Type (*ipNetToMediaType*)

The type of mapping. Setting this object to the value `invalid(2)` has the effect of invalidating the corresponding entry in the `ipNetToMediaTable`. That is, it effectively disassociates the interface identified with said entry from the mapping identified with said entry. It is an implementation-specific matter as to whether the agent removes an invalidated entry from the table. Accordingly, management stations must be prepared to receive tabular information from agents that corresponds to entries not currently in use. Proper interpretation of such entries requires examination of the relevant `ipNetToMediaType` object.

- `other(1)`—none of the following
- `invalid(2)`—an invalidated mapping
- `dynamic(3)`
- `static(4)`

Chapter 11 TCP

Chapter contents

Introduction	74
TCP main window	74
Retransmit-Timeout Algorithm (tcpRtoAlgorithm)	74
Retransmit-Timeout Minimum (tcpRtoMin)	74
Retransmit-Timeout Maximum (tcpRtoMax)	74
Maximum Connections (tcpMaxConn)	75
Active Opens (tcpActiveOpens)	75
Passive Opens (tcpPassiveOpens)	75
Attempt/Fails (tcpAttemptFails)	75
ESTABLISHED Resets (tcpEstabResets)	75
Current ESTABLISHED (tcpCurrEstab)	75
Total Received (tcpInSegs)	75
Total Sent (tcpOutSegs)	75
Total Retransmitted (tcpRetransSegs)	75
Total Received in Error (tcpInErrs)	75
Total Sent w/RST Flag (tcpOutRsts)	75
TCP (Details)	76
Local Port (tcpConnLocalPort)	76
Remote Address (tcpConnRemAddress)	76
Remote Port (tcpConnRemPort)	76
State (tcpConnState)	76

Introduction

Transmission Control Protocol (TCP) is in the Transport layer of the OSI model and sits on top of IP. It is one of the more widely used protocols among the TCP/IP suite. The TCP subsystem web pages of the 2604 DACS provides management and statistical information on TCP. Detailed information regarding the SNMP MIB variables may be downloaded from RFC1213: Management Information Base for Network Management of TCP/IP-based internets: MIB-II.

Click on TCP under the Configuration Menu to display the TCP main window (see figure 28) to monitor TCP statistics.

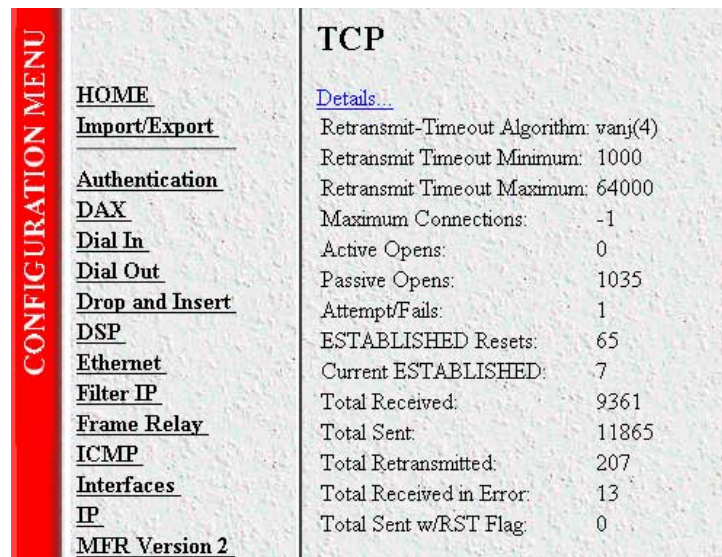


Figure 28. TCP main window

TCP main window

Retransmit-Timeout Algorithm (tcpRtoAlgorithm)

The algorithm that determines the timeout value used for retransmitting unacknowledged octets.

Retransmit-Timeout Minimum (tcpRtoMin)

The minimum value permitted by a TCP implementation for the retransmission timeout, measured in milliseconds. More refined semantics for objects of this type depend upon the algorithm used to determine the retransmission timeout. In particular, when the timeout algorithm is rsre(3), an object of this type has the semantics of the LBOUND quantity described in RFC 793.

Retransmit-Timeout Maximum (tcpRtoMax)

The maximum value permitted by a TCP implementation for the retransmission timeout, measured in milliseconds. More refined semantics for objects of this type depend upon the algorithm used to determine the retransmission timeout. In particular, when the timeout algorithm is rsre(3), an object of this type has the semantics of the UBOUND quantity described in RFC 793.

Maximum Connections (*tcpMaxConn*)

The limit on the total number of TCP connections the entity can support. In entities where the maximum number of connections is dynamic, this object should contain the value -1.

Active Opens (*tcpActiveOpens*)

The number of times TCP connections have made a direct transition to the SYN-SENT state from the CLOSED state.

Passive Opens (*tcpPassiveOpens*)

The number of times TCP connections have made a direct transition to the SYN-RCVD state from the LISTEN state.

Attempt/Fails (*tcpAttemptFails*)

The number of times TCP connections have made a direct transition to the CLOSED state from either the SYN-SENT state or the SYN-RCVD state, plus the number of times TCP connections have made a direct transition to the LISTEN state from the SYN-RCVD state.

ESTABLISHED Resets (*tcpEstabResets*)

The number of times TCP connections have made a direct transition to the CLOSED state from either the ESTABLISHED state or the CLOSE-WAIT state.

Current ESTABLISHED (*tcpCurrEstab*)

The number of TCP connections for which the current state is either ESTABLISHED or CLOSE-WAIT.

Total Received (*tcpInSegs*)

The total number of segments received, including those received in error. This count includes segments received on currently established connections.

Total Sent (*tcpOutSegs*)

The total number of segments sent, including those on current connections but excluding those containing only retransmitted octets.

Total Retransmitted (*tcpRetransSegs*)

The total number of segments retransmitted—that is, the number of TCP segments transmitted containing one or more previously transmitted octets.

Total Received in Error (*tcpInErrs*)

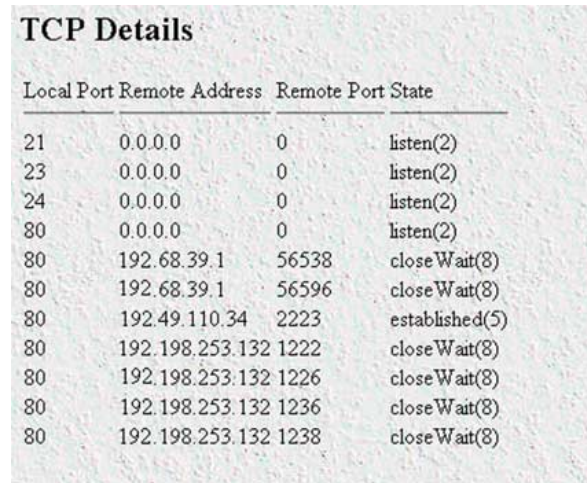
The total number of segments received in error (e.g., bad TCP checksums).

Total Sent w/RST Flag (*tcpOutRsts*)

The number of TCP segments sent containing the RST flag.

TCP (Details)

From this screen you can view port details for remote and local TCP connections (see figure 29).



Local Port	Remote Address	Remote Port	State
21	0.0.0.0	0	listen(2)
23	0.0.0.0	0	listen(2)
24	0.0.0.0	0	listen(2)
80	0.0.0.0	0	listen(2)
80	192.68.39.1	56538	closeWait(8)
80	192.68.39.1	56596	closeWait(8)
80	192.49.110.34	2223	established(5)
80	192.198.253.132	1222	closeWait(8)
80	192.198.253.132	1226	closeWait(8)
80	192.198.253.132	1236	closeWait(8)
80	192.198.253.132	1238	closeWait(8)

Figure 29. TCP Details window

Local Port (*tcpConnLocalPort*)

The local port number for this TCP connection.

Remote Address (*tcpConnRemAddress*)

The remote IP address for this TCP connection.

Remote Port (*tcpConnRemPort*)

The remote port number for this TCP connection.

State (*tcpConnState*)

The state of this TCP connection. The only value which may be set by a management station is deleteTCB(12). Accordingly, it is appropriate for an agent to return a 'badValue' response if a management station attempts to set this object to any other value.

If a management station sets this object to the value deleteTCB(12), Transmission Control Block, then this has the effect of deleting the TCB (as defined in RFC 793) of the corresponding connection on the managed node, resulting in immediate termination of the connection. As an implementation-specific option, a RST segment may be sent from the managed node to the other TCP endpoint (note however that RST segments are not sent reliably).

- closed(1)
- listen(2)
- synSent(3)
- synReceived(4)
- established(5)

- finWait1(6)
- finWait2(7)
- closeWait(8)
- lastAck(9)
- closing(10)
- timeWait(11)
- deleteTCB(12)

Chapter 12 **UDP**

Chapter contents

Introduction	80
Handling of NETBIOS UDP Broadcasts (boxNetbiosUdpBridging)	80
Received (udpInDatagrams)	80
Received With No Ports (udpNoPorts)	80
Others Received with No Delivery (udpInErrors)	80
Sent (udpOutDatagrams)	80
Listener Table (udpTable)	81
Local Address (udpLocalAddress)	81
Local Port (udpLocalPort)	81

Introduction

User Datagram Protocol (UDP) is supported by the DACS. Detailed information regarding the SNMP management information base (MIB) variables can be found in *RFC1213: Management Information Base for Network Management of TCP/IP-based internets: MIB-II*.

To manage and collect statistics on UDP, click on UDP under the Configuration Menu to display the UDP window (see figure 30).

The screenshot shows the 'UDP DATAGRAMS' configuration window. On the left is a vertical 'CONFIGURATION MENU' with links to HOME, Import/Export, Authentication, DAX, Dial In, Dial Out, Drop and Insert, DSP, Ethernet, Filter IP, Frame Relay, ICMP, Interfaces, IP, MFR Version 2, RIP Version 2, and SNMP. The main content area is titled 'UDP DATAGRAMS' and includes a 'DACs' button in the top right. Below the title, there is a section for 'Handling of NETBIOS UDP Broadcasts' with a dropdown menu set to 'doNotPassNetbiosBroadcasts(0)' and a 'Submit' button. The statistics shown are: Received: 251270, Received w/No Ports: 3019543, Others Received w/No Delivery: 0, and Sent: 2661. Below this is a 'Listener Table' with columns for 'Local Address' and 'Local Port', listing several entries with their respective addresses and ports.

Local Address	Local Port
0.0.0.0	0
0.0.0.0	161
0.0.0.0	520
0.0.0.0	581
0.0.0.0	1701
0.0.0.0	3000
192.49.110.253	513

Figure 30. UDP window

Handling of NETBIOS UDP Broadcasts (*boxNetbiosUdpBridging*)

Enables the passing of broadcast UDP packets with a port of 137 and 138 from other interfaces to the local LAN interface. Netbios uses these packets to communicate with WINS servers. A WINS server can work without this option enabled, but the remote PC will appear to be on the LAN. The following options are available:

- doNotPassNetbiosBroadcasts(0)
- passNetbiosBroadcasts(1)

Received (*udpInDatagrams*)

The total number of UDP datagrams delivered to UDP users.

Received With No Ports (*udpNoPorts*)

The total number of received UDP datagrams for which there was no application at the destination port.

Others Received with No Delivery (*udpInErrors*)

The number of received UDP datagrams that could not be delivered for reasons other than the lack of an application at the destination port.

Sent (*udpOutDatagrams*)

The total number of UDP datagrams sent from this entity.

Listener Table (udpTable)

A table containing UDP listener information.

Local Address (udpLocalAddress)

The local IP address for this UDP listener. In the case of a UDP listener that is willing to accept datagrams for any IP interface associated with the node, the value 0.0.0.0 is used.

Local Port (udpLocalPort)

The local port number for this UDP listener.

Chapter 13 RIP Version 2

Chapter contents

Introduction	84
RIP Version 2 main window.....	84
Route Changes Made (rip2GlobalRouteChanges)	84
Responses Sent (rip2GlobalQueries)	84
Adding a RIP address	84
RIP Version 2—Configuration.....	85
Address (rip2IfConfAddress)	85
Domain (rip2IfConfDomain)	86
Authentication Type (rip2IfConfAuthType)	86
Authentication Key (rip2IfConfAuthKey)	86
Send (rip2IfConfSend)	86
Receive (rip2IfConfReceive)	86
Metric (rip2IfConfDefaultMetric)	86
Status (rip2IfConfStatus)	87
RIP Version 2 (Statistics).....	87
Subnet IP Address (rip2IfStatAddress)	87
Bad Packets (rip2IfStatRcvBadPackets)	87
Bad Routes (rip2IfStatRcvBadRoutes)	87
Sent Updates (rip2IfStatSentUpdates)	87
Status (rip2IfStatStatus)	87

Introduction

The RIP Version 2 main window (see figure 31) describes routing information as defined by the Routing Information Protocol (RIP). All object identifiers described in this chapter comply with those contained in *RFC 1724: RIP Version 2 MIB Extension*.

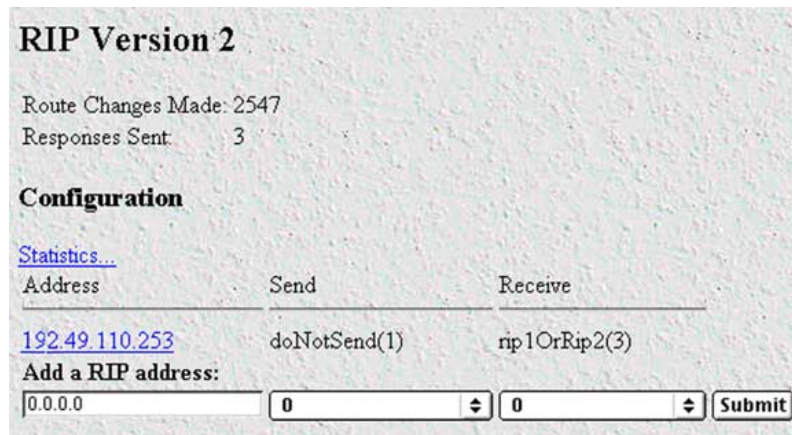


Figure 31. RIP Version 2 window

Click on RIP Version 2 under the Configuration Menu to display the RIP Version 2 main window.

RIP Version 2 main window

The RIP Version 2 window describes routing information as defined by the Routing Information Protocol (RIP). The window also contains the following links:

- **Statistics**—Clicking on the Statistics link displays the RIP Version 2 Status window (see “RIP Version 2 (Statistics)” on page 87). In this window you can view each subnet IP address, Bad Packets, Bad Routes, Sent Updates, and Status.
- **Address (xxx.xxx.xxx.xxx)**— After adding a RIP address, click on the IP Address under the Address column to display the RIP Version 2 Configuration window. You can modify the configuration here. (see “RIP Version 2 (Statistics)” on page 87).

Route Changes Made (rip2GlobalRouteChanges)

The number of route changes made to the IP Route Database by RIP. This does not include the refresh of a route's age.

Responses Sent (rip2GlobalQueries)

The number of responses sent to RIP queries from other systems.

Adding a RIP address

Do the following:

1. Enter the IP network address of the interface on the 2604 DACS that you want to enable RIP. This will be the LAN IP address, in other words, the IP address of the 2604. This is *not* the IP address of the device you want to direct RIP packets to.

2. Enter the protocol version to be used *for sending RIP packets*. The following choices are available:
 - *doNotSend (1)*
 - *ripVersion1 (2)*—ripVersion 1 implies sending RIP updates compliant with RFC 1058
 - *rip1Compatible (3)*—rip1Compatible implies broadcasting RIP-2 updates using RFC 1058 route subsumption rules
 - *ripVersion2 (4)*—ripVersion2 implies multicasting RIP-2 updates
3. Enter the protocol version to be used *for receiving RIP packets*. The following choices are available (note that rip2 and rip1OrRip2 implies reception of multicast packets.):
 - *rip1 (1)*—ripVersion 1 implies sending RIP updates compliant with RFC 1058
 - *rip2(2)*—rip1Compatible implies broadcasting RIP-2 updates using RFC 1058 route subsumption rules
 - *rip1OrRip2(3)*
 - *doNotReceive(4)*
4. Click on **Submit Query**.

Note To delete the RIP Address, click on the IP Address under the column named Address. Select Status to be invalid(2) and click on **Submit Query**.

Further modifications can be made by clicking on the Address link of the specific subnet (see “RIP Version 2—Configuration”).

RIP Version 2—Configuration

The RIP Version 2 Configuration window (see figure 32), seen by clicking on the IP Address under the column named Address, displays the RIP IP Address followed by configurable parameters. The configurable parameters are Domain, Authentication Type, Authentication Key, Send, Receive, Metric, and Status.

RIP Version 2 Configuration	
Address:	192.49.110.253
Domain:	0x00:00 <input type="button" value="Submit"/>
Authentication Type:	noAuthentication(1) <input type="button" value="Submit"/>
Authentication Key:	0x00:00:00:00:00:00:00:00 <input type="button" value="Submit"/>
Send:	doNotSend(1) <input type="button" value="Submit"/>
Receive:	rip1OrRip2(3) <input type="button" value="Submit"/>
Metric:	1 <input type="button" value="Submit"/>
Status:	valid(1) <input type="button" value="Submit"/>

Figure 32. RIP Version 2—Statistics Configuration window

Address (*rip2IfConfAddress*)

The IP Address of this system on the indicated subnet. For unnumbered interfaces, the value 0.0.0.N, where the least significant 24 bits (N) is the ifIndex for the IP Interface in network byte order.

Domain (*rip2IfConfDomain*)

Value inserted into the Routing Domain field of all RIP packets sent on this interface.

Authentication Type (*rip2IfConfAuthType*)

The type of Authentication used on this interface.

- noAuthentication (1)
- simplePassword (2)

Authentication Key (*rip2IfConfAuthKey*)

This value is used as the Authentication Key whenever Authentication Type (*rip2IfConfAuthType*) has a value other than *noAuthentication(1)*. A modification of Authentication Type does not change the value of Authentication Key. If the Authentication Key string is shorter than 16 octets, it will be left justified, then padded to 16 octets with nulls (0x00) on the right.

Reading this object always results in an octet string of length zero. Authentication may not be bypassed by reading the MIB object.

Send (*rip2IfConfSend*)

Send is what the router sends on this interface. *ripVersion 1* implies sending RIP updates compliant with RFC 1058. There are four options, *doNotSend(1)*, *ripVersion1(2)*, *rip1Compatible(3)*, and *ripVersion2(4)*. *rip1Compatible* implies broadcasting RIP-2 updates using RFC 1058 route subsumption rules. *ripVersion2* implies multicasting RIP-2 updates. *ripV1Demand* indicates the use of Demand RIP on a WAN interface under RIP Version 1 rules. *ripV2Demand* indicates the use of Demand RIP on a WAN interface under Version 2 rules.

- doNotSend (1)
- ripVersion1 (2)
- rip1Compatible (3)—*rip1Compatible* implies broadcasting RIP-2 updates using RFC 1058 route subsumption rules
- ripVersion2 (4)—*ripVersion2* implies multicasting RIP-2 updates

Receive (*rip2IfConfReceive*)

This indicates which version of RIP updates are to be accepted. Note that *rip2* and *rip1OrRip2* implies reception of multicast packets.

- rip1 (1)
- rip2 (2)
- rip1OrRip2 (3)
- doNotRecieve (4)

Metric (*rip2IfConfDefaultMetric*)

This variable indicates the metric that is to be used for the default route entry in RIP updates originated on this interface. A value of zero indicates that no default route should be originated; in this case, a default route via another router may be propagated.

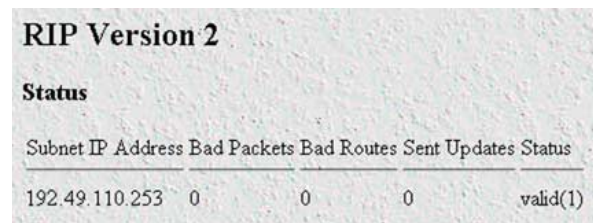
Status (*rip2IfConfStatus*)

Writing invalid has the effect of deleting this interface.

- valid (1)
- invalid (2)

RIP Version 2 (Statistics)

By clicking on Statistics in the RIP Version 2 main window, you enter the RIP Version 2 Status window (see figure 33)It displays routing and update information for each subnet address.



RIP Version 2				
Status				
Subnet IP Address	Bad Packets	Bad Routes	Sent Updates	Status
192.49.110.253	0	0	0	valid(1)

Figure 33. RIP Version 2 details window

Subnet IP Address (*rip2IfStatAddress*)

The IP Address of this system on the indicated subnet. For unnumbered interfaces, the value 0.0.0.N, where the least significant 24 bits (N) is the ifIndex for the IP Interface in network byte order.

Bad Packets (*rip2IfStatRcvBadPackets*)

The number of RIP response packets received by the RIP process which were subsequently discarded for any reason (e.g. a version 0 packet, or an unknown command type).

Bad Routes (*rip2IfStatRcvBadRoutes*)

The number of routes, in valid RIP packets, which were ignored for any reason (e.g. unknown address family, or invalid metric).

Sent Updates (*rip2IfStatSentUpdates*)

The number of triggered RIP updates actually sent on this interface. This explicitly does NOT include full updates sent containing new information.

Status (*rip2IfStatStatus*)

Displays whether the Subnet IP Address is valid or invalid.

Chapter 14 **SNMP**

Chapter contents

Introduction	90
SNMP window.....	90
In	90
Packets (snmpInPkts)	90
Bad Version (snmpInBadVersions)	90
Bad Community Names (snmpInBadCommunityNames)	91
Bad Community Uses (snmpInBadCommunity)	91
ASN ParseErrors (snmpInASNParseErrs)	91
Error Status “Too Big” (snmpInTooBigs)	91
No Such Names (snmpInNoSuchNames)	91
Bad Values (snmpInBadValues)	91
Error Status “Read Only” (snmpInReadOnlys)	91
Generated Errors (snmpInGenErrs)	91
Get/Get Next Variables (snmpInTotalReqVars)	91
Set Variables (snmpInTotalSetVars)	91
Get Requests (snmpInGetRequests)	91
Get Next Requests (snmpInGetNexts)	92
Set Requests (snmpInSetRequests)	92
Get Responses (snmpInGetResponses)	92
Traps (snmpInTraps)	92
Out	92
Out Packets (snmpOutPkts)	92
Error Status “Too Big” (snmpOutTooBigs)	92
No Such Names (snmpOutNoSuchNames)	92
Bad Values (snmpOutBadValues)	92
Generated Errors (snmpOutGenErrs)	92
Get Requests (snmpOutGetRequests)	92
Get Next Requests (snmpOutGetNexts)	92
Set Requests (snmpOutSetRequests)	92
Get Responses (snmpOutGetResponses)	93
Traps (snmpOutTraps)	93
Authentication Failure Traps (snmpEnableAuthenTraps)	93

Introduction

The DACS provides management and statistical information on SNMP. Detailed information on the SNMP MIB variables may be downloaded from the RFC. Click on SNMP under the Configuration Menu to display the SNMP window (see figure 34).

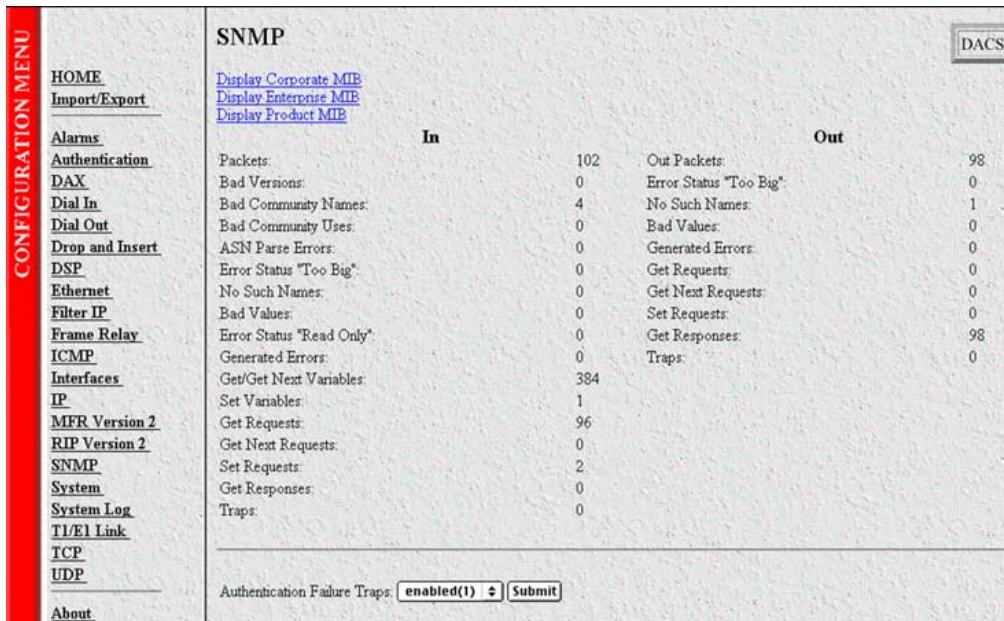


Figure 34. SNMP window

SNMP window

The SNMP window displays incoming and outgoing SNMP statistics, and has links for downloading and displaying the following MIB documents:

- Corporate MIB
- Enterprise MIB
- Product MIB

In

Packets (snmplnPkts)

The total number of Messages delivered to the SNMP entity from the transport service. Typically this would be UDP since the SNMP engine sits on top of UDP

Bad Version (snmplnBadVersions)

The total number of SNMP Messages that were delivered to the SNMP protocol entity and were for an unsupported SNMP version.

Bad Community Names (*snmplnBadCommunityNames*)

The total number of SNMP Messages delivered to the SNMP protocol entity which used a SNMP community name not known to said entity.

Bad Community Uses (*snmplnBadCommunity*)

The total number of SNMP messages delivered to the SNMP protocol entity which represented an SNMP operation which was not allowed by the SNMP community named in the message.

ASN ParseErrors (*snmplnASNParseErrs*)

The total number of ASN.1 or BER errors encountered by the SNMP protocol entity when decoding received SNMP messages.

Error Status "Too Big" (*snmplnTooBig*)

The total number of SNMP PDUs that were delivered to the SNMP protocol entity and for which the value of the error-status field is *tooBig*.

No Such Names (*snmplnNoSuchNames*)

The total number of SNMP PDUs that were delivered to the SNMP protocol entity and for which the value of the error-status field is *noSuchName*.

Bad Values (*snmplnBadValues*)

The total number of SNMP PDUs that were delivered to the SNMP protocol entity and for which the value of the error-status field is *badValue*.

Error Status "Read Only" (*snmplnReadOnly*)

The total number of valid SNMP PDUs that were delivered to the SNMP protocol entity and for which the value of the error-status field is *readOnly*. It should be noted that it is a protocol error to generate an SNMP PDU which contains the *readOnly* value in the error-status field, as such this object is provided as a means of detecting incorrect implementations of the SNMP.

Generated Errors (*snmplnGenErrs*)

The total number of SNMP PDUs that were delivered to the SNMP protocol entity and for which the value of the error-status field is *genErr*.

Get/Get Next Variables (*snmplnTotalReqVars*)

The total number of MIB objects that have been retrieved successfully by the SNMP protocol entity as the result of receiving valid SNMP Get-Request and Get-Next PDUs.

Set Variables (*snmplnTotalSetVars*)

The total number of MIB objects that have been altered successfully by the SNMP protocol entity as the result of receiving valid SNMP Set-Request PDUs.

Get Requests (*snmplnGetRequests*)

The total number of SNMP Get-Request PDUs that have been accepted and processed by the SNMP protocol entity.

Get Next Requests (*snmpInGetNexts*)

The total number of SNMP Get-Next PDUs that have been accepted and processed by the SNMP protocol entity.

Set Requests (*snmpInSetRequests*)

The total number of SNMP Set-Request PDUs that have been accepted and processed by the SNMP protocol entity.

Get Responses (*snmpInGetResponses*)

The total number of SNMP Get-Response PDUs that have been accepted and processed by the SNMP protocol entity.

Traps (*snmpInTraps*)

The total number of SNMP Trap PDUs that have been accepted and processed by the SNMP protocol entity.

Out

Out Packets (*snmpOutPkts*)

The total number of SNMP messages that were passed from the SNMP protocol entity to the transport service.

Error Status "Too Big" (*snmpOutTooBig*s)

The total number of SNMP PDUs that were generated by the SNMP protocol entity and for which the value of the error-status field is *tooBig*.

No Such Names (*snmpOutNoSuchNames*)

The total number of SNMP PDUs that were generated by the SNMP protocol entity and for which the value of the error-status is *noSuchName*.

Bad Values (*snmpOutBadValues*)

The total number of SNMP PDUs that were generated by the SNMP protocol entity and for which the value of the error-status field is *badValue*.

Generated Errors (*snmpOutGenErrs*)

The total number of SNMP PDUs that were generated by the SNMP protocol entity and for which the value of the error-status field is *genErr*.

Get Requests (*snmpOutGetRequests*)

The total number of SNMP Get-Request PDUs that have been generated by the SNMP protocol entity.

Get Next Requests (*snmpOutGetNexts*)

The total number of SNMP Get-Next PDUs that have been generated by the SNMP protocol entity.

Set Requests (*snmpOutSetRequests*)

The total number of SNMP Set-Request PDUs that have been generated by the SNMP protocol entity.

Get Responses (*snmpOutGetResponses*)

The total number of SNMP Get-Response PDUs that have been generated by the SNMP protocol entity.

Traps (*snmpOutTraps*)

The total number of SNMP Trap PDUs that have been generated by the SNMP protocol entity.

Authentication Failure Traps (*snmpEnableAuthenTraps*)

This value indicates whether the SNMP agent process is permitted to generate authentication-failure traps. The variable is global. This means that by being disabled, all authentication-failure traps are disabled.

Note Note: It is strongly recommended that upon selecting either *enabled(1)* or *disabled(2)*, it be saved in non-volatile memory by clicking on **Record Current Configuration** under Immediate Actions on the Home page of the 2604 DACS. If the network management system is re-initialized (implying power cycling) while assuming the current configuration has been recorded, the current configuration will not be lost.

The two options for this variable are:

- enabled(1)
- disabled(2)

Chapter 15 System

Chapter contents

Introduction	97
System main window.....	98
CPU	98
Percentage CPU Idle (boxidletime)	98
Time Slices Fully Utilized (boxCPUcritical)	98
Time Slices 90% Utilized (boxCPUWarning)	98
SNMP and HTTP	98
Version (boxSnmpVersion)	98
Super User Password (boxSnmpMasterPassword)	98
User Password (boxSnmpMonitorPassword)	98
LAN IP	98
How to Obtain Address (boxIPAddressTechnique)	99
Address(boxIPAddress)	99
Mask(boxIPMask)	99
Manufacturer	99
Serial Number (boxManufactureDatecode)	99
PCB Revision (boxManufacturePcbRevision)	99
General Information (boxManufactureGeneralInfo)	99
Message Blocks	99
Packet Holding Message Blocks...	99
Total (boxMsgBlksConfigured)	99
Free (boxMsgBlksFree)	99
Total Time Waited (boxCountMsgBlkTaskWait)	99
Total Times Unavailable (boxCountMsgBlkUnavailable)	100
Operating System Heap Memory	100
Total Size (boxHeapSize)	100
Free (boxHeapFreeSpace)	100
Largest (boxHeapLargestSpace)	100
Enclosure System	100
Internal Temperature (boxTemperature)	100
Highest Temperature (boxMaxTemperature)	100
Installation	100
Country (installCountry)	100
Other	100
Total DRAM Detected (boxDetectedMemory)	100
SystemID (sysObjectID)	100
Running Since Last Boot (sysUpTime)	101
System Manager (sysContact)	101
Box Name (sysName)	101

Physical Location (sysLocation)	101
Web Settings (boxBackgroundFlag)	101
Monitor Privilege (boxMonitorPrivilege)	101
System—Modify window	102
SNMP and HTTP	102
Version (boxSnmpVersion)	102
Super User Password (boxSnmpMasterPassword)	102
User Password (boxSnmpMonitorPassword)	103
LAN IP	103
Method to Obtain Address (boxIPAddressTechnique)	103
Address (boxIPAddress)	103
Mask (boxIPMask)	103
Installation	103
Country (installCountry)	103
Other	104
System Manager (sysContact)	104
Box Name (sysName)	104
Physical Location (sysLocation)	104
Web Settings (boxBackgroundFlag)	104
Monitor Privilege (boxMonitorPrivilege)	104
System—Packet Holding Message Blocks.....	105
Buffer Size (boxbuffersize)	105
No. of Buffers (boxbuffercount)	105
No. Free (boxbuffersfree)	105
No. of Tasks Waited (boxCountBufferTaskWait)	105
No. of Times Unavailable(boxCountBufferUnavailable)	105

Introduction

The System main window (see figure 35) contains general setup information about the DACS. System parameters are Patton Enterprise MIB object identifiers, though some are contained in RFC 1213, *Management Information Base for Network Management of TCP/IP-based internets: MIB-II*. Click on System under the Configuration Menu to display the System main window.

System Main Window - 1st Screen Half

SYSTEM

[Modify...](#)

CPU

% CPU Idle: 99
Time Slices Fully Utilized: 1514
Time Slices 90% Utilized: 67

SNMP and HTTP

Version: snmpv1(1)
Super User Password: No Access
User Password: monitor

LAN IP

How to Obtain Address: static(1)
Address: 192.49.110.136
Mask: 255.255.255.0

Manufacturer

Serial Number: 03/13/01
PCB Revision: 1
General Information:

Message Blocks

[Packet Holding Message Blocks...](#)

Total: 19590
Free: 19182
Total Time Waited: 0
Total Times Unavailable: 0

System Main Window - 2nd Screen Half

Operating System Heap Memory

Total Size: 13380608
Free: 8405504
Largest: 8398848

Enclosure System

Internal Temperature: 52 celsius
Highest Temperature: 56 celsius

Installation

Country: unitedStates(1)

Other

Total DRAM Detected: 14431584
System ID: 1.3.6.1.4.1.1768.1
Running Since Last Boot: 2 days 22:27:59 hours
System Manager: Unknown Contact
Box Name: DACS
Physical Location: Unknown Location
Background Image: enableGraphics(1)
Monitor Privilege: readonly(2)

Figure 35. System main window
(CPU, SNMP and HTTP, LAN IP, Manufacturer, and Message Blocks)

System main window

From this window you can view information for the CPU, SNMP and HTTP, LAN IP, Manufacturer, Message Blocks, Operating System Heap Memory, Enclosure System, Installation, and Other.

The main window also has the following links:

- **Modify**—click on this link to change SNMP and HTTP, LAN IP, Installation, and Other (see “System—Modify window” on page 102)
- **Packet Holding Message Blocks**—click on this link to view message block statistics (see “System—Packet Holding Message Blocks...” on page 105)

CPU

This portion of the System main window contains information described in the following sections.

Percentage CPU Idle (boxidletime)

This indicates what percentage of the i960 CPU processing power is not being utilized (see figure 35 on page 97).

Time Slices Fully Utilized (boxCPUcritical)

This value represents a count of how many times the CPU was fully utilized expressed in 1/100th seconds (see figure 35 on page 97).

Time Slices 90% Utilized (boxCPUWarning)

This value represents a count of how many times the CPU approached full utilization expressed in 1/100th seconds (see figure 35 on page 97).

SNMP and HTTP

This portion of the System main window provides information about the SNMP version and the HTTP accessibility.

Version (boxSnmVersion)

This parameter indicates the SNMP version number supported by this unit (for example *snmpv1(1)* means SNMP version 1 is supported). Select *snmpv1(1)* only since SNMP2 is not currently supported.

Super User Password (boxSnmMasterPassword)

This is the super user password for complete access and configurability of the DACS through SNMP and HTTP (see figure 35 on page 97).

User Password (boxSnmMonitorPassword)

This displays the user monitoring password for read only access of certain selected information. Not all parameters shown using the superuser password are displayed under the user password. (see figure 35 on page 97).

LAN IP

This portion of the System main window contains information described in the following sections.

How to Obtain Address (boxIPAddressTechnique)

This displays the current method for obtaining the LAN IP address (see figure 35 on page 97).

Address(boxIPAddress)

If the address technique in use above is static, then the value displayed in the Address field is the LAN IP address (see figure 35 on page 97).

Mask(boxIPMask)

If the address technique in use above is static, then the value displayed in the Address field is the LAN IP mask (see figure 35 on page 97).

Manufacturer

This portion of the System main window contains manufacturing information described in the following sections.

Serial Number (boxManufactureDatecode)

The datecode of manufacture and serial number (see figure 35 on page 97).

PCB Revision (boxManufacturePcbRevision)

The revision of the printed circuit board (see figure 35 on page 97).

General Information (boxManufactureGeneralInfo)

A manufacturing notes area for additional information (see figure 35 on page 97).

Message Blocks

This portion of the System main window contains information about the usage of message blocks. A message block is essentially memory available for creating or storing packets where a packet is usually an Ethernet frame. There are four types of message blocks, and each type represents a collection of buffers which are of the same size.

Packet Holding Message Blocks...

Provides buffer usage of DACS message blocks based upon message block sizes (see figure 35 on page 97).

Total (boxMsgBlksConfigured)

The total number of message blocks on the system (see figure 35 on page 97).

Free (boxMsgBlksFree)

The number of free message blocks available (see figure 35 on page 97).

Total Time Waited (boxCountMsgBlkTaskWait)

The total number of times that the proper size message block was not available to hold a packet, and the CPU task went to sleep while waiting for it. (see figure 35 on page 97).

Total Times Unavailable (*boxCountMsgBlkUnavailable*)

The total number of times that the proper size message block was not available to hold a packet, and the CPU task dumped the packet. The difference between Total Time Waited and Total Times Unavailable is whether the CPU task goes to sleep or simply dumps the packet to continue on. (see figure 35 on page 97).

Operating System Heap Memory

This portion of the System main window contains information about the memory used by the CPU and its management tasks.

Total Size (*boxHeapSize*)

The size in octets of the operating system heap memory (see figure 35).

Free (*boxHeapFreeSpace*)

The amount of operating system heap memory in octets currently available (see figure 35).

Largest (*boxHeapLargestSpace*)

The largest contiguous memory block in octets in the memory heap (see figure 35).

Enclosure System

This portion of the System main window contains information about the internal temperature of the DACS.

Internal Temperature (*boxTemperature*)

Displays the current temperature in celsius (centigrade) (see figure 35).

Highest Temperature (*boxMaxTemperature*)

The highest temperature registered in celsius (centigrade) since the DACS was last re-booted (see figure 35 on page 97).

Installation

This portion of the System main window contains information described in this following section.

Country (*installCountry*)

Specifies the country that the DACS is installed in so it can be configured in accordance with local laws (see figure 35 on page 97).

Other

This portion of the System main window contains information described in the following sections.

Total DRAM Detected (*boxDetectedMemory*)

The total number of bytes of DRAM detected by the CPU (see figure 35 on page 97).

SystemID (*sysObjectID*)

This SNMP variable defines the type of the DACS being managed as defined by specification RFC1213.MIB (see figure 35 on page 97).

Running Since Last Boot (sysUpTime)

This SNMP variable represents the time since the network management portion of the system was last re-initialized (see figure 35 on page 97).

System Manager (sysContact)

This SNMP variable represents the textual identification of the contact person for this managed node, which may include information on how to contact this person as defined by specification RFC1213.MIB (see figure 35 on page 97). The maximum length of this field is 256 octets.

Box Name (sysName)

This is “An administratively assigned name for this managed node. By convention, this is the node’s fully-qualified domain name.” (RFC1213.MIB) (see figure 35 on page 97).

Physical Location (sysLocation)

“The physical location of this node (e.g., *telephone closet, 3rd floor*.)” (RFC1213.MIB) (see figure 35 on page 97).

Web Settings (boxBackgroundFlag)

The following options are available:

- `disableGraphics(0)`—When this option is selected, graphics on WWW pages will not be displayed. This results in faster page display times, but may make it more difficult to navigate WWW sites that rely heavily on graphics.
- `enableGraphics(1)`—When this option is selected, graphics on WWW pages are displayed.
- `disableWeb(2)`—When this option is selected, access to the WWW pages is denied for everyone.

Monitor Privilege (boxMonitorPrivilege)

Specifies the privileges given to the monitor user. Privileges can be removed or additional write access can be given beyond read-only access. The following options are available:

- `none(0)`—The monitor user can not log in.
- `read-only(2)`—This is the default setting. The monitor user can view but not change any parameters. Monitor can not view passwords.
- `writeUser(18)`—Not supported.
- `writeUserIp(50)`—The monitor user can change all parameters—except passwords— IP links.
- `writeUserIpWan(114)`—The monitor user can change all parameters—except passwords— IP, and T1/E1.
- `writeUserIpWanSystem(242)`—The monitor user can change all parameters—except passwords— IP, T1/E1, System, and System Log links.
- `writeUserIpWanSystemUpload(498)`—The monitor user can change all parameters—except passwords— IP, T1/E1, System, and System Log links. The monitor user can also load firmware updates into the DACS.

System—Modify window

The System—Modify window (see figure 36) is where you can change SNMP and HTTP, LAN IP, Installation, and Other.

The screenshot shows a web-based configuration interface with the following sections:

- SNMP AND HTTP**:
 - Version:
 - Superuser Password:
 - Superuser Password Verification:
 - User Password:
 - User Password Verification:
- LAN IP**:
 - Method to Obtain Address:
 - Address:
 - Mask:
 -
- Installation**:
 - Country:
- Other**:
 - System Manager:
 - Box Name:
 - Physical Location:
 - Web Settings:
 - Monitor Privilege:

Figure 36. System—Modify window

SNMP and HTTP

This portion of the System—Modify window provides information about the SNMP version and the HTTP accessibility.

Version (*boxSnmVersion*)

This parameter selects the SNMP version number supported by this unit (see figure 36). Select *snmpv1(1)* only, SNMP2 is not currently supported.

Super User Password (*boxSnmMasterPassword*)

This accesses the super user password for complete access and configurability of the DACS through SNMP and HTTP (see figure 36 on page 102).

User Password (boxSnmpMonitorPassword)

This accesses the user monitoring password for read only access of certain selected information. Not all parameters shown using the superuser password are displayed under the user password. (see figure 36 on page 102).

LAN IP

This portion of the System—Modify window contains configurable information for the IP addressing of the Ethernet LAN port.

Method to Obtain Address (boxIPAddressTechnique)

This indicates how to obtain the LAN IP address (see figure 36 on page 102). The following options are available:

- `disable(0)`—Ethernet port is disabled (DACS T1 to T1 usage only)
- `static(1)`—LAN IP address is obtained from EIA-232 Control Port
- `rarp(2)`—Reverse Address Resolution Protocol—A protocol defined in RFC 903 which provides the reverse function of ARP. RARP maps a hardware address (MAC address) to an Internet address. It is used primarily by diskless nodes, when they first initialize, to find their Internet address.
- `bootp(3)`—The Bootstrap Protocol. A protocol described in RFCs 951 and 1084 and used for booting diskless workstations.
- `dhcp(4)`—Dynamic Host Configuration Protocol—A protocol introduced by Microsoft on their NT server with version 3.5 in late 1994. This protocol provides a means to dynamically allocate IP addresses to IBM PCs running on a Microsoft Windows local area network. The system administrator assigns a range of IP addresses to DHCP and each client PC on the LAN has its TCP/IP software configured to request an IP address from the DHCP server. The request and grant process uses a lease concept with a controllable time period. More information can be found in the Microsoft documentation on NT Server.

Address (boxIPAddress)

If the address technique above is static then this represents the LAN IP address.

Mask (boxIPMask)

If the address technique above is static then this represents the LAN IP mask.

Installation

This portion of the System—Modify window contains information described in the following section.

Country (installCountry)

Specifies the country that the DACS is installed in so it can be configured in accordance with local laws. The following options are available:

- `other(0)`
- `unitedStates(1)`
- `australia(2)`
- `canada(3)`
- `europaUnion(4)`

- france(5)
- germany(6)

Other

This portion of the System—Modify window contains information described in the following sections.

System Manager (sysContact)

This SNMP variable represents the textual identification of the contact person for this managed node, together with information on how to contact this person as defined by specification RFC1213.MIB.

Box Name (sysName)

This is “An administratively assigned name for this managed node. By convention, this is the node’s fully-qualified domain name.” (RFC1213.MIB)

Physical Location (sysLocation)

“The physical location of this node (e.g., ‘telephone closet, 3rd floor’).” (RFC1213.MIB)

Web Settings (boxBackgroundFlag)

The following options are available:

- disableGraphics(0)—When this option is selected, graphics on WWW pages will not be displayed. This results in faster page display times, but may make it more difficult to navigate WWW sites that rely heavily on graphics.
- enableGraphics(1)—When this option is selected, graphics on WWW pages are displayed.
- disableWeb(2)—When this option is selected, access to the WWW pages is denied for everyone.

Monitor Privilege (boxMonitorPrivilege)

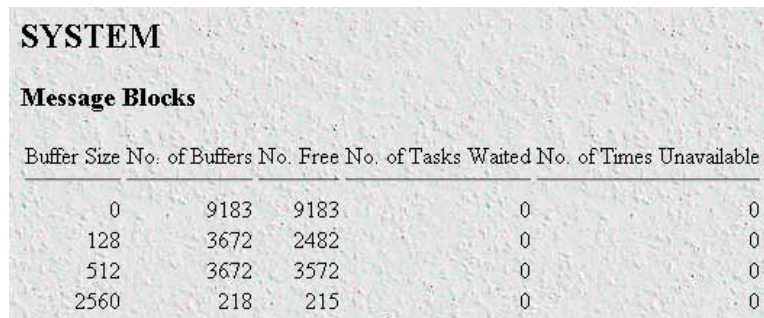
Specifies the privileges given to the monitor user. Privileges can be removed or additional write access can be given beyond read-only access. The following options are available:

- none(0)—The monitor user can not log in.
- read-only(2)—This is the default setting. The monitor user can view but not change any parameters. Monitor can not view passwords.
- writeUser(18)—The monitor user can change all parameters—except passwords— under authentication, drop-and-insert, and dial-in links.
- writeUserIp(50)—The monitor user can change all parameters—except passwords— under authentication, drop-and-insert, dial-in, and IP links.
- writeUserIpWan(114)—The monitor user can change all parameters—except passwords— under authentication, drop-and-insert, dial-in, IP, T1/E1, and Frame Relay links.
- writeUserIpWanSystem(242)—The monitor user can change all parameters—except passwords— under authentication, drop-and-insert, dial-in, IP, T1/E1, Frame Relay, System, and System Log links.

- `writeUserlpWanSystemUpload(498)`—The monitor user can change all parameters—except passwords—under authentication, drop-and-insert, dial-in, IP, T1/E1, Frame Relay, System, and System Log links. The monitor user can also load firmware updates into the DACS.

System—Packet Holding Message Blocks...

The DACS system manages the i960 processor utilization by allocating message blocks for packet management. This Message Blocks window (see figure 37) indicates buffer usage of DACS message blocks based upon message block sizes.



Buffer Size	No. of Buffers	No. Free	No. of Tasks Waited	No. of Times Unavailable
0	9183	9183	0	0
128	3672	2482	0	0
512	3672	3572	0	0
2560	218	215	0	0

Figure 37. Packet Holding Message Blocks window

Buffer Size (boxbuffersize)

The size in bytes of the buffer.

No. of Buffers (boxbuffercount)

The total number of buffers this size.

No. Free (boxbuffersfree)

The number of buffers this size which are currently free for use

No. of Tasks Waited (boxCountBufferTaskWait)

The total number of times that the proper size message block was not available to hold a packet, and the CPU task went to sleep while waiting for it.

No. of Times Unavailable(boxCountBufferUnavailable)

The total number of times that the proper size message block was not available to hold a packet, and the CPU task dumped the packet. The difference between Total Time Waited and Total Times Unavailable is whether the CPU task goes to sleep or simply dumps the packet to continue on.

Chapter 16 System Log

Chapter contents

Introduction	108
System Log Main Window	108
System Log—Modify	109
Daemons	109
SysLog Daemon IP Address(syslogDaemonIP)	109
SNMP Trap Daemon IP Address (syslogTrapIP)	109
Priority	109
Min Priority for SysLog Daemon (syslogDaemonPriority)	110
Min Priority for Console RS-232 (syslogConsolePriority)	110
Min Priority for Flash Storage (syslogFlashPriority)	110
Min Priority for SNMP Trap Daemon (syslogTrapPriority)	110
Min Priority for RAM (SyslogTablePriority)	111
Unix Facility (syslogUnixFacility)	111
Call Trace (syslogCallTrace)	112
Maintenance	112
Maintain Flash Storage (syslogFlashClear)	112
System Log—Volatile Memory.....	113
Time (slTick)	113
Message (slMessage)	113
System Log—Non-Volatile Memory	114
Time (slfTick)	114
Message (slfMessage)	114

Introduction

The System Log window (see figure 38) displays the results from the system-wide error reporting subsystem. The object parameters in the system log are all Patton Enterprise MIB object identifiers.

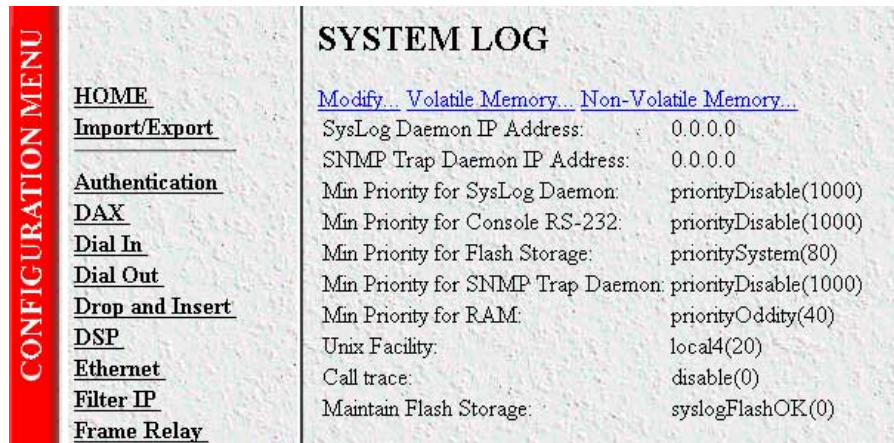


Figure 38. System Log main window

System Log Main Window

Besides displaying the results from the system-wide error reporting utility, the System Log main window also contains links to the following:

- **Modify**—Clicking on this link displays syslog and SNMP trap daemon IP addresses, message priorities, and maintenance information in the flash memory (see “System Log—Modify” on page 109)
- **Volatile Memory**—Clicking on this link displays timestamp and stored system log message information (“System Log—Volatile Memory” on page 113)
- **Non-Volatile Memory**—Clicking on this link displays non-volatile RAM messages with the 100-ms time stamp (see “System Log—Non-Volatile Memory” on page 114)

Click on System Log under the Configuration Menu to display the System Log main window.

System Log—Modify

The System Log—Modify window (see figure 39) displays SysLog and SNMP Trap Daemon IP Address locations, message priorities for the offered SysLog destinations, priority and maintenance information.

SYSTEM LOG

Daemons

SysLog Daemon IP Address:

SNMP Trap Daemon IP Address:

Submit

Priority

Min Priority for SysLog Daemon:

Min Priority for Console RS-232:

Min Priority for Flash Storage:

Min Priority for SNMP Trap Daemon:

Min Priority for RAM:

Unix Facility:

Call trace:

Submit

Maintenance

Maintain Flash Storage:

Submit

Figure 39. System Log—Modify window

Daemons

This portion of the System Log—Modify window contains information about the SysLog Daemon and SNMP Trap Daemon IP Addresses.

SysLog Daemon IP Address(syslogDaemonIP)

The IP address of a host system which is running a syslog daemon. System messages with a priority greater than or equal to the configurable syslogDaemonPriority will be sent to this IP address (see section “Priority”).

SNMP Trap Daemon IP Address (syslogTrapIP)

The IP address of a host system which is running a SNMP trap daemon. SNMP Trap messages with a priority greater than or equal to the configurable syslogTrapPriority will be sent to this IP address.

Priority

This portion of the System Log—Modify window describes the configuration of the Message Priority for each of the SysLog destinations.

Min Priority for SysLog Daemon (syslogDaemonPriority)

System messages which have a priority equal to or greater than this setting will be sent to the syslog daemon defined by the SysLog Daemon IP Address (syslogDaemonIP).

- prioritySystem(80)
- priorityDisable(1000)

Min Priority for Console RS-232 (syslogConsolePriority)

System messages which have a priority equal to or greater than this setting will be sent directly to the RS-232 Config Port on the rear of the 2604. Messages will be sent regardless of the current operating state of the RS-232 configuration port. The lower the number next to the priority listed below, the more details system logging will provide. *priorityVerbose* will generate the most messages, while *priorityDisable* will turn off all messages.

- priorityVerbose(5)
- priorityDebug(10)
- priorityInfo(20)
- priorityOddity(40)
- priorityService(60)
- prioritySystem(80)
- priorityDisable(1000)

Min Priority for Flash Storage (syslogFlashPriority)

System messages which have a priority equal to or greater than this setting will be permanently stored in the Flash PROM. Due to being permanent memory, the Flash memory eventually becomes filled. When this occurs, the memory must be cleared before accepting more messages. Some maximum number of messages may be stored in the Flash PROM before this storage area must be cleared.

- prioritySystem(80)—Flash PROM will be used to store system-level messages.
- priorityDisable(1000)—No messages will be stored.

Min Priority for SNMP Trap Daemon (syslogTrapPriority)

System messages which have a priority equal to or greater than this setting will be sent to the SNMP Trap Daemon IP Address (syslogTrapIP). The lower the number next to the priority listed below, the more details system logging will provide. *priorityVerbose* will generate the most messages, while *priorityDisable* will turn off all messages.

- priorityVerbose(5)
- priorityDebug(10)
- priorityInfo(20)
- priorityOddity(40)
- priorityService(60)

- `prioritySystem(80)`
- `priorityDisable(1000)`

Min Priority for RAM (SyslogTablePriority)

System messages which have a priority equal to or greater than this setting will appear in System Log—Volatile Memory. The lower the number next to the priority listed below, the more details system logging will provide. *priorityVerbose* will generate the most messages, while *priorityDisable* will turn off all messages.

- `priorityVerbose(5)`
- `priorityDebug(10)`
- `priorityInfo(20)`
- `priorityOddity(40)`
- `priorityService(60)`
- `prioritySystem(80)`
- `priorityDisable(1000)`

Unix Facility (syslogUnixFacility)

This setting is used when syslog messages are sent to a Unix-type syslog daemon. In this case the message will include the facility and priority coding.

- `disable(0)`
- `user(1)`
- `mail(2)`
- `daemon(3)`
- `auth(4)`
- `syslog(5)`
- `lpr(6)`
- `news(7)`
- `uucp(8)`
- `cron(9)`
- `authpriv(10)`
- `ftp(11)`
- `local0(16)`
- `local1(17)`
- `local2(18)`
- `local3(19)`
- `local4(20)`

- local5(21)
- local6(22)
- local7(23)

Call Trace (syslogCallTrace)

Enabling this will activate the call tracing utility. This is a powerful debugging utility which will log every single function call and return. At the death of a box the call trace will be printed out and can be sent to tech support. This utility will take a large amount of CPU power.

- disable(0)—Disable function call tracing.
- enable(1)—Enable function call tracing.
- dump(2)—Display function call tracing on the computer monitor.

Maintenance

This portion of the System Log—Modify window contains information described in the following section.

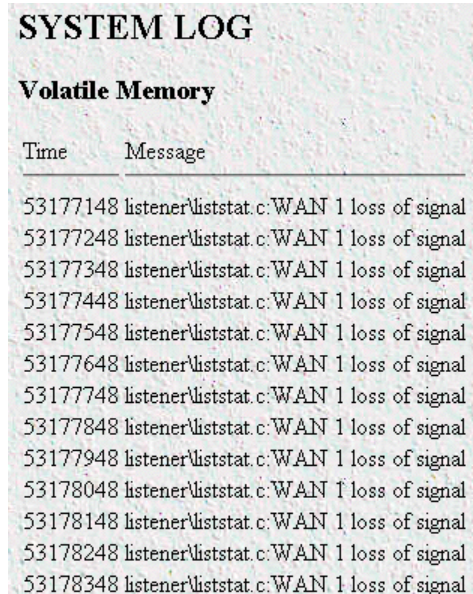
Maintain Flash Storage (syslogFlashClear)

Setting this variable to syslogFlashClear will cause the erasing of any system messages which have been saved in the Flash. On reading this variable will indicate if the syslog Flash is rejecting messages because it is full.

- syslogFlashOK(0)—Flash is accepting messages.
- syslogFlashFull(1)—Flash is rejecting messages because it is full. To empty the flash memory, see option *syslogFlashClear(2)*.
- syslogFlashClear(2)—Erase system messages stored in Flash. Be sure to return to the 2604's Home page and click on **Record Current Configuration** to store this change in permanent memory.

System Log—Volatile Memory

The System Log—Volatile Memory window (see figure 40) displays timestamp and stored system log message information.



Time	Message
53177148	listener\liststat.c:WAN 1 loss of signal
53177248	listener\liststat.c:WAN 1 loss of signal
53177348	listener\liststat.c:WAN 1 loss of signal
53177448	listener\liststat.c:WAN 1 loss of signal
53177548	listener\liststat.c:WAN 1 loss of signal
53177648	listener\liststat.c:WAN 1 loss of signal
53177748	listener\liststat.c:WAN 1 loss of signal
53177848	listener\liststat.c:WAN 1 loss of signal
53177948	listener\liststat.c:WAN 1 loss of signal
53178048	listener\liststat.c:WAN 1 loss of signal
53178148	listener\liststat.c:WAN 1 loss of signal
53178248	listener\liststat.c:WAN 1 loss of signal
53178348	listener\liststat.c:WAN 1 loss of signal

Figure 40. System Log—Volatile Memory window

Time (slTick)

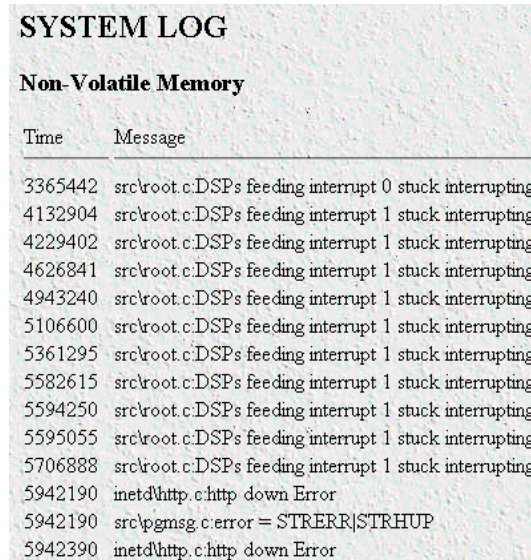
Time stamps are generated every 10 ms.

Message (slMessage)

This is the message stored in RAM. If the 2604 DACS loses power, the messages in RAM will be lost.

System Log—Non-Volatile Memory

The System Log—Non-Volatile window (see figure 41) displays the time stamp and the messages kept in the non-volatile Flash memory.



Time	Message
3365442	src/root.c:DSPs feeding interrupt 0 stuck interrupting
4132904	src/root.c:DSPs feeding interrupt 1 stuck interrupting
4229402	src/root.c:DSPs feeding interrupt 1 stuck interrupting
4626841	src/root.c:DSPs feeding interrupt 1 stuck interrupting
4943240	src/root.c:DSPs feeding interrupt 1 stuck interrupting
5106600	src/root.c:DSPs feeding interrupt 1 stuck interrupting
5361295	src/root.c:DSPs feeding interrupt 1 stuck interrupting
5582615	src/root.c:DSPs feeding interrupt 1 stuck interrupting
5594250	src/root.c:DSPs feeding interrupt 1 stuck interrupting
5595055	src/root.c:DSPs feeding interrupt 1 stuck interrupting
5706888	src/root.c:DSPs feeding interrupt 1 stuck interrupting
5942190	inetd/http.c:http down Error
5942190	src/pgmsg.c:error = STRERR STRHUP
5942390	inetd/http.c:http down Error

Figure 41. System Log—Non-Volatile Memory window

Time (*slfTick*)

Time stamps are generated every 10 ms.

Message (*slfMessage*)

This is the message stored in Flash memory. If the 2604 DACS loses power, the messages will *not* be lost.

Chapter 17 T1/E1 Link

Chapter contents

Introduction	118
T1/E1 Link Activity main window	119
Link (dsx1LineIndex)	119
Type (dsx1LineType)	119
Circuit ID (dsx1CircuitIdentifier)	119
Line Status (dsx1LineStatus).....	120
Failure States	120
Far End Alarm Failure	120
Alarm Indication Signal (AIS) Failure	121
Loss Of Frame Failure	121
Loss Of Signal Failure	121
Loopback Pseudo-Failure	121
TS16 Alarm Indication Signal Failure	121
Loss Of MultiFrame Failure	121
Far End Loss Of Multiframe Failure	121
Line Status—Configuration.....	122
Time Elapsed (dsx1TimeElapsed)	122
Valid Intervals (dsx1ValidIntervals)	122
WAN Circuit Configuration—Modify.....	123
Line Interface Settings	123
Circuit ID (dsx1CircuitIdentifier)	123
Line Type (dsx1LineType) Type (dsx1LineType)	123
Line Coding (dsx1LineCoding)	124
Receive Equalizer (linkRxEqualizer)	124
Line Build Out (linkLineBuildOut)	124
Yellow Alarm Format (linkYellowFormat)	124
FDL (dsx1FDL)	125
Test Settings	125
Force Yellow Alarm (linkYellowForce)	125
Loopback Config (dsx1LoopbackConfig)	125
Send Code (dsx1SendCode)	125
Error Injection (linkInjectError)	126
Yellow Alarm Severity ()	126
Red Alarm Severity ()	126
Near End Line Statistics—Current.....	127
Errored Seconds (dsx1CurrentESs)	127
Severely Errored Seconds (dsx1CurrentSESs)	127
Severely Errored Frame Seconds (dsx1CurrentSEFSs)	127
Unavailable Seconds (dsx1CurrentUASs)	127

Controlled Slip Seconds (dsx1CurrentCSSs)	127
Path Code Violations (dsx1CurrentPCVs)	127
Line Errored Seconds (dsx1CurrentLESs)	127
Bursty ErroredSeconds (dsx1CurrentBESs)	127
Degraded Minutes (dsx1CurrentDMs)	128
Line Code Violations (dsx1CurrentLCVs)	128
Near End Line Statistics—History	128
Interval (dsx1IntervalNumber)	128
Errored Seconds (dsx1intervaless)	128
Severely Errored Seconds (dsx1IntervalSESs)	128
Severely Errored Frame Seconds (dsx1IntervalSEFSs)	129
Unavailable Seconds (dsx1IntervalUASs)	129
Controlled Slip Seconds (dsx1IntervalCSSs)	129
Path Code Violations (dsx1IntervalPCVs)	129
Line Errored Seconds (dsx1IntervalLESs)	129
Bursty ErroredSeconds (dsx1IntervalBESs)	129
Degraded Minutes (dsx1IntervalDMs)	129
Line Code Violations (dsx1IntervalLCVs)	129
Near End Line Statistics—Totals	130
Errored Seconds (dsx1TotalESs)	130
Severely Errored Seconds (dsx1TotalSESs)	130
Severely Errored Frame Seconds (dsx1TotalSEFSs)	130
Unavailable Seconds (dsx1TotalUASs)	130
Controlled Slip Seconds (dsx1TotalCSSs)	130
Path Code Violations (dsx1TotalPCVs)	130
Line Errored Seconds (dsx1TotalLESs)	130
Bursty ErroredSeconds (dsx1TotalBESs)	130
Degraded Minutes (dsx1TotalDMs)	131
Line Code Violations (dsx1TotalLCVs)	131
Far End Line Statistics—Current	131
Time Elapsed (dsx1FarEndTimeElapsed)	131
Errored Seconds (dsx1FarEndCurrentESs)	131
Severely Errored Seconds (dsx1FarEnd CurrentSESs)	131
Severely Errored Frame Seconds (dsx1FarEndCurrentSEFSs)	131
Unavailable Seconds (dsx1FarEndCurrentUASs)	131
Controlled Slip Seconds (dsx1FarEndCurrentCSSs)	132
Line Errored Seconds (dsx1FarEndCurrentLESs)	132
Path Code Violations (dsx1FarEndCurrentPCVs)	132
Bursty Errored Seconds (dsx1FarEndCurrentBESs)	132
Degraded Minutes (dsx1FarEndCurrentDMs)	132
Far End Line Statistics—History	132
Interval (dsx1FarEndIntervalNumber)	133
Errored Seconds (dsx1FarEndIntervalESs)	133
Severely Errored Seconds (dsx1FarEndIntervalSESs)	133

Severely Errored Frame Seconds (dsx1FarEndIntervalSEFSs)	133
Unavailable Seconds (dsx1FarEndIntervalUASs)	133
Controlled Slip Seconds (dsx1FarEndIntervalCSSs)	133
Line Errored Seconds (dsx1FarEndIntervalLESs)	133
Path Code Violations (dsx1FarEndIntervalPCVs)	133
Bursty Errored Seconds (dsx1FarEndIntervalBESs)	133
Degraded Minutes (dsx1FarEndIntervalDMs)	133
Far End Line Statistics—Totals	134
Errored Seconds (dsx1FarEndTotalESs)	134
Severely Errored Seconds (dsx1FarEndTotalSESSs)	134
Severely Errored Frame Seconds (dsx1FarEndTotalSEFSs)	134
Unavailable Seconds (dsx1FarEndTotalUASs)	134
Controlled Slip Seconds (dsx1FarEndTotalCSSs)	134
Line Errored Seconds (dsx1FarEndTotalLESs)	134
Path Code Violations (dsx1FarEndTotalPCVs)	134
Bursty Errored Seconds (dsx1FarEndTotalBESs)	135
Degraded Minutes (dsx1FarEndTotalDMs)	135

Introduction

The T1/E1 Link Activity window (see figure 42) shows the configuration of the T1/E1 Interface, and reports statistics on the quality of the T1/E1 connection. The statistics listed in this section comprise those contained in *RFC 1406—Definitions of Managed Objects for the DS1 and E1 Interface Types*.



Figure 42. T1/E1 Link Activity main window

Click on T1/E1 Link under the Configuration Menu to display the T1/E1 Link Activity main window.

The T1/E1 Link Activity main window contains the following items:

- Information that identifies the DS1 Interface on a managed device, indicates the type of DS1 line using the circuit, and shows the transmission vendor's circuit identifier (see figure 42). For more information about the objects in this window, refer to "T1/E1 Link Activity main window" on page 119.
- Line Status—This variable indicates interface line status. If any condition other than No Alarms exists, you can click on the Alarms Present link to view the Line Status Alarms window. For more information about these objects, refer to "Line Status (dsx1LineStatus)" on page 120.
- Line Status—Configuration... link—clicking on this link takes you to the page that displays the WAN Circuit Configuration window. This window contains general information about the DS1 interface, amount of time intervals passed, and kind of line coding). For more information about this page, refer to "Line Status—Configuration" on page 122.
- Near End Line Statistics—Current... link—clicking on this link takes you to the page that displays line statistics for the current 15-minute interval. For more information about this page, refer to "Near End Line Statistics—Current" on page 127.

- Near End Line Statistics—History... link—clicking on this link takes you to the page that displays line statistics for the previous 15-minute interval. For more information about this page, refer to “Near End Line Statistics—History” on page 128.
- Near End Line Statistics—Totals... link—clicking on this link takes you to the page that displays the total statistics of errors that occurred during the previous 24-hour period. For more information about this page, refer to “Near End Line Statistics—Totals” on page 130.
- Far End Line Statistics—Current... link—clicking on this link takes you to the page that displays far-end statistics for the current 15-minute interval. For more information about this page, refer to “Far End Line Statistics—Current” on page 131.
- Far End Line Statistics—History... link—clicking on this link takes you to the page that displays far-end statistics for the previous 15-minute interval. For more information about this page, refer to “Far End Line Statistics—History” on page 132.
- Far End Line Statistics—Totals... link—clicking on this link takes you to the page that displays the total far-end statistics of errors that occurred during the previous 24-hour period. For more information about this page, refer to “Far End Line Statistics—Totals” on page 134.

T1/E1 Link Activity main window

The T1/E1 Link Activity window has three main sections that display the following T1/E1 parameters:

- Line Status—Shows the configuration of the T1/E1 Interface and service provided on each user time slot.
- Near End Line Statistics—Show error statistics collected from the near-end of the T1/E1 line.
- Far End Line Statistics—Show statistics collected from the far-end T1/E1 line. Far End Line Statistics can be used by devices that support the facility data link (FDL)

Link (*dsx1LineIndex*)

This object identifies a DS1 Interface on a managed device. If there is an ifEntry directly associated with this DS1 interface, it must have the same value as ifIndex. Otherwise, the value exceeds ifNumber, and is assigned a unique identifier by following this rule: inside interfaces (equipment side) with even numbers and outside interfaces (network side) with odd numbers.

Type (*dsx1LineType*)

This variable indicates the type of DS1 line using the circuit. The circuit type determines the bits-per-second rate that the circuit can carry and how it interprets error statistics. The values are as follows:

- dsx1ESF—Extended Superframe DS1
- dsx1D4—AT&T D4 format DS1
- dsx1E1—Based on CCITT/ITU G.704 without CRC
- dsx1E1-CRC—Based on CCITT/ITU G.704 with CRC

Circuit ID (*dsx1CircuitIdentifier*)

This is the transmission vendor's circuit identifier. Knowing the circuit ID can be helpful during troubleshooting.

Line Status (dsx1LineStatus)

This variable indicates interface line status. It contains loopback, failure, received alarm and transmitted alarm information. If any condition other than No Alarms exists, you can click on the Alarms Present link to view the Line Status Alarms window (see figure 43).

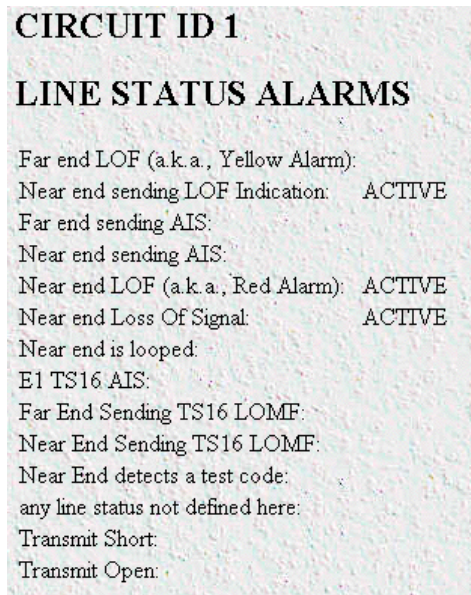


Figure 43. Line Status Alarms window

The alarms currently present on the line will be indicated by the ACTIVE label next to the alarm type.

Failure States

The following failure states are reported in the dsx1LineStatus object. The items listed in this section comprise those contained in *RFC 1406—Definitions of Managed Objects for the DS1 and E1 Interface Types*.

Far End Alarm Failure

Far End Alarm failure is also known as a *Yellow Alarm* in the T1 case or *Distant Alarm* in the E1 case.

For D4 links, the Far End Alarm failure occurs when bit 6 of all channels has been zero for at least 335 ms. The alarm is cleared when bit 6 of at least one channel is non-zero for a period T , where T is usually less than 1 second and always less than 5 seconds. The Far End Alarm failure is not declared for D4 links when a Loss of Signal is detected.

For ESF links, the Far End Alarm failure is declared if the Yellow Alarm signal pattern occurs in at least 7 out of 10 contiguous 16-bit pattern intervals. The alarm is cleared when the Yellow Alarm signal pattern has not occurred for 10 contiguous 16-bit signal pattern intervals.

For E1 links, the Far End Alarm failure is declared when bit 3 of time-slot zero is received set to 1 on two consecutive occasions. The Far End Alarm failure is cleared when bit 3 of time-slot zero is received set to zero.

Alarm Indication Signal (AIS) Failure

The Alarm Indication Signal failure is declared when an AIS defect is detected at the input and the AIS defect still exists after the Loss Of Frame failure (which is caused by the unframed nature of the *all-ones* signal) is declared. The AIS failure is cleared when the Loss Of Frame failure is cleared.

Loss Of Frame Failure

For T1 links, the Loss Of Frame failure is declared when an OOF or LOS defect has persisted for T seconds, where $2 \leq T \leq 10$. The Loss Of Frame failure is cleared when there have been no OOF or LOS defects during a period T where $0 \leq T \leq 20$. Many systems will perform *hit integration* within the period T before declaring or clearing the failure (for more information, see TR 62411 [16]).

For E1 links, the Loss Of Frame Failure is declared when an OOF defect is detected.

Loss Of Signal Failure

For T1, the Loss Of Signal failure is declared upon observing 175 +/- 75 contiguous pulse positions with no pulses of either positive or negative polarity. The LOS failure is cleared upon observing an average pulse density of at least 12.5% over a period of 175 ±75 contiguous pulse positions, starting with the receipt of a pulse.

For E1 links, the Loss Of Signal failure is declared when greater than 10 consecutive zeroes are detected (see O.162 Section 3.4.4).

Loopback Pseudo-Failure

The Loopback Pseudo-Failure is declared when the near end equipment has placed a loopback (of any kind) on the DS1. This allows a management entity to determine from one object whether the DS1 can be considered to be in service or not (from the point of view of the near end equipment).

TS16 Alarm Indication Signal Failure

For E1 links, the TS16 Alarm Indication Signal failure is declared when time-slot 16 is received as all ones for all frames of two consecutive multiframes (see G.732 Section 4.2.6). This condition is never declared for T1.

Loss Of MultiFrame Failure

The Loss Of MultiFrame failure is declared when two consecutive multiframe alignment signals (bits 4 through 7 of TS16 of frame 0) have been received with an error. The Loss Of Multiframe failure is cleared when the first correct multiframe alignment signal is received. The Loss Of Multiframe failure can only be declared for E1 links operating with G.732 [18] framing (sometimes called *Channel Associated Signalling* mode).

Far End Loss Of Multiframe Failure

The Far End Loss Of Multiframe failure is declared when bit 2 of TS16 of frame 0 is received set to one on two consecutive occasions. The Far End Loss Of Multiframe failure is cleared when bit 2 of TS16 of frame 0 is received set to zero. The Far End Loss Of Multiframe failure can only be declared for E1 links operating in *Channel Associated Signalling* mode.

Line Status—Configuration

Clicking on the Line Status—Configuration link in the T1/E1 Link Activity window displays the WAN Circuit Configuration window (see figure 44). This window contains general information about the DS1 interface, including the type of line (D4 Superframe or Extended Superframe), and kind of line coding (B8ZS or AMI). To modify the WAN circuit configuration, click on the Modify... link. For more information about modifying WAN circuit settings, refer to “WAN Circuit Configuration—Modify” on page 123.

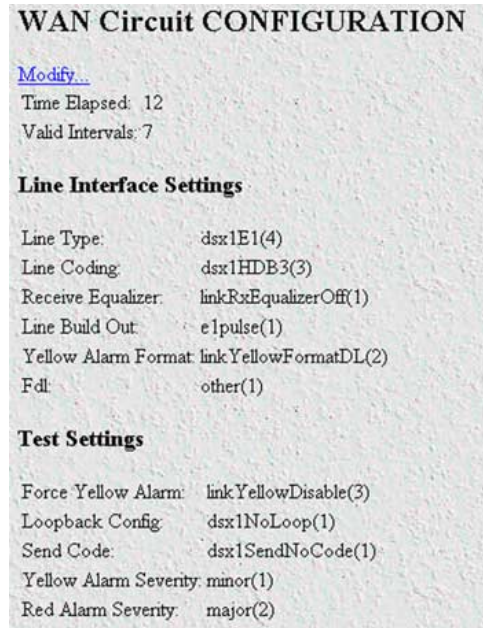


Figure 44. WAN Circuit Configuration window

Note Click on the Modify link to change the settings of any of the following parameters (see “WAN Circuit Configuration—Modify” on page 123).

The WAN Circuit Configuration window also displays the amount of time that has passed and the number of intervals passed during which valid data was collected.

Time Elapsed (*dsx1TimeElapsed*)

The number of seconds that have elapsed since the beginning of the current error-measurement period.

Valid Intervals (*dsx1ValidIntervals*)

The number of previous intervals for which valid data was collected. The value will be 96 unless the interface was brought on-line within the last 24-hours, in which case the value will be the number of completed 15-minute intervals since the interface has been online. Statistics are collected for up to the last 24 hour period broken down into 96 individual 15-minute intervals.

WAN Circuit Configuration—Modify

Clicking on the Configuration link in the T1/E1 Link Activity window displays the WAN Circuit Configuration—Modify window (see figure 45). From this window, you can change line interface settings, signalling settings, test settings, and change the T1/E1 pulse shapes.

Figure 45. WAN Circuit Configuration—Modify window

Line Interface Settings

This portion of the WAN Circuit Configuration window contains information described in the following sections.

Circuit ID (*dsx1CircuitIdentifier*)

This variable contains the transmission vendor's circuit identifier, for the purpose of facilitating troubleshooting.

Line Type (*dsx1LineType*) Type (*dsx1LineType*)

This variable indicates the type of DS1 Line implemented on this circuit. The type of circuit affects the number of bits per second that the circuit can reasonably carry, as well as the interpretation of the usage and error statistics. The values, in sequence, are:

- other(1) —Link is disabled
- dsx1ESF(2)—Extended Superframe DS1
- dsx1D4(3)—AT&T D4 format DS1

- dsx1E1(4)—Based on CCITT/ITU G.704 without CRC
- dsx1E1-CRC(5)—Based on CCITT/ITU G.704 with CRC

Line Coding (dsx1LineCoding)

This variable describes the type of Zero Code Suppression used on the link, which in turn affects a number of its characteristics.

- dsx1JBZS(1)—Jammed Bit Zero Suppression, in which the AT&T specification of at least one pulse every 8 bit periods is literally implemented by forcing a pulse in bit 8 of each channel. Thus, only seven bits per channel, or 1.344 Mbps, is available for data. This feature is not currently implemented.
- dsx1B8ZS (2)—The use of a specified pattern of normal bits and bipolar violations which are used to replace a sequence of eight zero bits.
- dsx1HDB3(3)
- dsx1ZBTSI(4)—May use *dsx1ZBTSI*, or Zero Byte Time Slot Interchange. This feature is not currently implemented.
- dsx1AMI(5)—Refers to a mode wherein no zero code suppression is present and the line encoding does not solve the problem directly. In this application, the higher layer must provide data which meets or exceeds the pulse density requirements, such as inverting HDLC data.
- other(6)—This feature is not currently supported.

Receive Equalizer (linkRxEqualizer)

This variable determines the equalization used on the received signal. Long haul signals should have the equalization set for more. Short haul signals require less equalization.

- linkRxEqualizerOff(1)
- linkRxEqualizerOn(2)

Line Build Out (linkLineBuildOut)

This variable is used in T1 applications to adjust the T1 pulse shape at the cross connect point. Select the pulse strength needed to minimize distortion at the remote T1 receiver end. The default is *t1pulse0dB*, which should be adequate for most situations.

- triState(0)
- e1pulse(1)
- t1pulse0dB(2)—Strong pulse amplitude.
- t1pulse-7dB(3)—Medium pulse amplitude.
- t1pulse-15dB(4)—Weak pulse amplitude.

Yellow Alarm Format (linkYellowFormat)

This variable identifies which standard will be used to transmit and identify the Yellow Alarm.

- linkYellowFormatBit2(1)—Bit-2 equal zero in every channel
- YellowFormatDL(2)—FF00 pattern in the Data Link

- YellowFormatFrame12FS(3)—FS bit of frame 12

FDL (dsx1FDL)

This describes which implementation of FDL is being used, if any. FDL applies only to T1 circuits.

- other(1)—Indicates that a protocol other than one following is used.
- dsx1Ansi-T1-403(2)—Refers to the FDL exchange recommended by ANSI.
- dsx1Att-54016(3)—Refers to ESF FDL exchanges.
- dsx1Fdl-none(4)—Indicates that the device does not use the FDL.

Test Settings

This portion of the WAN Circuit Configuration window contains information described in the following sections.

Force Yellow Alarm (linkYellowForce)

This variable identifies which standard will be used to transmit and identify the Yellow Alarm.

- linkYellowAuto—Do *not* force the transmission of a yellow alarm. But, yellow alarm may be automatically transmitted.
- linkYellowOn—Force the transmission of a yellow alarm even if the received signal is in frame.
- linkYellowDisable—Do NOT transmit a yellow alarm even if the received signal is out of frame.

Loopback Config (dsx1LoopbackConfig)

This variable represents the loopback configuration of the DS1 interface. Agents supporting read/write access should return badValue in response to a requested loopback state that the interface does not support. The values mean:

- dsx1NoLoop—Not in the loopback state. A device that is not capable of performing a loopback on the interface shall always return this as its value.
- dsx1PayloadLoop—The received signal at this interface is looped through the device. Typically the received signal is looped back for retransmission after it has passed through the device's framing function.
- dsx1LineLoop—The received signal at this interface does not go through the device (minimum penetration) but is looped back out.
- dsx1OtherLoop—Loopbacks that are not defined here.

Send Code (dsx1SendCode)

This variable indicates what type of code is being sent across the DS1 interface by the device. The values mean:

- dsx1SendNoCode—Sending looped or normal data
- dsx1SendLineCode—Sending a request for a line loopback
- dsx1SendPayloadCode—Sending a request for a payload loopback
- dsx1SendResetCode—Sending a loopback termination request
- dsx1SendQRS—Sending a Quasi-Random Signal (QRS) test pattern

- `dsx1Send511Pattern`—Sending a 511 bit fixed test pattern
- `dsx1Send3in24Pattern`—Sending a fixed test pattern of 3 bits set in 24
- `dsx1SendOtherTestPattern`—Sending a test pattern other than those described by this object.

Error Injection (`linkInjectError`)

Force an output error to see if the other end detects it

- `noErrorInjection(0)`
- `injectCRCErrorBurst(1)`
- `injectLineErrorBurst(2)`

Yellow Alarm Severity ()

This reference is identical to the reference on the Alarms page in the 2604 Configuration Menu. The configuration may be changed here or in the Alarms page.

- `ignore(0)`
- `minor(1)`
- `major(2)`
- `minorSelfClearing(3)`
- `majorSelfClearing(4)`

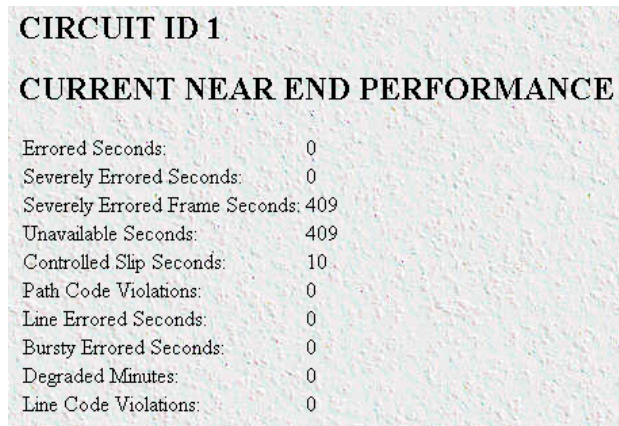
Red Alarm Severity ()

This reference is identical to the reference on the Alarms page in the 2604 Configuration Menu. The configuration may be changed here or in the Alarms page.

- `ignore(0)`
- `minor(1)`
- `major(2)`
- `minorSelfClearing(3)`
- `majorSelfClearing(4)`

Near End Line Statistics—Current

Click on Near End Line Statistics—Current to display line statistics for the current 15-minute interval (see figure 46).



CIRCUIT ID 1	
CURRENT NEAR END PERFORMANCE	
Errored Seconds:	0
Severely Errored Seconds:	0
Severely Errored Frame Seconds:	409
Unavailable Seconds:	409
Controlled Slip Seconds:	10
Path Code Violations:	0
Line Errored Seconds:	0
Bursty Errored Seconds:	0
Degraded Minutes:	0
Line Code Violations:	0

Figure 46. Current Near End Performance window

Errored Seconds (*dsx1CurrentESs*)

The number of errored seconds, encountered by a DS1 interface in the current 15-minute interval.

Severely Errored Seconds (*dsx1CurrentSESs*)

The number of severely errored seconds encountered by a DS1 interface in the current 15-minute interval.

Severely Errored Frame Seconds (*dsx1CurrentSEFSs*)

The number of severely errored framing seconds encountered by a DS1 interface in the current 15-minute interval.

Unavailable Seconds (*dsx1CurrentUASs*)

The number of unavailable seconds encountered by a DS1 interface in the current 15-minute interval.

Controlled Slip Seconds (*dsx1CurrentCSSs*)

The number of Controlled Slip Seconds encountered by a DS1 interface in the current 15-minute interval.

Path Code Violations (*dsx1CurrentPCVs*)

The number of path coding violations encountered by a DS1 interface in the current 15-minute interval.

Line Errored Seconds (*dsx1CurrentLESs*)

The number of line errored seconds encountered by a DS1 interface in the current 15-minute interval.

Bursty Errored Seconds (*dsx1CurrentBESs*)

The number of bursty errored seconds (BESs) encountered by a DS1 interface in the current 15-minute interval.

Degraded Minutes (dsx1CurrentDMs)

The number of degraded minutes (DMs) encountered by a DS1 interface in the current 15-minute interval.

Line Code Violations (dsx1CurrentLCVs)

The number of line code violations (LCVs) encountered by a DS1 interface in the current 15-minute interval.

Near End Line Statistics—History

Click on Near End Line Statistics—History to display line statistics for prior completed 15-minute intervals within the last 24 hours (see figure 47). This does not include the current 15-minute interval.

Interval	Errored Seconds	Severely Errored Seconds	Errored Frame Seconds	Unavailable Seconds	Controlled Slip Seconds	Path Code Violations	Line Errored Seconds	Bursty Errored Seconds	Degraded Minutes	Line Code Violations
1	0	0	900	900	22	0	0	0	0	0
2	0	0	900	900	22	0	0	0	0	0
3	0	0	900	900	22	0	0	0	0	0
4	0	0	900	900	23	0	0	0	0	0
5	0	0	900	900	22	0	0	0	0	0
6	0	0	900	900	22	0	0	0	0	0
7	0	0	900	900	22	0	0	0	0	0
8	0	0	900	900	22	0	0	0	0	0
9	0	0	900	900	22	0	0	0	0	0
10	0	0	900	900	22	0	0	0	0	0
11	0	0	900	900	22	0	0	0	0	0
12	0	0	900	900	22	0	0	0	0	0
13	0	0	900	900	22	0	0	0	0	0

Figure 47. History of Near End Performance window

Interval (dsx1IntervalNumber)

A number between 1 and 96, where 1 is the most recently completed 15-minute interval and 96 is the least recently completed 15-minute interval. When all 96 intervals are visible, then the 2604 has been operating (powered-on) for at least 24 hours. If less than 96 intervals are visible, then it has been less than 24 hours since the 2604 was powered up.

Errored Seconds (dsx1Intervaless)

The number of errored Seconds encountered by a DS1 interface in one of the previous 96, individual 15-minute, intervals.

Severely Errored Seconds (dsx1IntervalSESs)

The number of severely errored seconds encountered by a DS1 interface in one of the previous 96, individual 15-minute, intervals.

Severely Errored Frame Seconds (*dsx1IntervalSEFSs*)

The number of severely errored framing seconds encountered by a DS1 interface in one of the previous 96, individual 15-minute, intervals.

Unavailable Seconds (*dsx1IntervalUASs*)

The number of unavailable seconds encountered by a DS1 interface in one of the previous 96, individual 15-minute, intervals.

Controlled Slip Seconds (*dsx1IntervalCSSs*)

The number of controlled slip seconds encountered by a DS1 interface in one of the previous 96, individual 15-minute, intervals.

Path Code Violations (*dsx1IntervalPCVs*)

The number of path coding violations encountered by a DS1 interface in one of the previous 96, individual 15-minute, intervals.

Line Errored Seconds (*dsx1IntervalLESs*)

The number of line errored seconds encountered by a DS1 interface in one of the previous 96, individual 15-minute, intervals.

Bursty Errored Seconds (*dsx1IntervalBESs*)

The number of bursty errored seconds (BESs) encountered by a DS1 interface in one of the previous 96, individual 15-minute, intervals.

Degraded Minutes (*dsx1IntervalDMs*)

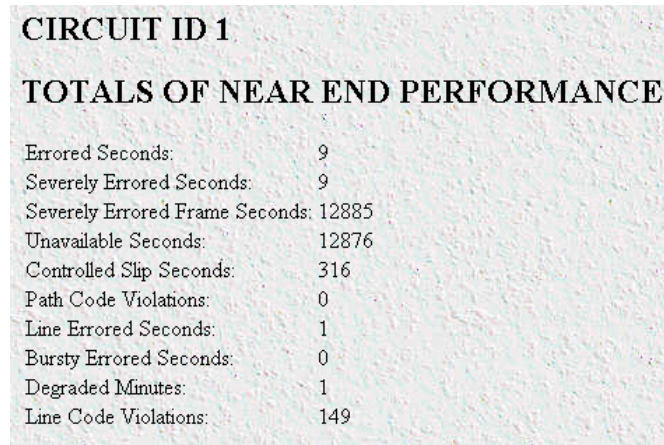
The number of degraded minutes (DMs) encountered by a DS1 interface in one of the previous 96, individual 15-minute, intervals.

Line Code Violations (*dsx1IntervalLCVs*)

The number of line code violations (LCVs) encountered by a DS1 interface in the current 15-minute interval.

Near End Line Statistics—Totals

Click on Near End Line Statistics—Totals to display the total statistics of errors that occurred during the previous 24-hour period, the previous 96 15-minute intervals (see figure 48).



CIRCUIT ID 1	
TOTALS OF NEAR END PERFORMANCE	
Errored Seconds:	9
Severely Errored Seconds:	9
Severely Errored Frame Seconds:	12885
Unavailable Seconds:	12876
Controlled Slip Seconds:	316
Path Code Violations:	0
Line Errored Seconds:	1
Bursty Errored Seconds:	0
Degraded Minutes:	1
Line Code Violations:	149

Figure 48. Totals of Near End Performance window

Errored Seconds (*dsx1TotalESs*)

The number of errored seconds encountered by a DS1 interface in the previous 24-hour interval.

Severely Errored Seconds (*dsx1TotalSEsS*)

The number of severely errored seconds encountered by a DS1 interface in the previous 24-hour interval.

Severely Errored Frame Seconds (*dsx1TotalSEFSs*)

The number of severely errored framing seconds encountered by a DS1 interface in the previous 24-hour interval.

Unavailable Seconds (*dsx1TotalUASs*)

The number of unavailable seconds encountered by a DS1 interface in the previous 24-hour interval.

Controlled Slip Seconds (*dsx1TotalCSSs*)

The number of controlled slip seconds encountered by a DS1 interface in the previous 24-hour interval.

Path Code Violations (*dsx1TotalPCVs*)

The number of path coding violations encountered by a DS1 interface in the previous 24-hour interval.

Line Errored Seconds (*dsx1TotalLESs*)

The number of line errored seconds encountered by a DS1 interface in the previous 24-hour interval.

Bursty Errored Seconds (*dsx1TotalBESs*)

The number of bursty errored seconds (BESs) encountered by a DS1 interface in the previous 24-hour interval.

Degraded Minutes (*dsx1TotalDMs*)

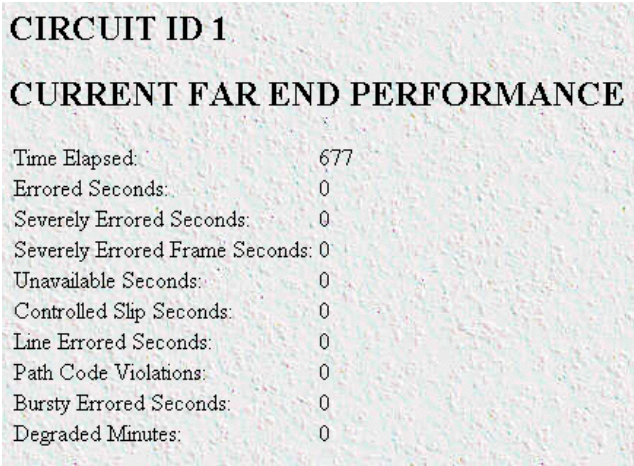
The number of degraded minutes (DMs) encountered by a DS1 interface in the previous 24-hour interval.

Line Code Violations (*dsx1TotalLCVs*)

The number of line code violations (LCVs) encountered by a DS1 interface in the current 15-minute interval.

Far End Line Statistics—Current

Click on Near End Line Statistics—Current to display far-end statistics for the current 15-minute interval (see figure 49).



CIRCUIT ID 1	
CURRENT FAR END PERFORMANCE	
Time Elapsed:	677
Errored Seconds:	0
Severely Errored Seconds:	0
Severely Errored Frame Seconds:	0
Unavailable Seconds:	0
Controlled Slip Seconds:	0
Line Errored Seconds:	0
Path Code Violations:	0
Bursty Errored Seconds:	0
Degraded Minutes:	0

Figure 49. Current Far End Performance window

Time Elapsed (*dsx1FarEndTimeElapsed*)

The number of seconds that have elapsed since the beginning of the far-end current error-measurement period.

Errored Seconds (*dsx1FarEndCurrentESs*)

The number of far-end errored seconds encountered by a DS1 interface in the current 15-minute interval.

Severely Errored Seconds (*dsx1FarEndCurrentSESs*)

The number of far-end severely errored seconds encountered by a DS1 interface in the current 15-minute interval.

Severely Errored Frame Seconds (*dsx1FarEndCurrentSEFSs*)

The number of far-end severely errored framing seconds encountered by a DS1 interface in the current 15-minute interval.

Unavailable Seconds (*dsx1FarEndCurrentUASs*)

The number of far-end unavailable seconds encountered by a DS1 interface in the current 15-minute interval.

Controlled Slip Seconds (*dsx1FarEndCurrentCSSs*)

The number of far-end controlled slip seconds encountered by a DS1 interface in the current 15-minute interval.

Line Errored Seconds (*dsx1FarEndCurrentLESs*)

The number of far-end line errored seconds encountered by a DS1 interface in the current 15-minute interval

Path Code Violations (*dsx1FarEndCurrentPCVs*)

The number of far-end path coding violations reported via the far-end block error count encountered by a DS1 interface in the current 15-minute interval.

Bursty Errored Seconds (*dsx1FarEndCurrentBESs*)

The number of far-end bursty errored seconds (BESs) encountered by a DS1 interface in the current 15-minute interval.

Degraded Minutes (*dsx1FarEndCurrentDMs*)

The number of far-end degraded minutes (DMs) encountered by a DS1 interface in the current 15-minute interval.

Far End Line Statistics—History

Click on Far End Line Statistics—History to display far-end statistics for previously completed 15-minute intervals (see figure 50).

Interval	Errored Seconds	Severely Errored Seconds	Severely Errored Frame Seconds	Unavailable Seconds	Controlled Slip Seconds	Line Errored Seconds	Path Code Violations	Bursty Errored Seconds	Degraded Minutes
1	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0

Figure 50. History of Far End Performance window

Interval (dsx1FarEndIntervalNumber)

A number between 1 and 96, where 1 is the most recently completed 15-minute interval and 96 is the least recently completed 15-minute interval (assuming that all 96 intervals are valid).

Errored Seconds (dsx1FarEndIntervalESs)

The number of far-end errored seconds encountered by a DS1 interface in one of the previous 96, individual 15-minute, intervals.

Severely Errored Seconds (dsx1FarEndIntervalSESs)

The number of far-end severely errored seconds encountered by a DS1 interface in one of the previous 96, individual 15-minute, intervals.

Severely Errored Frame Seconds (dsx1FarEndIntervalSEFSs)

The number of far-end severely errored framing seconds encountered by a DS1 interface in one of the previous 96, individual 15-minute, intervals.

Unavailable Seconds (dsx1FarEndIntervalUASs)

The number of far-end unavailable seconds encountered by a DS1 interface in one of the previous 96, individual 15-minute, intervals.

Controlled Slip Seconds (dsx1FarEndIntervalCSSs)

The number of far-end controlled slip seconds encountered by a DS1 interface in one of the previous 96, individual 15-minute, intervals.

Line Errored Seconds (dsx1FarEndIntervalLESs)

The number of far-end line errored seconds encountered by a DS1 interface in one of the previous 96, individual 15-minute, intervals.

Path Code Violations (dsx1FarEndIntervalPCVs)

The number of far-end path coding violations encountered by a DS1 interface in one of the previous 96, individual 15-minute, intervals.

Bursty Errored Seconds (dsx1FarEndIntervalBESs)

The number of far-end bursty errored seconds (BESs) encountered by a DS1 interface in one of the previous 96, individual 15-minute, intervals.

Degraded Minutes (dsx1FarEndIntervalDMs)

The number of far-end degraded minutes (DMs) encountered by a DS1 interface in one of the previous 96, individual 15-minute, intervals.

Far End Line Statistics—Totals

Click on Far End Line Statistics—Totals to display the total statistics of errors that occurred during the previous 24-hour period (see figure 51). This is the sum of the current 15-minute interval and all time prior intervals within the last 24 hours.

CIRCUIT ID 1	
FAR END PERFORMANCE	
Errored Seconds:	0
Severely Errored Seconds:	0
Severely Errored Frame Seconds:	0
Unavailable Seconds:	0
Controlled Slip Seconds:	0
Line Errored Seconds:	0
Path Code Violations:	0
Bursty Errored Seconds:	0
Degraded Minutes:	0

Figure 51. Far End Performance window

Errored Seconds (*dsx1FarEndTotalESs*)

The number of far-end errored seconds encountered by a DS1 interface in the previous 24-hour interval.

Severely Errored Seconds (*dsx1FarEndTotalSESs*)

The number of far-end severely errored seconds encountered by a DS1 interface in the previous 24-hour interval.

Severely Errored Frame Seconds (*dsx1FarEndTotalSEFSs*)

The number of far-end severely errored framing seconds encountered by a DS1 interface in the previous 24-hour interval.

Unavailable Seconds (*dsx1FarEndTotalUASs*)

The number of far-end unavailable seconds encountered by a DS1 interface in the previous 24-hour in-24-hour interval.

Controlled Slip Seconds (*dsx1FarEndTotalCSSs*)

The number of far-end controlled slip seconds encountered by a DS1 interface in the previous 24-hour interval.

Line Errored Seconds (*dsx1FarEndTotalLESs*)

The number of far-end line errored seconds encountered by a DS1 interface in the previous 24-hour interval.

Path Code Violations (*dsx1FarEndTotalPCVs*)

The number of far-end path coding violations reported via the far-end block error count encountered by a DS1 interface in the previous 24-hour interval.

Bursty Errored Seconds (dsx1FarEndTotalBESs)

The number of far-end bursty errored seconds (BESs) encountered by a DS1 interface in the previous 24-hour interval.

Degraded Minutes (dsx1FarEndTotalDMs)

The number of far-end degraded minutes (DMs) encountered by a DS1 interface in the previous 24-hour interval.

Chapter 18 **T1/E1 Assignment**

Chapter contents

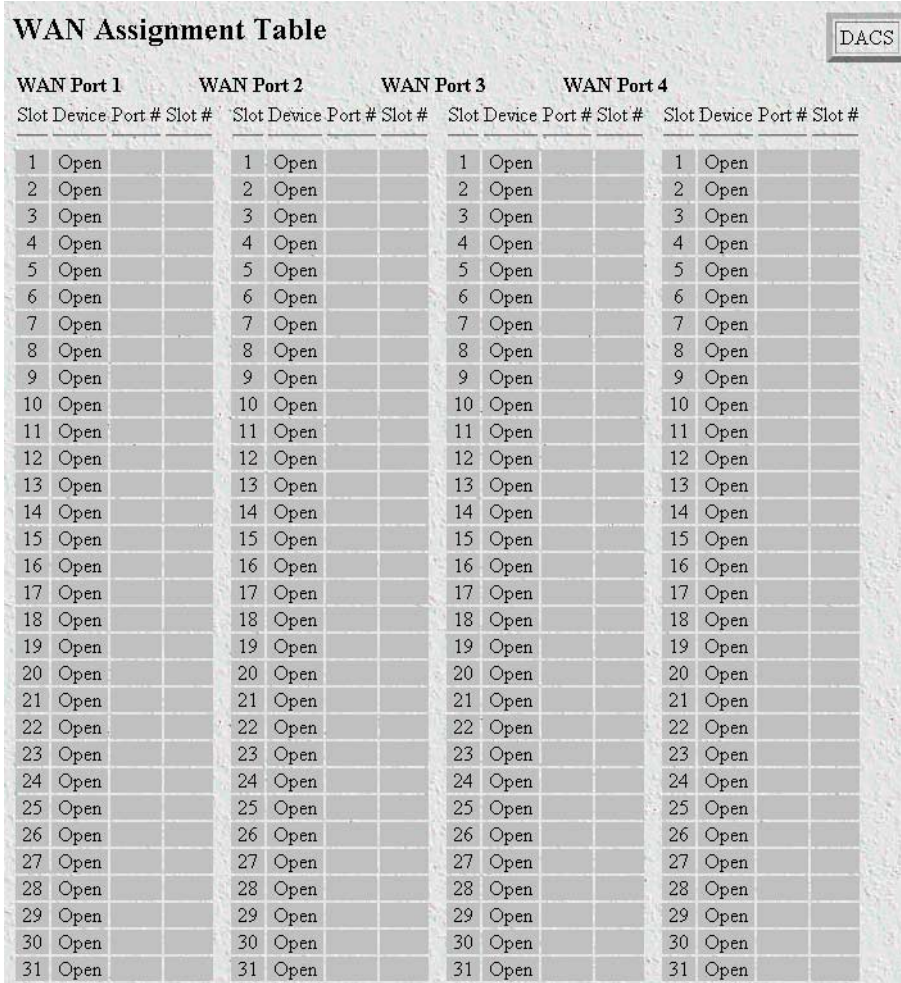
Introduction	138
Displaying the T1/E1 Assignment window.....	138
Slot	139
Device	139
Port #	139
Slot #	139

Introduction

The T1/E1 Assignment subsystem displays the WAN Assignment Table giving the DS0 mapping connection for each time slot for all four WAN ports. At the top of the Web page are four hyperlinks showing the status of each WAN Port's Alarms. For example, the column for *WAN Port 1* shows to which type of device the WAN's time slot connects, the device's Port and Slot number.

Displaying the T1/E1 Assignment window

Click on T1/E1 Assignment link under the Configuration Menu to display the WAN Assignment Table window (figure 52).



WAN Assignment Table												DACS
WAN Port 1			WAN Port 2			WAN Port 3			WAN Port 4			
Slot	Device	Port # Slot #	Slot	Device	Port # Slot #	Slot	Device	Port # Slot #	Slot	Device	Port # Slot #	
1	Open		1	Open		1	Open		1	Open		
2	Open		2	Open		2	Open		2	Open		
3	Open		3	Open		3	Open		3	Open		
4	Open		4	Open		4	Open		4	Open		
5	Open		5	Open		5	Open		5	Open		
6	Open		6	Open		6	Open		6	Open		
7	Open		7	Open		7	Open		7	Open		
8	Open		8	Open		8	Open		8	Open		
9	Open		9	Open		9	Open		9	Open		
10	Open		10	Open		10	Open		10	Open		
11	Open		11	Open		11	Open		11	Open		
12	Open		12	Open		12	Open		12	Open		
13	Open		13	Open		13	Open		13	Open		
14	Open		14	Open		14	Open		14	Open		
15	Open		15	Open		15	Open		15	Open		
16	Open		16	Open		16	Open		16	Open		
17	Open		17	Open		17	Open		17	Open		
18	Open		18	Open		18	Open		18	Open		
19	Open		19	Open		19	Open		19	Open		
20	Open		20	Open		20	Open		20	Open		
21	Open		21	Open		21	Open		21	Open		
22	Open		22	Open		22	Open		22	Open		
23	Open		23	Open		23	Open		23	Open		
24	Open		24	Open		24	Open		24	Open		
25	Open		25	Open		25	Open		25	Open		
26	Open		26	Open		26	Open		26	Open		
27	Open		27	Open		27	Open		27	Open		
28	Open		28	Open		28	Open		28	Open		
29	Open		29	Open		29	Open		29	Open		
30	Open		30	Open		30	Open		30	Open		
31	Open		31	Open		31	Open		31	Open		

Figure 52. WAN Assignment Table window

The WAN Assignment Table window consists of four column groups, one for each WAN port. The top of each column group identifies the WAN port by name, e.g., WAN Port 1: Alarms. This is a hyperlink leading to the

Line Status Alarms web page. The “Line Status Alarms” page gives the status of the T1/E1 WAN port. This is the same web page seen under the T1/E1 Link Activity page in the T1/E1 Link subsystem.

Under each WAN port column group are four columns named Slot, Device, Port #, and Slot #.

Slot

Slot refers to the time slot in the T1/E1 port. Whether you have chosen T1 or E1, all 31 channels will be displayed although in T1, only those numbered 1–24 are applicable.

Device

Device (*daxWAN0DeviceType*, *daxWAN1DeviceType*, *daxWAN2DeviceType*, *daxWAN3DeviceType*) refers to the device type to which the WAN slot connects. The device type options are:

- open(0)
- t1-e1(1)

Note There are four variables for Device Type where *daxWAN0DeviceType* applies to those in WAN Port 1. Similarly *daxWAN1DeviceType* applies to those in WAN Port 2. Likewise for the others.

Port

Port # (*daxWAN0DeviceNumber*, *daxWAN1DeviceNumber*, *daxWAN2DeviceNumber*, *daxWAN3DeviceNumber*) refers to the port number of the Device Type in the second sub-column. Since there are only four WAN ports, the Device Number (Port #) may be chosen from port1(1) to port4(4).

Slot

Slot # (*daxWAN0DeviceSlot*, *daxWAN1DeviceSlot*, *daxWAN2DeviceSlot*, *daxWAN3DeviceSlot*) refer to the slot number (or time slot) of the Device. For t1-e1 Device Types, *Slot#* may vary from 1 to 31.

Chapter 19 **About**

Chapter contents

Introduction	142
Patton Electronics Company contact information	142

Introduction

The **About** link displays Patton Electronics Company contact information (see “Patton Electronics Company contact information”). Click on **About** under the Configuration Menu to display the **About** main window (see figure 53).



Figure 53. About window

Patton Electronics Company contact information

Patton Electronics Company
7622 Rickenbacker Drive
Gaithersburg, Maryland 20879
U.S.A.

Phone: +1 (301) 975-1000

Fax: +1 (301) 869-9293

E-mail: sales@patton.com
support@patton.com

WWW: www.patton.com

Chapter 20 License

Chapter contents

- Introduction144
- End User License Agreement144
 - 1. Definitions:144
 - 2. Title:145
 - 3. Term:145
 - 4. Grant of License:145
 - 5. Warranty:145
 - 6. Termination:145

Introduction

The license link presents the End User License Agreement for the DACS software. Click on License under the Configuration Menu to display the License main window (see figure 54).

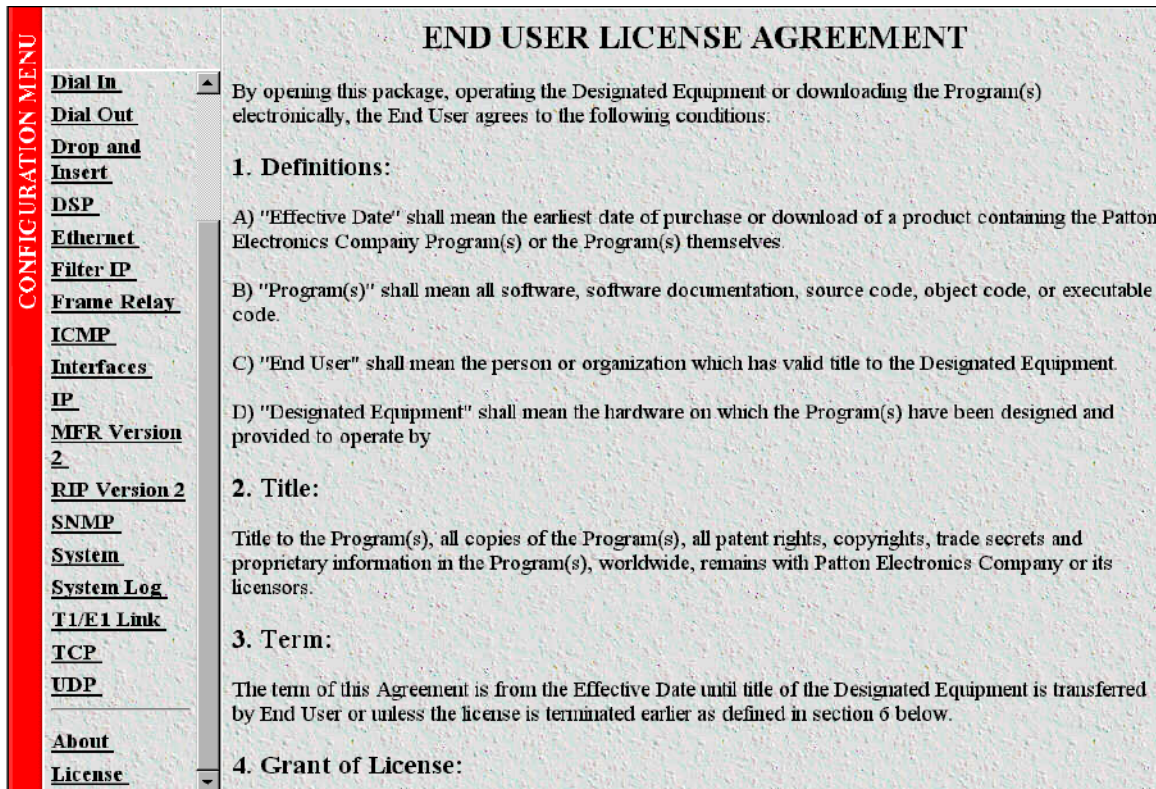


Figure 54. License window

By opening the DACS, operating the Designated Equipment or downloading the Program(s) electronically, the End User agrees to the conditions in the “End User License Agreement” below.

End User License Agreement

By opening this package, operating the Designated Equipment or downloading the Program(s) electronically, the End User agrees to the following conditions:

1. Definitions:

- A) “Effective Date” shall mean the earliest date of purchase or download of a product containing the Patton Electronics Company Program(s) or the Program(s) themselves.
- B) “Program(s)” shall mean all software, software documentation, source code, object code, or executable code.
- C) “End User” shall mean the person or organization which has valid title to the Designated Equipment.
- D) “Designated Equipment” shall mean the hardware on which the Program(s) have been designed and provided to operate by

2. Title:

Title to the Program(s), all copies of the Program(s), all patent rights, copyrights, trade secrets and proprietary information in the Program(s), worldwide, remains with Patton Electronics Company or its licensors.

3. Term:

The term of this Agreement is from the Effective Date until title of the Designated Equipment is transferred by End User or unless the license is terminated earlier as defined in "6. Termination:" below.

4. Grant of License:

A) During the term of this Agreement, Patton Electronics Company grants a personal, non-transferable, non-assignable and non-exclusive license to the End User to use the Program(s) only with the Designated Equipment at a site owned or leased by the End User.

B) The End User may copy licensed Program(s) as necessary for backup purposes only for use with the Designated Equipment that was first purchased or used or its temporary or permanent replacement.

C) The End User is prohibited from disassembling; decompiling, reverse-engineering or otherwise attempting to discover or disclose the Program(s), source code, methods or concepts embodied in the Program(s) or having the same done by another party.

D) Should End User transfer title of the Designated Equipment to a third party after entering into this license agreement, End User is obligated to inform the third party in writing that a separate End User License Agreement from Patton Electronics Company is required to operate the Designated Equipment.

5. Warranty:

The Program(s) are provided "as is" without warranty of any kind. Patton Electronics Company and its licensors disclaim all warranties, either express or implied, including but not limited to the implied warranties of merchantability, fitness for a particular purpose or non-infringement. In no event shall Patton Electronics Company or its licensors be liable for any damages whatsoever (including, without limitation, damages for loss of business profits, business interruption, loss of business information, or other pecuniary loss) arising out of the use of or inability to use the Program(s), even if Patton Electronics Company has been advised of the possibility of such damages. Because some states do not allow the exclusion or limitation of liability for consequential or incidental damages, the above limitation may not apply to you.

If the Program(s) are acquired by or on behalf of a unit or agency of the United States Government, the Government agrees that such Program(s) are "commercial computer software" or "computer software documentation" and that, absent a written agreement to the contrary, the Government's rights with respect to such Program(s) are limited by the terms of this Agreement, pursuant to Federal Acquisition Regulations 12.212(a) and/or DEARS 227.7202-1(a) and/or sub-paragraphs (a) through (d) of the "Commercial Computer Software—Restricted Rights" clause at 48 C.F.R. 52.227-19 of the Federal Acquisition Regulations as applicable.

6. Termination:

A) The End User may terminate this agreement by returning the Designated Equipment and destroying all copies of the licensed Program(s).

B) Patton Electronics Company may terminate this Agreement should End User violate any of the provisions of "4. Grant of License:" above.

C) Upon termination for A or B above or the end of the Term, End User is required to destroy all copies of the licensed Program(s)

