

CA eHealth[®] SystemEDGE

User Guide

r4.3



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

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

Notwithstanding the foregoing, licensed users may print a reasonable number of copies of the documentation for their own internal use, and may make one copy of the related software as reasonably required for back-up and disaster recovery purposes, provided that all CA copyright notices and legends are affixed to each reproduced copy. Only authorized employees, consultants, or agents of the user who are bound by the provisions of the license for the product are permitted to have access to such copies.

The right to print copies of the documentation and to make a copy of the related software is limited to the period during which the applicable license for the Product remains in full force and effect. Should the license terminate for any reason, it shall be the user's responsibility to certify in writing to CA that all copies and partial copies of the Documentation have been returned to CA or destroyed.

EXCEPT AS OTHERWISE STATED IN THE APPLICABLE LICENSE AGREEMENT, TO THE EXTENT PERMITTED BY APPLICABLE LAW, CA PROVIDES THIS DOCUMENTATION "AS IS" WITHOUT WARRANTY OF ANY KIND, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IN NO EVENT WILL CA BE LIABLE TO THE END USER OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE, DIRECT OR INDIRECT, FROM THE USE OF THIS DOCUMENTATION, INCLUDING WITHOUT LIMITATION, LOST PROFITS, BUSINESS INTERRUPTION, GOODWILL, OR LOST DATA, EVEN IF CA IS EXPRESSLY ADVISED OF SUCH LOSS OR DAMAGE.

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

The manufacturer of this Documentation is CA.

Provided with "Restricted Rights." Use, duplication or disclosure by the United States Government is subject to the restrictions set forth in FAR Sections 12.212, 52.227-14, and 52.227-19(c)(1) - (2) and DFARS Section 252.227-7014(b)(3), as applicable, or their successors.

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

Copyright © 2007 CA. All rights reserved.

CA Product References

This document references the following CA products:

- CA eHealth®
- CA eHealth® AdvantEDGE View
- CA eHealth® Application Insight Manager (CA eHealth AIM)
- CA eHealth® Live Health® Application
- CA eHealth® SystemEDGE
- CA Unicenter® Network and Systems Management (CA Unicenter NSM)

Contact Technical Support

For online technical assistance and a complete list of locations, primary service hours, and telephone numbers, contact Technical Support at <http://ca.com/support>.

Contents

Chapter 1: Introduction 19

| | |
|---|----|
| Introducing CA eHealth SystemEDGE | 19 |
| Microsoft Data Center Certification | 19 |
| Using CA eHealth SystemEDGE | 20 |
| Supported MIBs | 20 |
| CA eHealth SystemEDGE Self-Monitoring Features | 23 |
| Identifying Top Processes | 25 |
| Tracking Assets | 25 |
| Supporting Custom MIB Objects | 26 |
| Supporting Windows Registry and Perfmon Extensions | 26 |
| Specifying Corrective Actions | 26 |
| CA eHealth SystemEDGE in Windows Clustered Environment | 27 |
| Using CA eHealth AdvantEDGE View | 28 |
| Using CA eHealth Service Availability | 29 |
| Monitoring Voice and Call Quality | 30 |
| Using the CA eHealth AIMs | 30 |
| Using CA eHealth with CA eHealth SystemEDGE | 31 |
| Using CA eHealth Live Health Application - Fault Manager with CA eHealth SystemEDGE | 32 |
| Guidelines for Using the CA eHealth SystemEDGE Agent | 33 |
| Simple Network Management Protocol | 36 |
| SNMP Message Types | 36 |
| SNMP Version 1 and SNMP Version 2c Communities | 37 |
| SNMP Version 3 User and Key Management | 38 |
| SNMP Traps | 39 |

Chapter 2: Installing the CA eHealth SystemEDGE Agent 41

| | |
|--|----|
| Installing CA eHealth SystemEDGE on Windows Systems | 41 |
| Install the Software with InstallShield | 41 |
| Install the Software from the Command Line | 44 |
| Installing CA eHealth SystemEDGE on Solaris Systems | 45 |
| Install the Software on Solaris | 45 |
| Install CA eHealth SystemEDGE in a Non-Default Directory for Solaris | 48 |
| Installing CA eHealth SystemEDGE on HP-UX Systems | 48 |
| Install the Software on HP-UX | 49 |
| Install CA eHealth SystemEDGE in a Non-Default Directory for HP-UX | 52 |
| Installing CA eHealth SystemEDGE on Linux Systems | 52 |
| Install the Software on Linux | 53 |

| | |
|--|----|
| Install CA eHealth SystemEDGE in a Non-Default Directory for Linux | 54 |
| Installing CA eHealth SystemEDGE on AIX Systems | 54 |
| Install the Software on AIX | 54 |
| Installing CA eHealth SystemEDGE on Tru64 UNIX Systems | 56 |
| Install the Software on Tru64 UNIX | 56 |
| Reviewing the Configuration Files | 57 |
| Configuration Files for UNIX Systems..... | 58 |
| Configuration Files for Windows Systems | 59 |
| Uninstalling the CA eHealth SystemEDGE Agent | 59 |
| Uninstall CA eHealth SystemEDGE for Windows Systems | 59 |
| Uninstall CA eHealth SystemEDGE for UNIX Systems | 61 |

Chapter 3: Configuring the CA eHealth SystemEDGE Agent 63

| | |
|---|----|
| Configuration Files | 63 |
| Interactions Between sysedge.cf and sysedge.mon | 64 |
| Configuring the Agent During the Installation Procedure..... | 64 |
| Before You Begin | 65 |
| Configuration Using sysedge.cf | 65 |
| Configuring System Information | 65 |
| Configuring Access Communities | 66 |
| Configuring SNMPv1 Traps | 68 |
| Configuring Authentication Failure Traps..... | 69 |
| Agent Addresses of Traps from SystemEDGE | 69 |
| Configuring Support for Who Table Information | 70 |
| Configuring Support for User and Group Information | 70 |
| Configuring Support for Remote Shell Capability | 71 |
| Configuring Alternative Syslog Facilities (UNIX Only) | 71 |
| Configuring Alternative Syslog Facilities (Windows Only)..... | 72 |
| Configuring Support for Agent Debugging | 72 |
| Configuring Support for Floppy Status Checking..... | 73 |
| Configuring Support for Serial Port Status Checking | 73 |
| Configuring Support for Disk Probing | 74 |
| Configuring Support for Actions | 74 |
| Disabling Support for Remote File System Checking (UNIX Only)..... | 75 |
| Configuring Support for Threshold Monitoring | 75 |
| Configuring Support for Process Monitoring | 75 |
| Configuring Support for Process Group Monitoring | 77 |
| Configuring Support for Log File Monitoring | 77 |
| Configuring Support for Windows Event Log Monitoring (Windows Only) | 78 |
| Configuring History Collection..... | 78 |
| Configuring User and Group Permissions for Subprograms (UNIX Only) | 78 |
| Configuring the SNMP Bind Address | 79 |

| | |
|--|----|
| Configuring IP Family for SNMP User Datagram Protocol Communications | 79 |
| Enabling Federal Information Processing Standard Mode | 80 |
| Configuring Support for CA eHealth AIMs | 80 |
| Configuring Support for the Monitored Windows AIM | 81 |
| Configuring Support for Linux Free Memory | 81 |
| Recommendations for Configuring Security | 82 |
| Using the SystemEDGE Control Panel for Windows | 83 |

Chapter 4: Starting the CA eHealth SystemEDGE Agent 85

| | |
|---|----|
| Starting the Agent Manually | 85 |
| Start CA eHealth SystemEDGE on Windows Systems | 85 |
| Start CA eHealth SystemEDGE on UNIX Systems | 87 |
| Starting the Agent Automatically at System Boot | 89 |
| Starting the Agent Automatically for Solaris Systems | 89 |
| Starting the Agent Automatically for Windows Systems | 90 |
| Starting the Agent Automatically for HP-UX Systems | 90 |
| Starting the Agent Automatically for Linux Systems | 90 |
| Starting the Agent Automatically for AIX Systems | 91 |
| Starting the Agent Automatically for Tru64 UNIX Systems | 91 |
| Logging Agent Operation Messages | 91 |
| Logging Messages for UNIX | 92 |
| Logging Messages for Windows | 92 |

Chapter 5: Using the CA eHealth SystemEDGE Agent with Other SNMP Agents 93

| | |
|--|----|
| Supporting Multiple SNMP Agents | 93 |
| Agent Multiplexing | 94 |
| Using the CA eHealth SystemEDGE Agent with the Solaris Solstice Enterprise Agent | 95 |
| Using the CA eHealth SystemEDGE Agent with the Microsoft Windows SNMP Agent | 96 |
| Using the CA eHealth SystemEDGE Agent with the HP SNMP Agent | 97 |
| Using the CA eHealth SystemEDGE Agent with the AIX SNMP Agent | 97 |
| Using the CA eHealth SystemEDGE Agent with the Tru64 UNIX SNMP Agent | 98 |
| Using the CA eHealth SystemEDGE Agent with the Compaq Insight Manager | 98 |

Chapter 6: Systems Management MIB 99

| | |
|--|-----|
| Systems Management MIB Information | 100 |
| Host System Information | 101 |
| Mounted Devices | 101 |
| File System Space Monitoring | 102 |
| Unmount a Mounted Device | 102 |
| Kernel Configuration | 103 |

| | |
|---|-----|
| Boot Configuration | 104 |
| Streams Group | 104 |
| User Information | 106 |
| Group Information | 107 |
| Process Information | 108 |
| Change the nice Value of a Process (UNIX only)..... | 109 |
| Send a Signal to a Process | 109 |
| Who Table Information | 110 |
| Remote Command Execution | 111 |
| Execute a Remote Command | 111 |
| Kernel Performance Statistics | 111 |
| Interprocess Communication: Queues, Shared Memory, and Semaphores | 113 |
| Deleting an Interprocess Communication | 114 |
| Message Buffer Allocation and Usage Statistics | 115 |
| Stream Buffers | 116 |
| I/O Buffer Cache | 117 |
| RPC Group | 118 |
| NFS Group | 118 |
| Windows-Specific Groups | 120 |
| NT System Group | 120 |
| NT Thread Group | 122 |
| NT Registry Group | 123 |
| NT Service Group | 124 |
| NT System Performance Group | 125 |
| NT Cache Performance Group | 126 |
| NT Memory Performance Group | 128 |
| NT Page File Performance Group | 130 |
| NT Event Monitor Group | 130 |
| NT Registry and Performance Extension Group | 131 |
| Unsupported MIB Objects on Windows | 132 |
| Monitor Table | 134 |
| Process Monitor Table | 135 |
| Process Group Monitor Table | 135 |
| Log Monitor Table | 136 |
| History Table | 136 |
| History Sampling Examples | 137 |
| Disk Statistics Group | 139 |
| Enabling Collection of Disk-Performance Statistics | 140 |
| CPU Statistics Group | 141 |
| Extension Group | 142 |

Chapter 7: Private Enterprise Traps **143**

| | |
|---------------------------------|-----|
| Format of Trap PDUs | 143 |
| monitor Trap | 144 |
| monitorEntryNotReady Trap | 145 |
| logMonMatch Trap | 146 |
| logMonNotReady Trap | 147 |
| ntEventMonMatch Trap | 148 |
| ntEventMonNotReady Trap | 149 |
| monitorClear Trap | 150 |
| processStop Trap | 151 |
| processStart Trap | 152 |
| processThreshold Trap | 153 |
| processClear Trap | 154 |
| license Trap | 155 |
| addrChangeTrap | 155 |
| procGroupChangeTrap | 156 |
| SNMPv1 Trap Format | 156 |

Chapter 8: Host Resources MIB **159**

| | |
|--|-----|
| Host Resources System Group | 160 |
| Host Resources Storage Group | 161 |
| Host Resources Device Group | 162 |
| Device Table | 162 |
| Processor Table | 163 |
| Disk Storage Table | 164 |
| Partition Table | 165 |
| File System Table | 166 |
| Host Resources Running Software Group | 167 |
| Host Resources Installed Software Group | 168 |
| Unsupported MIB Objects on Windows Systems | 169 |

Chapter 9: Configuring Threshold Monitoring **171**

| | |
|---|-----|
| Threshold Monitoring | 171 |
| The Monitor Table | 172 |
| Sample Entry in the Monitor Table | 172 |
| Columns of the Monitor Table | 173 |
| Optimizing Row Creation | 177 |
| Monitor Table Flags | 178 |
| Monitor Table Actions | 181 |
| Monitor Entry Correlation | 182 |

| | |
|--|-----|
| View the Monitor Table with CA eHealth AdvantEDGE View | 184 |
| Assigning Entry Rows for the Monitor Table | 184 |
| Setting Local Policy | 184 |
| Reserve Blocks of Rows | 185 |
| Configuring the Monitor Table | 185 |
| Initial Configuration During Startup | 186 |
| Dynamic Configuration During Operation | 187 |
| monitor Directive--Add Entries to the Monitor Table | 188 |
| Threshold Monitoring Examples | 190 |
| edgemon Utility--Monitor Thresholds | 202 |
| edgemon Commands for Threshold Monitoring | 205 |
| edgemon Examples | 207 |
| Removing Threshold Monitoring Entries | 208 |
| Removing Entries from the sysedge.cf File | 209 |
| Removing Entries with the edgemon Utility | 209 |
| Remove Entries Manually | 209 |

Chapter 10: Configuring Process and Service Monitoring 211

| | |
|--|-----|
| Monitoring Processes and Windows Services | 211 |
| Monitoring Windows Services | 212 |
| Sample Process Monitor Table Entry | 212 |
| The Process Monitor Table | 213 |
| Columns of the Process Monitor Table | 214 |
| Process Attributes | 217 |
| Optimizing Row Creation | 219 |
| Process Monitor Table Flags | 220 |
| Process Monitor Table Actions | 226 |
| View the Process Monitor Table with CA eHealth AdvantEDGE View | 227 |
| Assigning Entry Rows for the Process Monitor Table | 228 |
| Configuring the Process Monitor Table | 228 |
| Dynamic Configuration During Operation | 228 |
| Initial Configuration During Startup | 229 |
| Monitoring a Process to Make Sure It Is Running | 230 |
| watch process Directive--Monitor Process Attributes | 232 |
| Process Monitoring Examples | 234 |
| edgemon Utility--Monitor Processes | 237 |
| edgemon Commands for Process Monitoring | 240 |
| edgemon Examples | 242 |
| Removing Process Monitoring Entries | 245 |
| Removing Entries from the sysedge.cf File | 245 |
| Removing Entries with the edgemon Utility | 245 |
| Remove Entries Manually | 246 |

| | |
|--|-----|
| Recommendations for Process and Service Monitoring | 247 |
|--|-----|

Chapter 11: Configuring Process Group Monitoring 249

| | |
|--|-----|
| Monitoring Process Groups | 249 |
| The Process Group Monitor Table | 250 |
| Columns of the Process Group Monitor Table | 250 |
| Optimizing Row Creation | 255 |
| Process Group Monitor Table Flags | 255 |
| Process Group Monitor Table Actions | 257 |
| View the Process Group Monitor Table with CA eHealth AdvantEDGE View | 257 |
| Assigning Entry Rows for the Process Group Monitor Table | 258 |
| Configuring the Process Group Monitor Table | 258 |
| Dynamic Configuration During Operation | 258 |
| Initial Configuration During Startup | 259 |
| watch procgroup Directive--Monitor a Process Group | 259 |
| Process Group Monitoring Examples | 260 |
| Removing Process Group Monitoring Entries | 261 |
| Removing Entries from the sysedge.cf File | 261 |
| Removing Entries with the edgemon Utility | 262 |
| Remove Entries Manually | 262 |

Chapter 12: Configuring Log File Monitoring 263

| | |
|--|-----|
| Monitoring Log Files | 263 |
| Log Monitor Table | 264 |
| Columns of the Log Monitor Table | 265 |
| Optimizing Row Creation | 267 |
| Log Monitor Table Flags | 267 |
| Log Monitor Table Actions | 270 |
| View the Log Monitor Table with CA eHealth AdvantEDGE View | 272 |
| Configuring the Log Monitor Table | 272 |
| Initial Configuration During Start-Up | 273 |
| Dynamic Configuration During Operation | 275 |
| edgewatch Utility--Monitor Log Files | 275 |
| edgewatch Commands for Log File Monitoring | 279 |
| Sample Uses of the edgewatch Utility | 280 |
| Removing Log Monitoring Entries | 282 |
| Removing Entries from the sysedge.cf File | 282 |
| Removing Entries with the edgewatch Utility | 282 |
| Remove Entries Manually | 283 |
| Recommendations for Log File Monitoring | 283 |
| Monitor Log File Size | 286 |

| | |
|---|------------|
| Rotating Log Files | 288 |
| Chapter 13: Configuring Windows Event Monitoring | 289 |
| Monitoring Windows Events | 289 |
| Monitoring Windows Event Logs | 290 |
| Checking Log File Status | 290 |
| Search Criteria | 290 |
| NT Event Monitor Table | 291 |
| Columns of the NT Event Monitor Table | 292 |
| Optimizing Row Creation | 294 |
| NT Event Monitor Table Flags | 295 |
| NT Event Monitor Table Actions | 297 |
| View the NT Event Monitor Table with CA eHealth AdvantEDGE View | 298 |
| Configuring the NT Event Monitor Table | 299 |
| Dynamic Configuration During Operation | 299 |
| Initial Configuration During Start-Up | 299 |
| edgwatch Utility--Monitor Windows Events | 302 |
| edgwatch Commands for Windows Event Monitoring | 306 |
| Sample Uses of edgwatch for Monitoring Windows Events | 308 |
| Removing NT Event Monitoring Entries | 309 |
| Removing Entries from the sysedge.cf File | 309 |
| Removing Entries with the edgemon Utility | 310 |
| Remove Entries Manually | 310 |
| Chapter 14: Configuring History Collection | 311 |
| History Collection | 311 |
| History Sampling | 311 |
| History Control Table and the Data Table | 312 |
| Columns of the History Control Table | 312 |
| Columns of the History Table | 314 |
| Optimizing Row Creation | 315 |
| View the History Control Table with CA eHealth AdvantEDGE View | 316 |
| Configuring the History Control Table | 317 |
| Initial Configuration During Start-Up | 317 |
| Dynamic Configuration During Operation | 318 |
| emphistory Utility--Manage Entries in History Control Table | 319 |
| emphistory Commands for Managing Entries in the History Control Table | 322 |
| emphistory Utility Examples | 324 |

Chapter 15: Adding Custom MIB Objects 327

| | |
|--|-----|
| Systems Management MIB Extension Group | 327 |
| Features of the Extension Group | 328 |
| Configuring Extension Variables | 329 |
| extension Keyword--Add Entries to the Extension Group | 329 |
| Additional Parameters | 330 |
| Extension Examples | 330 |
| Writing Extension Scripts | 332 |
| Testing Your Script: An Example | 332 |
| Using Extension Variables with Your Management Software | 333 |
| How to Edit empire.asn1 for Extension Variables | 333 |
| How to Edit a Separate MIB Specification for Extension Variables | 334 |
| Recommendations for Using Extensions | 334 |

Chapter 16: Adding Windows Registry and Performance MIB Objects 335

| | |
|--|-----|
| Systems Management MIB ntRegPerf Group | 335 |
| Windows Registry and Performance Functionality | 336 |
| Registry Data | 337 |
| Performance Data | 337 |
| Configuring Windows Registry and Performance Variables | 339 |
| ntRegPerf Keyword--Add Entries to the ntregperf Group | 340 |
| Windows Registry and Performance Examples | 341 |
| Using Windows Registry and Performance Variables with Your Management Software | 342 |
| How to Edit empire.asn1 for ntRegPerf Variables | 342 |
| How to Add a Separate MIB Specification for ntRegPerf Variables | 343 |

Chapter 17: Deploying the CA eHealth SystemEDGE Agent 345

| | |
|--|-----|
| Introduction | 345 |
| Deploy CA eHealth SystemEDGE with CA eHealth AdvantEDGE View | 346 |
| How the Automated Deployment Works | 346 |
| Deploy CA eHealth SystemEDGE from the Web | 347 |
| Deploy CA eHealth SystemEDGE through Email | 348 |
| Third-Party Deployment Tools | 348 |
| How to Automate Deployment | 348 |
| Making Software Available to Remote Systems | 349 |
| Installing Software on Remote Systems | 350 |
| Configuring Software for Distributed Systems | 350 |
| Security Issues | 351 |

Chapter 18: Command Line Utilities 353

| | |
|--|-----|
| SNMP Command Line Utilities | 353 |
| diagsysedge.exe Utility--Troubleshooting the Agent | 354 |
| edgemon Utility--Monitor Thresholds | 359 |
| edgwatch Utility--Monitor Processes | 363 |
| emphistory Utility--Manage Entries in History Control Table | 367 |
| se_enc Utility--Encrypt the SNMPV3 Configuration File | 371 |
| sendtrap Utility--Send a SNMP UDP Trap | 372 |
| snmpget Utility--Retrieve an OID value | 380 |
| snmpset Utility--Set Value of an OID | 384 |
| sysvariable Utility--Retrieve a System Value | 390 |
| walktree Utility--Retrieve Values of OID Tree | 395 |
| xtrapmon Utility--Capture SNMP Traps | 399 |
| Additional Command Line Utilities | 403 |
| bounce.exe Utility--Forcibly Reboot the System (Windows Only) | 404 |
| checkfile.exe Utility--Display the File Size | 405 |
| email.exe Utility--Send an Email | 406 |
| getver.exe Utility--Display File Information (Windows Only) | 407 |
| nt4bigmem.exe Utility--Display Memory Information (Windows Only) | 408 |
| restartproc.exe Utility--Restart a Process (Windows Only) | 408 |
| restartsvc.exe Utility--Restart a Service (Windows Only) | 410 |
| restartproc.sh Utility--Restart a Process (UNIX Only) | 411 |

Chapter 19: Troubleshooting and Usage Suggestions 413

| | |
|---|-----|
| Using diagsysedge.exe | 413 |
| Determine Whether the Agent Is Running | 413 |
| Obtain a Report for Troubleshooting | 414 |
| Common Problems and Questions | 415 |
| Agent Not Responding to SNMP Requests | 415 |
| Management System Not Receiving SNMP Trap Messages | 418 |
| Agent Does Not Run on A Particular Operating System Version | 419 |
| Bind Failed: Address Already In Use | 419 |
| Update the Monitor Configuration File | 420 |
| How to Automatically Restart Processes | 421 |
| Implementing Trap Severity Levels | 422 |
| Required and Recommended System Patches | 422 |

Appendix A: Error Messages 423

| | |
|--|-----|
| CA eHealth SystemEDGE Agent Error Messages | 423 |
| Command-line Utility Error Messages | 471 |

| | |
|--------------------------------|-----|
| Common Error Messages | 471 |
| edgemon Error Messages | 478 |
| edgewatch Error Messages | 478 |
| sendtrap Error Messages | 480 |
| walktree Error Messages | 483 |
| xtrapmon Error Messages | 483 |

Appendix B: Using the syslog Facility **487**

| | |
|--|-----|
| Logging syslog Messages | 487 |
| Creating a Log File for Daemon Messages | 489 |
| Create a Daemon Log File for Solaris SPARC Systems | 489 |
| Create a Daemon Log File for HP-UX Systems | 489 |
| Create a Daemon Log File for AIX Systems | 490 |
| Create a Daemon Log File for Linux Systems | 490 |

Appendix C: Adding Self-Monitoring Entries to the sysedge.mon File **491**

| | |
|--|-----|
| CA eHealth SystemEDGE Table Backing Store | 491 |
| Adding Monitor Table Entries to the sysedge.mon File | 492 |
| Sample Monitor Table Entries in sysedge.mon | 495 |
| Adding Process Monitor Table Entries to the sysedge.mon File | 496 |
| Sample Process Monitor Entries in sysedge.mon | 498 |
| Adding Process Group Monitor Table Entries to the sysedge.mon File | 499 |
| Sample Process Group Monitor Entry in sysedge.mon | 501 |
| Adding Log Monitor Table Entries to the sysedge.mon File | 501 |
| Sample Log Monitor Entry in sysedge.mon | 503 |
| Adding NT Event Monitor Table Entries to the sysedge.mon File | 503 |
| Sample NT Event Monitor Entries in sysedge.mon | 505 |
| Adding History Control Table Entries to the sysedge.mon File | 506 |
| Sample History Control Table Entries in sysedge.mon | 507 |

Appendix D: Textual Conventions for Row Status **509**

| | |
|--|-----|
| RFC 1443: Textual Conventions for SNMPv2 | 509 |
| Conceptual Row Creation | 513 |
| Interaction 1: Selecting an Instance-Identifier | 513 |
| Interaction 2: Creating the Conceptual Row | 514 |
| Interaction 3: Initializing Non-defaulted Objects | 516 |
| Interaction 4: Making the Conceptual Row Available | 517 |
| Conceptual Row Suspension | 517 |
| Conceptual Row Deletion | 518 |

Appendix E: SNMPv3 in CA eHealth SystemEDGE **519**

| | |
|--|-----|
| SNMPv3 Configuration | 519 |
| Modifying the SNMPv3 Configuration File | 520 |
| SNMPv3 User Configuration | 520 |
| Address Filtering for SNMPv3 Users | 523 |
| Configuring SNMPv2c/SNMPv3 Traps | 526 |
| Agent Addresses of Traps from SystemEDGE | 527 |
| Configuring FIPS 140-2 Mode | 528 |
| Encrypt the SNMPv3 Configuration File | 528 |
| Disable SNMPv1/SNMPv2c | 528 |
| Command Line Utilities using SNMPv3 | 529 |

Appendix F: Using the Monitored Windows AIM **531**

| | |
|--|-----|
| Monitored Windows AIM | 531 |
| Operation of the Monitored Windows AIM | 531 |
| Limitations of the Monitored Windows AIM | 533 |

Appendix G: FIPS 140-2 Encryption **535**

| | |
|--|-----|
| FIPS 140-2 Mode | 535 |
| Installing FIPS Libraries | 535 |
| Platform Support | 536 |
| Supported Encryption Protocols | 536 |
| Supported Authentication Protocols | 536 |
| Configuring FIPS 140-2 Mode | 537 |
| FIPS Mode Considerations | 538 |
| Protecting Keys in SystemEDGE | 538 |

Appendix H: CA eHealth Advanced Encryption **539**

| | |
|---|-----|
| CA eHealth Advanced Encryption | 539 |
| Supported Platforms | 539 |
| Supported Encryption Protocols | 540 |
| Supported Authentication Protocols | 540 |
| FIPS Compatibility | 540 |
| Prerequisites for Installation | 540 |
| Installing CA eHealth Advanced Encryption | 540 |
| Install CA eHealth Advanced Encryption on Windows | 541 |
| Uninstall CA eHealth Advanced Encryption on Windows | 541 |
| Install CA eHealth Advanced Encryption on UNIX | 541 |
| Uninstall CA eHealth Advanced Encryption on UNIX | 542 |
| Installed Files | 542 |

| | |
|--|-----|
| CA eHealth SystemEDGE Considerations | 543 |
|--|-----|

| | |
|--------------|------------|
| Index | 545 |
|--------------|------------|

Chapter 1: Introduction

CA eHealth SystemEDGE increases the productivity of system administration staff by enabling them to control all workstations on their networks from a single, central location. The agent extends management beyond the network boundary and into attached systems to automate systems management tasks and inventory tracking, increasing productivity and system stability while helping to reduce rising system support costs. You can use the CA eHealth SystemEDGE agent to distribute management tasks to the host systems.

This guide is intended for an administrator who installs, configures, and uses the CA eHealth SystemEDGE agent to manage UNIX and Windows workstations. It assumes that you have a basic familiarity with your system's operating system environment and with the Simple Network Management Protocol (SNMP).

This section contains the following topics:

[Introducing CA eHealth SystemEDGE](#) (see page 19)

[Using CA eHealth SystemEDGE](#) (see page 20)

[Using CA eHealth AdvantEDGE View](#) (see page 28)

[Using CA eHealth Service Availability](#) (see page 29)

[Monitoring Voice and Call Quality](#) (see page 30)

[Using the CA eHealth AIMS](#) (see page 30)

[Using CA eHealth with CA eHealth SystemEDGE](#) (see page 31)

[Using CA eHealth Live Health Application - Fault Manager with CA eHealth SystemEDGE](#) (see page 32)

[Guidelines for Using the CA eHealth SystemEDGE Agent](#) (see page 33)

[Simple Network Management Protocol](#) (see page 36)

Introducing CA eHealth SystemEDGE

The CA eHealth SystemEDGE agent provides powerful system management through the industry-standard SNMP. It enables remote management systems to access important information about the system's configuration, status, performance, users, processes, file systems and much more. In addition, the agent includes intelligent self-monitoring capabilities that enable reporting and managing of exceptions and that eliminate the need for excessive polling.

Microsoft Data Center Certification

The CA eHealth SystemEDGE agent does not touch the system kernel, whether it runs on a UNIX, Linux, or Windows system. CA eHealth SystemEDGE does not require Data Center Certification.

Using CA eHealth SystemEDGE

To use the CA eHealth SystemEDGE agent, you must first install it on every workstation or server that you want to monitor. You can then configure it to monitor that system for variables that you specify. The CA eHealth SystemEDGE agent interoperates with SNMP network management system (NMS) platforms, such as CA eHealth, CA SPECTRUM, Sun Domain Manager, HP OpenView, IBM NetView 6000, Micromuse Netcool, and others. In addition, the CA eHealth SystemEDGE agent supports the ability to monitor objects from several management information bases (MIBs).

Supported MIBs

A MIB is a virtual information store in which an agent stores information about the elements under its control. Each item of management information is represented by an object, and the MIB is a structured collection of these objects.

A management system monitors a managed resource by reading the values of its MIB objects. It can also control the resource by modifying (setting) the values of objects in the resource's MIB through SNMP commands.

MIBs are defined in a MIB specification that describes the management objects relating to a particular resource. The MIB specification also defines how the collection of objects is structured. The MIB module resembles a data-definition document used by both the management system and the agent.

The CA eHealth SystemEDGE agent supports the following MIBs:

- MIB-II (RFC 1213)
- Host Resources MIB (RFC 1514)
- Systems Management MIB

MIB II

MIB-II is the standard that provides information about network interfaces and protocol statistics. This MIB includes information about the following protocols:

- Internet Protocol (IP)
- Transfer Control Protocol (TCP)
- Internet Control Message Protocol (ICMP)
- User Datagram Protocol (UDP)
- Simple Network Management Protocol (SNMP)

IPv6 MIB Tables

CA eHealth SystemEDGE supports the IPv6 MIB tables shown in the table below. Supported MIB tables vary by platform.

| MIB Table | Windows | Solaris | HP-UX | AIX | Linux | Tru64 |
|---------------------------------|---------|---------|-------|-----|-------|-------|
| ipSystemStatsTable (RFC 4293) | No | Yes | No | Yes | Yes | Yes |
| ipIfStatsTable (RFC 4293) | No | No | Yes | No | No | No |
| ipAddressPrefixTable (RFC 4293) | No | No | Yes | No | Yes | No |
| ipNetToPhysicalTable (RFC 4293) | No | Yes | Yes | No | No | No |
| ipAddressTable (RFC 4293) | No | Yes | Yes | Yes | Yes | Yes |
| ipDefaultRouterTable (RFC 4293) | No | Yes | Yes | Yes | Yes | Yes |
| icmpStatsTable (RFC 4293) | Yes | Yes | No | Yes | Yes | Yes |
| tcpConnectionTable (RFC 4293) | Yes | Yes | Yes | Yes | Yes | Yes |
| udpEndpointTable (RFC 4293) | Yes | Yes | Yes | Yes | Yes | Yes |

Host Resources MIB

The Host Resources MIB is defined by the Internet Engineering Task Force (IETF) to provide management information for generic host systems. The MIB includes information especially useful for asset management, such as the following:

- Storage areas, such as file systems and disk partitions
- Running and installed software
- System devices, such as keyboards, disks, and network cards

Note: For more information about how the CA eHealth SystemEDGE agent uses the Host Resources MIB, see the chapter “Host Resources MIB”.

Systems Management MIB

The Systems Management MIB is a private-enterprise MIB that includes objects for monitoring the health and performance of the underlying system and its applications. This MIB defines management information for the following:

- Kernel and system parameters
- Boot configuration
- Network, streams, and I/O buffer statistics
- Network file system (NFS) and Remote Procedure Call (RPC) statistics
- Kernel performance statistics, such as the number of context switches and page faults
- File systems
- Mounted devices
- Users
- Processes
- Interprocess communications
- System resources

The following list describes the self-monitoring tables provided in the Systems Management MIB. These tables configure the CA eHealth SystemEDGE agent's autonomous self-monitoring and data-storage capabilities.

Monitor table

Contains MIB objects that the CA eHealth SystemEDGE agent monitors and compares to user-specified thresholds.

Process Monitor table

Contains processes that the CA eHealth SystemEDGE agent monitors for status (whether they are running) and resource utilization.

Process Group Monitor table

Contains groups of processes that the CA eHealth SystemEDGE agent monitors for status and resource utilization.

Log Monitor table

Contains regular expression strings for which the CA eHealth SystemEDGE agent searches through user-specified log files.

NT Event Monitor table

Contains event logs that the CA eHealth SystemEDGE agent searches for specific events.

History Control table

Contains the sample interval and number of samples for MIB objects that the CA eHealth SystemEDGE agent monitors and stores in the History table for future retrieval by the management system.

Note: For more information about how the CA eHealth SystemEDGE agent uses the Systems Management MIB, see the chapter, "Systems Management MIB".

CA eHealth SystemEDGE Self-Monitoring Features

When you manage a large enterprise network with hundreds of systems, you may need to put limits on the information monitored, the poll rate, and even the number of systems managed. The self-monitoring capability of the CA eHealth SystemEDGE agent provides the kind of management by exception which is necessary in distributed network environments.

The CA eHealth SystemEDGE agent provides the following types of monitoring:

- Threshold monitoring
- Process and service monitoring
- Process group monitoring
- Log file monitoring
- Windows event monitoring
- History collection

Threshold Monitoring

The CA eHealth SystemEDGE agent can monitor exception conditions automatically, reducing or eliminating the need for constant polling by a network management system (NMS). You can configure the agent's flexible Monitor table to monitor any integer-based MIB object that the agent supports. You set the polling interval, comparison operator (greater than, equal to, and so on), and threshold value, and CA eHealth SystemEDGE automatically monitors the MIB objects that you specify. You can tailor entries for time over threshold to reduce noise. CA eHealth SystemEDGE can also send traps to an NMS if exceptions occur.

For example, you can configure CA eHealth SystemEDGE to monitor the available space on a particular file system and to notify the NMS when the file system becomes too full.

Note: For more information, see the chapter, "Configuring Threshold Monitoring".

Process and Service Monitoring

With CA eHealth SystemEDGE, you can monitor process attributes for mission-critical processes, Windows services, and applications. For example, you can monitor whether a process is running, the network I/O, system calls, and other attributes.

If any processes stop running, CA eHealth SystemEDGE can automatically notify the NMS and restart them, if necessary. You can configure CA eHealth SystemEDGE to monitor processes in the Process Monitor table. On Windows systems, CA eHealth SystemEDGE can also monitor Windows services.

Note: For more information, see the chapter, "Configuring Process and Service Monitoring".

Process Group Monitoring

You can use the Process Group Monitor table to define a set of processes that the CA eHealth SystemEDGE agent can track. CA eHealth SystemEDGE can track the number of processes (by name and arguments) that match the regular expression you specified. It can also indicate when a process group changes through the processGroupChange trap. In addition, it can match processes by user name and group name.

Note: For more information, see the chapter, "Configuring Process Group Monitoring".

Log File Monitoring

CA eHealth SystemEDGE can monitor any ASCII-based system or application log file for regular expressions. For example, you can configure the CA eHealth SystemEDGE agent to monitor the log file `/var/adm/sulog` for a regular expression that you specify. Whenever a message that matches the regular expression you specified is logged to the file, CA eHealth SystemEDGE notifies the NMS through an SNMP trap and includes the log entry that matched the expression.

Note: For more information, see the chapter, "Configuring Log File Monitoring".

Windows Event Monitoring

CA eHealth SystemEDGE can also monitor Windows event logs for important event types, event identifiers, or events that match specific regular expressions. Whenever an event that matches the search criteria occurs, CA eHealth SystemEDGE notifies the NMS through an SNMP trap.

Note: For more information, see the chapter "Configuring Windows Event Monitoring".

History Collection

You can configure the CA eHealth SystemEDGE History Control Table to sample MIB variables and to use the collected data for baselining and trend analysis of your system without having to constantly poll from the NMS. CA eHealth SystemEDGE collects the data, and the NMS can periodically retrieve the history.

Note: For more information about configuring CA eHealth SystemEDGE history collection, see the chapter, "Configuring History Collection".

Note: CA eHealth SystemEDGE history collection capability is short-term only. Use eHealth for long-term history collection. For more information about using eHealth to monitor your systems, see the *CA eHealth System and Application Administration Guide*.

Identifying Top Processes

CA eHealth SystemEDGE provides a flexible architecture that supports the addition of plug-in modules for monitoring processes and applications. One of these plug-ins is the Top Processes application insight module (AIM), through which the agent can report on processes which consume the most CPU resources at any given time. You can use Top Processes to detect and isolate the CPU-dominating processes. Then, you can reallocate resources for optimal system and application availability and performance. The Top Processes AIM ships with the CA eHealth SystemEDGE agent. You can enable this AIM during the CA eHealth SystemEDGE agent installation.

Tracking Assets

You can use CA eHealth SystemEDGE to automate asset tracking and provide a current picture of your installed hardware and software. CA eHealth SystemEDGE can determine whether your systems are properly configured and whether they include the current patches and service packs. This information can help simplify system management, improve performance, and reduce security risks.

Note: For more information about tracking assets, see Inventory Tracking and Asset Management in the chapter "Host Resources MIB."

Supporting Custom MIB Objects

The CA eHealth SystemEDGE agent enables you to create your own scalar MIB objects for customized management. You can configure CA eHealth SystemEDGE with each MIB object's number and type and the name of a script or program to run when the new MIB variable is queried or set.

Note: For more information, see the chapter, "Adding Custom MIB Objects".

Supporting Windows Registry and Perfmon Extensions

You can also configure the CA eHealth SystemEDGE agent for Windows to report additional registry parameters and performance data without using external programs or scripts. This feature enable you to monitor the health of applications that make performance data available through the performance registry.

Note: For more information, see the chapter, "Adding Windows Registry and Performance MIB Objects".

Specifying Corrective Actions

The CA eHealth SystemEDGE agent can automatically respond to critical situations on a system by invoking actions, which are specific commands or scripts that the agent can run automatically when configured to do so. For example, you can specify actions that enable the agent to restart a failed process, send email to an administrator in the event of an unauthorized access to the system, and so on. You can also configure the CA eHealth SystemEDGE agent to perform corrective actions in response to traps. For example, you can configure the agent to run a script or program when a variable's value crosses a specified threshold, or when the agent discovers specific matches on regular expressions in log files or Windows event logs.

When using actions, you must specify the full path of the command (with any parameters). The agent runs this command each time the conditions are met for the monitoring table entry. If you do not specify an action, the agent does not call a command or script in response to meeting the conditions you specified. For sample actions, see the contrib subdirectory of the CA eHealth SystemEDGE agent installation and the CA eHealth SystemEDGE contributed information Web page on the CA eHealth Support Web site (support.concord.com).

Note: Do not use Windows batch files for actions; they impose severe programmatic limitations and often do not work correctly with desktop applications. Instead, use a more powerful and flexible scripting language, such as Perl or Visual Basic.

For more information about specifying actions, see the following sections:

- Monitor Table Actions in the chapter "Configuring Threshold Monitoring"
- Process Monitor Table Action in the chapter "Configuring Process and Service Monitoring"
- Process Group Monitor Table Actions in the chapter "Configuring Process Group Monitoring"
- Log Monitor Table Actions in the chapter "Configuring Log File Monitoring"
- NT Event Monitor Table Actions in the chapter "Configuring Windows Event Monitoring"

CA eHealth SystemEDGE in Windows Clustered Environment

Clusters are groups of servers and other resources that function as a single system to enable high availability and shared workload. Clusters can protect against failure of applications, services, or hardware (including CPUs and disk drives).

The CA eHealth SystemEDGE agent can operate in a Windows cluster to monitor individual components on the physical servers in Windows clusters based on MIB objects in the Systems Management MIB (empire.asn1).

CA eHealth SystemEDGE's Windows event monitoring can send a trap upon a cluster failover event.

The Systems Management MIB provides basic information about the cluster with the following MIB objects:

- ntIsClustered
- ntClusterName
- ntClusterMembers
- ntClusterIsActive
- ntClusterActiveNode

For more information about these objects, refer to the Systems Management MIB (empire.asn1) in the doc subdirectory of your CA eHealth SystemEDGE agent's installation.

You can view cluster data by running a CA AdvantEDGE View System Information query. You can also use CA AdvantEDGE View to apply a custom CA eHealth SystemEDGE template for monitoring clusters to your systems. For more information, see the CA AdvantEDGE View Web Help.

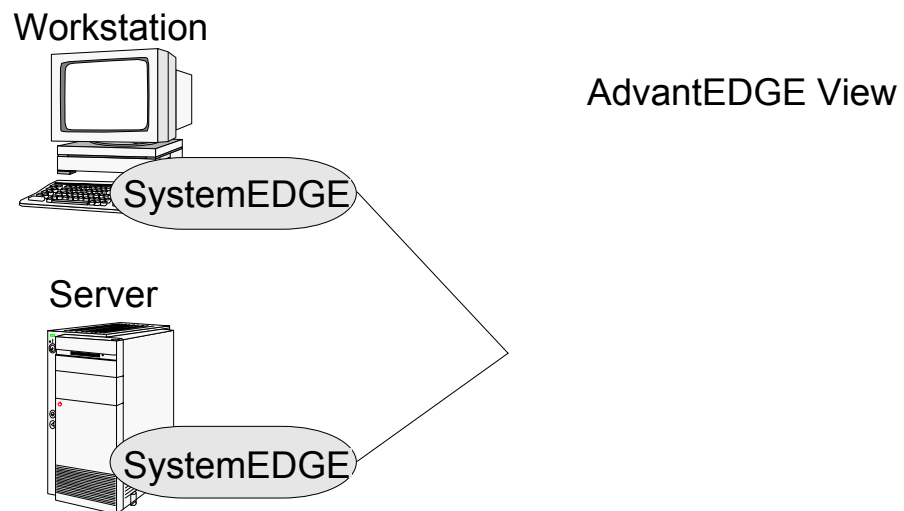
Note: While CA eHealth SystemEDGE can monitor the health of individual servers, monitoring clustered environments requires additional intelligence to distinguish between a failure or failover. CA eHealth SystemEDGE cannot be used in these environments.

To monitor clustered environments, use CA Unicenter NSM with its Advanced Systems Management option.

Using CA eHealth AdvantEDGE View

If you are using the CA eHealth SystemEDGE agent with the CA eHealth AdvantEDGE View element manager, you can run queries on the data collected by CA eHealth SystemEDGE agents through the CA eHealth AdvantEDGE View Web-based graphical user interface. CA eHealth AdvantEDGE View can also automate deployment and configuration of your CA eHealth SystemEDGE agents.

You can access CA eHealth AdvantEDGE View from the Systems & Apps tab of the CA eHealth Web interface. The following illustration shows the CA eHealth AdvantEDGE View interface.



Using CA eHealth Service Availability

CA eHealth Service Availability is a plug-in to the CA eHealth SystemEDGE agent that provides management and monitoring of the response time and availability of the following Internet services:

- Active Directory
- Dynamic Host Configuration Protocol (DHCP)
- Domain Name System (DNS)
- File Input/Output (I/O)
- File Transfer Protocol (FTP)
- Hypertext Transfer Protocol (HTTP) and secure HTTP (HTTPS) Internet Message Access Protocol (IMAP)
- Lightweight Directory Access Protocol (LDAP)
- Messaging Application Program Interface (MAPI - Windows only)
- Network Information System (NIS/NIS+)
- Network News Transfer Protocol (NNTP)
- Packet internetnetwork groper (PING)
- Post Office Protocol (POP3)
- Round-Trip Email
- Simple Mail Transfer Protocol (SMTP)
- Simple Network Management Protocol (SNMP)
- SQL Query
- TCP Connect
- Trivial File Transfer Protocol (TFTP)

You can also test custom scripts with the Custom test and WinTask record-playback scripts with the Virtual User test.

Note: For more information, see the Service Availability Web Help. To access it, open the CA eHealth Advantage View console and click the question mark (?) icon on the top right of the screen, which launches the CA eHealth Web Help. From the CA eHealth Web Help, drill down to Systems, Application Insight Modules (AIMs), and click Service Availability in the left pane.

Monitoring Voice and Call Quality

CA eHealth for Cisco CallManager (CCM) and CA eHealth Voice Quality Monitor (VQM) are plug-ins to the CA eHealth SystemEDGE agent that you can use with CA eHealth and CA eHealth AdvantEDGE View to monitor your Cisco CallManager systems and applications, and voice quality and jitter on response paths.

Note: For more information, see the Web Help for CCM and VQM. To access it, open the CA eHealth Advantage View console and click the question mark (?) icon on the top right of the screen, which launches the CA eHealth Web Help. From the CA eHealth Web Help, drill down to Systems, Application Insight Modules (AIMs), and click Cisco CallManager (CCM) or Voice Quality Monitor (VQM) in the left pane.

Using the CA eHealth AIMs

The CA eHealth SystemEDGE agent provides a plug-in architecture through which it can load optional CA eHealth AIMs when it initializes. These CA eHealth AIMs (previously named AdvantEDGE Point modules) provide an extensible and flexible approach to supporting application-specific semantic knowledge. Following are the existing CA eHealths AIMs and the applications for which they provide management:

CA eHealth AIM for Microsoft Exchange

Manages and monitors Microsoft Exchange application.

CA eHealth AIM for Microsoft IIS

Manages and monitors Microsoft IIS application.

CA eHealth AIM for Microsoft SQL Server

Manages and monitors Microsoft SQL Server application.

CA eHealth AIM for Apache

Manages and monitors Apache Web Server.

CA eHealth AIM for Oracle

Manages and monitors Oracle database and application.

CA eHealth AIM for Check Point FireWall-1

Manages and monitors Check Point FireWall-1 application.

CA eHealth AIM for Network Services for UNIX

Manages and monitors vital network services for UNIX systems, including Sendmail, DNS, Lightweight Directory Access Protocol (LDAP), NFS, Network Information Services (NIS), and Dynamic Host Configuration Protocol (DHCP).

CA eHealth AIM for Network Services for Windows

Manages and monitors vital network services for Windows systems, including Active Directory, DHCP, DNS and Windows Internet Naming Service (WINS).

Note: For more information about these CA eHealth AIMs, see the user guide for the module in which you are interested.

Using CA eHealth with CA eHealth SystemEDGE

You can use the CA eHealth SystemEDGE agent to monitor your CA eHealth systems, and you can use the agent and eHealth together to manage and monitor other systems within your enterprise.

When you are using the CA eHealth SystemEDGE agent to monitor eHealth, you can run the `nhAddSysEdgeMonEntries` command to configure the agent to monitor critical eHealth processes and system logs. The command adds entries to the `sysedge.cf` file, and it stops and restarts the CA eHealth SystemEDGE agent to implement the changes.

Note: For more information about the `nhAddSysEdgeMonEntries` command, see the *CA eHealth Administration Reference*.

When you use CA eHealth SystemEDGE with eHealth, the CA eHealth poller can collect data from CA eHealth SystemEDGE agents and store that data in the CA eHealth database. The information is then available to the CA eHealth reporting and real-time monitoring tools. You can run At-a-Glance (AAG), Trend, Top N, What-If Capacity Trend, System Health, and MyHealth reports for systems to perform capacity planning, accurately document service problems, and troubleshoot potential problems.

Note: For more information, see the *CA eHealth System and Application Administration Guide*.

Using CA eHealth Live Health Application - Fault Manager with CA eHealth SystemEDGE

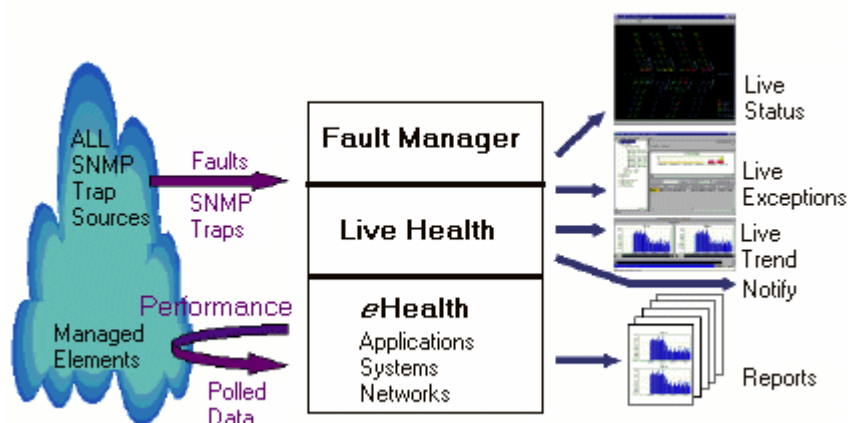
You can also use the CA eHealth SystemEDGE agent with CA eHealth Live Health Application - Fault Manager for real-time detection of potential problems. CA eHealth Live Health Application extends the features of CA eHealth to provide real-time performance and availability management for applications, systems, and networks. The CA eHealth SystemEDGE agent collects performance data, and CA eHealth Live Health Application analyzes the data with unique algorithms to identify outages and delays.

When you are using the CA eHealth SystemEDGE agent with CA eHealth Live Health Application, you can use the default Live Exceptions profiles for the CA eHealth SystemEDGE agent and the CA eHealth AIMS, or you can define your own profiles. The profiles organize alarm variables by delay, availability, unusual workload, and latency.

Fault Manager is an enhancement to Live Exceptions that enables CA eHealth to receive SNMP trap messages from devices and systems. Fault Manager interprets and processes trap information, reduces the noise of duplicate and repeated messages, and alerts you to the problems and conditions that interest you. When the CA eHealth system receives a trap, it processes the trap based on Live Exceptions rules and profiles that you configure. Thus, you can configure Fault Manager to raise an alarm for the associated element, or to ignore various trap messages.

Note: You can edit the `sysedge.cf` file to configure the CA eHealth SystemEDGE agent to feed specific traps to Fault Manager. For more information, see *Configuring SNMPv1 Traps* in the chapter "Configuring the CA eHealth SystemEDGE Agent" and *Configuring SNMPv2c/SNMPv3 Traps* in the appendix "SNMPv3 in CA eHealth SystemEDGE".

The following illustration shows how Fault Manager collects data from SNMP trap sources, such as the CA eHealth SystemEDGE agent, and sends it to a variety of displays:



Guidelines for Using the CA eHealth SystemEDGE Agent

You can gain the most value from the CA eHealth SystemEDGE agent by setting up effective management policies for your systems, networks, and applications. In particular, follow these guidelines:

- Use the CA eHealth SystemEDGE agent with CA eHealth AdvantEDGE View and CA eHealth. For more information, see the *CA eHealth AdvantEDGE View User Guide* and the *CA eHealth Administration Guide*.
- Automate management tasks through scheduling. If you are using CA eHealth SystemEDGE with CA eHealth, see the *CA eHealth Administration Guide* for more information.
- Limit SNMP access to the agent through access control lists and binding to private interfaces.
 - Use read-write communities for SNMP Sets and read-only communities for querying and polling.

Note: For more information about specifying community strings, see Configuring Access Communities in the chapter “Configuring the CA eHealth SystemEDGE Agent”.

- Use either port 161 (the SNMP industry standard port) or 1691 (which is reserved with the Internet Assigned Numbers Authority [IANA] for use with the CA eHealth SystemEDGE agent).

Note: For more information about configuring ports, see Configuring the SNMP Bind Address in the chapter “Configuring the CA eHealth SystemEDGE Agent”.

- Create groups in your management software based on the following:
 - Operating systems
 - Database systems
 - Clients
 - File servers
 - Email servers
 - Web servers
 - Physical location

Note: For more information about creating groups, see the *CA eHealth Administration Guide*.

- Define policies across groups of systems instead of on an individual system basis. Use management software such as CA eHealth AdvantEDGE View to push rules to groups of systems.

Note: For more information, see the *CA eHealth AdvantEDGE View User Guide*.

- Create a standard configuration for each group (through CA eHealth AdvantEDGE View Template Management or manually in the sysedge.cf file) based on system roles (for example, one configuration for email servers and another for Web servers; one for UNIX systems and another for Windows systems, and so on). Apply that configuration through CA eHealth AdvantEDGE View or by copying sysedge.cf to each system that you are monitoring.

Note: For more information about applying CA eHealth SystemEDGE Configuration templates, see the *CA eHealth AdvantEDGE View User Guide*. For more information about the sysedge.cf file, see the chapter, "Configuring the CA eHealth SystemEDGE Agent".

- Include meaningful information in the system location and contact field of your sysedge.cf file. For example, include information such as Rack 0, Slot 1, Atlanta and email: it@yourdomain.com.
- Use a standard table of row indexes across your self-monitoring tables. For example, use rows 10,000 to 10,999 across all self-monitoring tables for threshold monitoring, rows 11,000 to 11,999 for process monitoring, and so on. When defining these indexes and reserving rows, keep in mind the following:
 - Use large ranges of index numbers for each type of monitoring to enable for growth.
 - Use standard index entries for specific types of monitoring entries. For example, always use row 10,000 for monitoring the total amount of CPU available.
 - Use CA eHealth AdvantEDGE View Template Management to apply your self-monitoring entries to groups of systems.

Note: For more information about using standard index rows, see Assigning Entry Rows for the Monitor Table in the chapter “Configuring Threshold Monitoring”. For more information about CA eHealth AdvantEDGE View, see the *CA eHealth AdvantEDGE View User Guide*.

- Base any de-duplication on source index number or matched string within the trap--not on trap type alone.
- Keep the following points in mind as you create entries that you can use across multiple systems:
 - Monitor total CPU utilization states (which enable the configuration to be portable across single- and multi-processor systems).
 - Monitor thresholds and configure CA eHealth SystemEDGE to send a limited number of traps (for example, two or three) to prevent flooding of the NMS.
 - Enable Clear events to specify resetting of the event in the agent and the status in an NMS.

Note: For more information about effectively monitoring thresholds and clearing traps, see Monitor Table Flags in the chapter “Configuring Threshold Monitoring”.

- When you are managing a large network (hundreds or thousands of systems) and polling each system for granular data (at intervals of less than 15 minutes), do the following:
 - Use history collection to gain highly-granular data at the agent level, and let the management system poll the History table. You can use CA eHealth SystemEDGE for short-term history collection and CA eHealth (or another management system) for the long-term historical view.
 - For more information, see the chapter “Configuring the History Collection”.
 - Push the monitoring out to the agent, and configure the agent to send traps based on these self-monitoring entries.
- Limit the number of potential traps from a single monitoring entry by using the following CA eHealth SystemEDGE flags as you set up your self-monitoring entries:
 - 0x00000200: Send traps only after X occurrences of this event.
 - 0x00000400: Send up to X traps for this entry.

Note: For more information about traps, see Monitor Table Flags in the chapter “Configuring Threshold Monitoring”.

For more usage tips, see the following sections in this guide:

- Recommendations for Configuring Security in the chapter “Configuring the CA eHealth SystemEDGE Agent”

- Recommendations for Process and Service Monitoring in the chapter "Configuring Process and Service Monitoring"
- Recommendations for Log File Monitoring in the chapter "Configuring Log File Monitoring"
- Recommendations for Using Extensions in the chapter "Adding Custom MIB Objects"

Simple Network Management Protocol

SNMP is a standard for managing TCP/IP-based networks and devices. A typical network management environment contains many managed devices--each with an agent process--and at least one NMS, also referred to as the Manager or management system. The management system sends messages to the agent processes on the managed devices to request information or to modify parameters. The agent process carries out the management system's request and returns a reply. Additionally, the agent can send its own messages (traps) to the management system to notify it of important events. SNMP is the protocol that the agent and management system use to exchange this management information.

SNMP Message Types

The CA eHealth SystemEDGE agent uses standard SNMP messages to exchange management information with management systems. Following are the SNMP message types:

GetRequest

Obtains the value of a specific object-instance from the MIB.

GetNextRequest

Obtains the next object instance.

GetResponse

Returns requested information to an NMS.

SetRequest

Instructs an agent to change the value of an MIB's object parameters.

Trap

Notifies an NMS of exceptions.

SNMP Version 1 and SNMP Version 2c Communities

In SNMP, a community is a relationship between an agent and any number of management systems; it defines authentication and access-control permissions for communication between the management systems and the agent. An SNMP community is identified by a string of octets named the community name, which appears in the header portion of every SNMP message.

The agent checks the community name in the SNMP message header to determine if the message is authentic. If the community name matches one accepted by the agent, the message is considered to be authentic, and the agent processes the message. If it does not match, the agent records an authentication failure and drops the message. The community name also serves to determine what level of access (read-only or read-write) is available when the agent is using that community name.

Note: Although the community name is similar to a password, providing access to read--and even change--the values of an agent's MIB objects, the community name is *not* encrypted when it appears in an SNMP message header; it appears in clear text.

IP spoofing occurs when a system impersonates a trusted system to gain access to another system.

Although it is possible to attach IP-based access-control lists to individual communities, IP spoofing can circumvent the access control lists. Consequently, you should consider SNMP Version 1 (SNMPv1) communities insecure and take configuration steps to limit potential security violations. In addition, you should improve the overall security of the distributed system through router and system configuration.

Note: For more information, see the chapter, "Configuring the CA eHealth SystemEDGE Agent".

Access Communities for SNMPv1 and SNMPv2

Upon installation, the CA eHealth SystemEDGE agent is configured with only one community (default configuration), named public. This community provides read-only access to the agent's MIB object values. For security reasons, the default configuration does not define a read-write community. You can configure a read-write community string during installation or modify the string in the agent's configuration file, `sysedge.cf`. (The agent's configuration file contains a sample read-write community, but it is commented out.) Before you can modify (set) the agent's MIB values, you must define a community that provides read-write access.

Note: For information about configuring your own SNMP communities for the agent, see *Configuring Access Communities* in the chapter "Configuring the CA eHealth SystemEDGE Agent".

SNMP Version 3 User and Key Management

The CA eHealth SystemEDGE agent provides SNMPv3 user and key management through the `sysedgeV3.cf` configuration file, located in the `config` subdirectory of the agent's installation.

For information about how to enable SNMPv3 for the CA eHealth SystemEDGE agent, see the appendix "SNMPv3 in CA eHealth SystemEDGE".

SNMPv3 provides secure access to devices through a combination of authentication and encryption packets over the network, including the following security features:

- Message integrity - Helps ensure that the packet has not been altered during transmission
- Authentication - Verifies that the message is from a valid source
- Encryption - Scrambles package contents to prevent viewing from unauthorized sources

SNMPv3 uses both security models and security levels. A security model is an authentication strategy set up for a user and the group in which that user resides. A security level is the permitted level of security in a security model. A combination of a security model and a security level determines the security mechanism for handling SNMP packets. This combination of security models and security levels enables secure data collection from SNMP devices and encryption of confidential information to prevent exposure in network transmissions.

SNMP Traps

CA eHealth SystemEDGE can send SNMP traps. SNMP traps must be configured so that the agent can send them. SNMP trap configuration has trap communities for SNMPv1 and SNMPv2c traps, SNMPv3 configured users for SNMPv3 traps, and trap destinations.

Trap destinations indicate which management systems should receive the trap messages that the agent generates. The agent's self-monitoring features report exception conditions by sending trap messages to the management systems in the trap community. CA eHealth SystemEDGE can send SNMPv1, SNMPv2, or SNMPv3 traps.

Note: For information about configuring the CA eHealth SystemEDGE agent to send traps, see Configuring SNMPv1 Traps in the chapter "Configuring the CA eHealth SystemEDGE Agent" and Configuring SNMPv2c/SNMPv3 Traps in the appendix "SNMPv3 in CA eHealth SystemEDGE".

Note: For information about the types of traps that the CA eHealth SystemEDGE agent can send, see the chapter "Private Enterprise Traps."

Chapter 2: Installing the CA eHealth SystemEDGE Agent

This chapter explains how to install the CA eHealth SystemEDGE agent.

This section contains the following topics:

[Installing CA eHealth SystemEDGE on Windows Systems](#) (see page 41)

[Installing CA eHealth SystemEDGE on Solaris Systems](#) (see page 45)

[Installing CA eHealth SystemEDGE on HP-UX Systems](#) (see page 48)

[Installing CA eHealth SystemEDGE on Linux Systems](#) (see page 52)

[Installing CA eHealth SystemEDGE on AIX Systems](#) (see page 54)

[Installing CA eHealth SystemEDGE on Tru64 UNIX Systems](#) (see page 56)

[Reviewing the Configuration Files](#) (see page 57)

[Uninstalling the CA eHealth SystemEDGE Agent](#) (see page 59)

Installing CA eHealth SystemEDGE on Windows Systems

This section describes how to install the CA eHealth SystemEDGE agent on Windows systems.

Note: Before you begin installing the agent, verify that your system meets the system requirements in the *Release Notes*.

Note: Do *not* install CA eHealth SystemEDGE into a directory that includes spaces in the pathname (such as C:\Program Files). If you do, the agent will have difficulty locating and loading the CA eHealth AIMs.

Install the Software with InstallShield

The CA eHealth SystemEDGE agent for Windows is distributed as an InstallShield program named sysedge.exe.

Note: Throughout this guide, the term *Windows* encompasses supported versions of Windows. For a list of supported versions, see the *Release Notes*.

Note: You can also install CA eHealth SystemEDGE using a command line (non-graphical) installer. If you have previously installed the agent from the command line, you *can* upgrade using the InstallShield program. However, you cannot run the command line installation to upgrade an existing InstallShield installation. If you have installed CA eHealth SystemEDGE with InstallShield and you want to upgrade from the command line, you must uninstall the InstallShield version and then reinstall using the command line.

To install the software

1. Log on to the Windows system as an Administrator and make a local copy of the software package as CASysedge.exe from the distribution.

2. Go to the location where the software package is saved and double-click CASysedge.exe to run the InstallShield program.

A dialog appears, recommending that you turn off any anti-virus programs currently running.

3. After disabling your antivirus program, click OK.

The installation wizard appears.

Note: If you are upgrading the CA eHealth SystemEDGE agent, the CA eHealth SystemEDGE Maintenance dialog appears. Select Modify, and then click OK. After the program updates the agent, click Finish.

4. Click Next.

The Choose Destination Location dialog appears.

5. Click Next to accept the default installation directory, or click Browse, select the directory in which to install CA eHealth SystemEDGE, and then click Next.

Note: The recommended directory is C:\sysedge. You can install into a different directory. For example, you can install the agent in D:\CA\sysedge. If you select a different directory, make sure you install any CA eHealth SystemEDGE plug-ins in the same directory [for example, in D:\CA\sysedge\plugins].

The setup program copies the sysedge.cf and sysedge.mon configuration files into the system root directory, %SystemRoot%\system32. If you are upgrading from a previous version of the CA eHealth SystemEDGE agent, the installation updates all of the files in the installation directory. However, it will not update any configuration files.

An informational dialog appears, indicating that installation is almost complete, files are installed, and prompts you to click OK to continue with setup options.

6. Click OK.

A dialog appears, asking if you want to disable the Microsoft SNMP agent.

Note: If you do not have the Microsoft SNMP agent installed and enabled, skip to the next step about specifying the port on which you want the CA eHealth SystemEDGE agent to run.

- a. Click Yes to disable the Microsoft SNMP agent, or No to run both agents.

An informational dialog appears.

- b. Click OK.

The System Description installation wizard page appears.

- c. Specify a description, contact name, and system location in the Description, Contact, and Location fields, and then click Next.
- d. Click Yes to extract SNMP configuration information from the Microsoft SNMP agent, or No if you want to configure the CA eHealth SystemEDGE agent to use different community strings and trap destinations if the Microsoft SNMP agent is there.

The CA eHealth SystemEDGE SMNP Port installation wizard page appears.

- 7. Specify the port on which you want the CA eHealth SystemEDGE agent to run in the Port field, and then click Next.

Note: Port 161 is the default SNMP port for the CA eHealth SystemEDGE agent. If have the Microsoft SNMP agent installed and enabled, and if you have answered No to disable the Microsoft SNMP agent during the installation, CA eHealth SystemEDGE defaults to port 1691. Port 1691 is reserved for use with the CA eHealth SystemEDGE agent, but you can specify any port which is not in use on the system.

A dialog appears, asking if you want to activate CA eHealth SystemEDGE monitoring of top processes.

- 8. Click Yes to enable the Top Processes AIM.

A dialog appears, asking if you want to administer CA eHealth SystemEDGE.

- 9. Click Yes if you want administer CA eHealth SystemEDGE.

The CA eHealth SystemEDGE Control Panel appears.

To edit the sysedge.cf file, use the following steps:

- a. Click sysedge.cf. The file opens in a text editor.
- b. Add community strings or trap destinations.
- c. Save and close the file.

- 10. When you are finished modifying the CA eHealth SystemEDGE configuration, click Close in the CA eHealth SystemEDGE Control Panel.

A question dialog appears, asking permission to stop the Microsoft SMNP service and subagents, if the Microsoft SNMP service is installed and enabled.

Note: This dialog only appears if you have disabled the Microsoft SNMP service in the previous steps.

- 11. Click Yes.

Note: You must click Yes to continue and complete the installation. If you click No, the installation will be aborted.

The InstallShield Wizard Complete wizard page appears.

- 12. Click Finish.

Note: The *only* case in which the system reboots after you install the agent is if you are running a Windows 2000 system that does not include the Windows Installer service. Use the Windows `msiexec` command to determine which version of the Windows Installer is running on your system.

Note: CA eHealth SystemEDGE installation automatically installs "Microsoft Visual C++ 2005 Redistributable Package" if you are installing on Windows 64 bit operating systems (e.g. AMD64, Intel EM64T, or Itanium64). CA eHealth SystemEDGE agent will not function without this package installed.

Install the Software from the Command Line

You can install the CA eHealth SystemEDGE Windows package using a command line (non-graphical) version of the installer.

Note: You can also install CA eHealth SystemEDGE with InstallShield. If you have previously installed the agent from the command line, you *can* upgrade using the InstallShield program. However, you cannot run the command line installation to upgrade an existing InstallShield installation. If you have installed CA eHealth SystemEDGE with InstallShield and you want to upgrade from the command line, you must uninstall the InstallShield version and then reinstall using the procedure in this section.

To install CA eHealth SystemEDGE using the command line installer

1. Log on to the Windows system as an Administrator and save the distribution as `CASysedge.exe` at the top level of the `%SystemDrive%` path (for example, `C:\CASysedge.exe`).
2. Do *one* of the following as applicable:
 - For SystemEDGE 4.1p14 or earlier, stop the Microsoft master agent.
 - For SystemEDGE 4.2p11 or higher, stop the SystemEDGE Service.

Note: Failure to stop the service may cause the CA eHealth Top Processes AIM to not update.

A dialog appears, recommending that you turn off any anti-virus programs currently running.

3. Run the following at the command line:

```
sysedge.exe c:\
```

The distribution installs in the directory specified (for example, `C:\sysedge.`).

The CA eHealth SystemEDGE installation automatically installs "Microsoft Visual C++ 2005 Redistributable Package" if you are installing on Windows 64 bit operating systems (e.g. AMD64, Intel EM64T, or Itanium64). To install "Microsoft Visual C++ 2005 Redistributable Package" you must accept the license agreement that is displayed. The license agreement for "Microsoft Visual C++ 2005 Redistributable Package" is suppressed in the InstallShield version of the CA eHealth SystemEDGE installer.

Note: If you do not accept the license agreement, "Microsoft Visual C++ 2005 Redistributable Package" will not be installed. CA eHealth SystemEDGE agent will not function without this package installed.

4. Click Y to disable the Microsoft SNMP service, or N to run both the Microsoft SNMP service and the CA eHealth SystemEDGE agent.

If you disabled the Microsoft SNMP service, SystemEDGE automatically uses port 161. If you did not disable the Microsoft SNMP service, SystemEDGE automatically uses port 1691.

A message displays indicating the that installation is complete, and the distribution installs in the specified directory (for example, C:\sysedge).

Installing CA eHealth SystemEDGE on Solaris Systems

This section describes how to install the CA eHealth SystemEDGE agent on all Solaris systems, which include Solaris SPARC, Intel, and AMD64 versions.

These instructions explain how to install CA eHealth SystemEDGE in the /opt/EMPSysedge directory. If you want to install CA eHealth SystemEDGE in a different directory, see Install CA eHealth SystemEDGE in a Non-Default Directory for Solaris in this chapter.

Note: Before you begin installing the agent, verify that your system meets the system requirements in the *Release Notes*.

Install the Software on Solaris

The CA eHealth SystemEDGE agent for Solaris 2.x is distributed as a software package. This distribution uses the standard pkgadd utility to install the agent.

To install the software

1. Log in as root.

Note: You must stop any old versions of CA eHealth SystemEDGE before upgrading or installing CA eHealth SystemEDGE.

2. Change to the directory where you saved sysedge.pkg if you downloaded the software package from the web, or change to the mounted directory if the CA eHealth SystemEDGE software package is in a CD/DVD.

3. Install the package using the following command:

```
pkgadd -d ./sysedge.pkg
```

The installation starts and the resulting package will be in /opt/EMPSysedge.

Note: For each prompt, the default value is shown in brackets ([]). You can press Return to accept the default, or you can enter another value and then press Return.

4. Disable the native SNMP agent, if applicable, by pressing Return or y at the following 'Disable the native SNMP Agent if applicable (yes) [y,n,?]' prompt.

Note: For more information about running multiple SNMP agents simultaneously, see the chapter "Using the CA eHealth SystemEDGE Agent with Other SNMP Agents".

5. Configure the CA eHealth SystemEDGE agent to use UDP port 1691 if you are running another SNMP agent on port 161 by entering y at the 'Change SystemEDGE port to 1691 (no, default is 161) [y,n,?]' prompt.

Note: You may need to perform this step if you did not disable the native SNMP agent in the previous step, assuming the native agent uses port 161.

6. Press Return at the 'Configure system description (yes) [y,n,?]' prompt if you want to enter a description for your system.

Enter the system description, for example, **Test System 1**, at the 'Enter system description [?]' prompt.

7. Press Return at the 'Configure system location (yes) [y,n,?]' prompt if you want to enter a location for your system.

Enter the system location, for example, **QA Lab**, at the 'Enter system location (followed by newline)' prompt.

8. Press Return at the 'Configure system contact (yes) [y,n,?]' prompt if you want to enter the name of a contact for this system.

Enter the contact name, for example, **Test system contact**, and then press Return at the 'Enter system contact (followed by newline):' prompt.

9. Press Return at the 'Enable Top Processes AIM (yes) [y,n,?]' prompt to configure CA eHealth SystemEDGE to load the Top Processes AIM.

The installation script displays informational messages.

10. Press Return at the 'Configure a read-only community (yes) [y,n,?]' prompt to configure a read-only community.

Enter the name of the community that you want to configure as read-only at the 'Enter read-only community (no spaces, case-sensitive) (public):' prompt.

The script displays 'Setting read-only community to public.'

11. Press Return at the 'Configure a read-write community? (yes)' prompt to configure a read-write community.

Enter the name of the community that you want to configure as read-write (for example, private), at the 'Enter read-write community (no spaces, case-sensitive):' prompt.

The script displays 'Setting read-write community to private.'

12. Press Return at the 'Configure a SNMPv1 Trap Destination (yes) [y,n,?]' prompt to configure a trap destination.

- a. Enter a SNMPv1 Trap destination at the prompt 'Enter SNMPv1 Trap destination IP address [?]'. The trap destination can be a valid host-name, IPv4 address, or IPv6 address. For example, aview.ca.com, ea2f:fe90:abcd:0000:230:a2f:200:ad01, or 130.10.100.101.

- b. Enter the trap community, or press Return to configure public as a trap community at the 'Enter SNMPv1 Trap community (public) [?]:' prompt.

The script displays 'Adding trap community to config file Restarting SystemEDGE' and then indicates that installation is complete.

13. Continue with the installation by entering y at the prompt 'Do you want to continue with the installation of <EMPsysedg> [y,n,?]'.

If the installation is successful, a message 'Installation of <EMPsysedg> was successful' is displayed at the end of the installation.

Install CA eHealth SystemEDGE in a Non-Default Directory for Solaris

By default, the CA eHealth SystemEDGE agent for Solaris is installed in the /opt/EMPsysedge directory. You can install CA eHealth SystemEDGE in a different directory under the /opt directory. For example, you can install CA eHealth SystemEDGE in /opt/CA/EMPsysedge.

To install CA eHealth SystemEDGE in a different directory for Solaris

1. Create an administration text file that specifies the new installation directory, as follows:

```
basedir=/opt/CA
mail=
instance=unique
partial=ask
runlevel=ask
idepend=ask
rdepend=ask
space=ask
setuid=ask
conflict=ask
action=ask
```

Enter the following from a command prompt to instruct the pkgadd utility to use the text file you created (*myadmin.file* in this example) to install the agent to the directory you specified:

```
pkgadd -a myadmin.file -d ./sysedge.pkg
```

This example installs the agent in /opt/CA/EMPsysedge.

2. Follow the installation prompts.

Note: If you are installing any CA eHealth AIMs or other CA eHealth SystemEDGE modules, install them in the plugins subdirectory of the directory you specified (for example, /opt/CA/EMPsysedge/plugins).

Installing CA eHealth SystemEDGE on HP-UX Systems

This section describes how to install the CA eHealth SystemEDGE agent on HP-UX systems. For a list of supported systems and versions, see the *Release Notes*.

Note: Before you begin installing the agent, verify that your system meets the system requirements in the *Release Notes*.

Install the Software on HP-UX

The CA eHealth SystemEDGE agent software distribution for HP-UX is formatted as an HP software depot package. This distribution uses the `swinstall` utility to install the agent.

Note: Hewlett Packard defines a software depot as a group of related file sets.

To install the software on HP-UX

1. Log in as root.
2. Stop CA eHealth SystemEDGE if it running.
3. Download or copy the CA eHealth SystemEDGE HP-UX installation software package to `/tmp/CASysedge.depot`.
4. Run the HP-UX `swinstall` utility to install CA eHealth SystemEDGE by entering *one* of the following as appropriate:

- New installations: For new installations of CA eHealth SystemEDGE (clean installations), enter the following:

```
host% swinstall -s /tmp/sysedge.depot EMPsysedge
```

Note: Ensure that the depot file name (`-s` option) is specified with an absolute file path.

- Reinstall or upgrade installations: For existing installations of CA eHealth SystemEDGE (reinstalls and upgrades), enter the following:

```
host% swinstall -s /tmp/sysedge.depot -x reinstall=true EMPsysedge
```

Note: The above command can be used for a clean installation as well.

By default, the `swinstall` utility installs the software in the `/opt/EMPsysedge` directory.

5. Run the `swjob` command shown at the end of the `swinstall` installation. This command gives more information about the installation and indicates any errors. For example:

```
swjob -a log machine1-0133 @ machine1:/
```

6. Run the installation script from the `/opt/EMPsysedge/` directory by entering the following commands:

```
cd /opt/EMPsysedge
./Install
```

When you run the installation script, you must enter valid values for all prompts. If you enter invalid values or press Enter at a prompt that requires a value (and does not offer a default value), the script will not complete properly. If the script generates an error message or is unable to complete, you must run the script again and specify valid values for all prompts.

The installation script displays informational messages.

You can change the configuration information after the installation; for more information, see the chapter, "Configuring the CA eHealth SystemEDGE Agent."

Note: For each prompt, the default value is shown in brackets ([]). You can press Return to accept the default, or you can enter another value and then press Return.

7. Disable the native SNMP agent (if you want to disable it) by entering **yes** at the 'Disable native SNMP agent (if applicable)? [no]' prompt.

Note: For more information about running multiple SNMP agents simultaneously, see the chapter, "Using the CA eHealth SystemEDGE Agent with Other SNMP Agents".

The script displays the following:

```
Disabling native SNMP agent
snmpdm stopped
```

8. Configure CA eHealth SystemEDGE to use UDP port 1691 if you are running another SNMP agent on port 161 by pressing Return at the 'Change SystemEDGE port to 1691 (default is 161)? [yes]' prompt.

You must perform this step if you did not disable the native SNMP agent in the previous step.

9. Press Return at the 'Configure system description? [yes]' prompt if you want to enter a description for your system.

Enter the system description, for example, Test System, and then press Return at the 'Enter system description (followed by newline):' prompt.

10. Press Return at the 'Configure system location? [yes]' prompt if you want to enter a location for your system.

Enter the system location, for example, Test system location, and then press Return at the 'Enter system location (followed by newline):' prompt.

11. Press Return at the 'Configure system contact? [yes]' prompt if you want to enter the name of a contact for this system:

Enter the contact name, for example, Test system contact, and then press Return at the 'Enter system contact (followed by newline):' prompt.

12. Press Return at the 'Enable Top Processes AIM? [yes]' prompt to configure CA eHealth SystemEDGE to load the Top Processes AIM.

The installation script displays the following:

```
Enabling topprocs-hpux64bit.so AIM
Configuring community strings:
You should configure a read-only and a read-write community.
You need a read-only community to discover SystemEDGE.
```

13. Press Return at the 'Configure a read-only community-string? [yes]' prompt to configure a read-only community.

Enter the name of the community that you want to configure as read-only at the 'Enter read-only community (no spaces, case-sensitive) [public]:' prompt.

The script displays, 'Setting read-only community to public.'

14. Press Return at the 'Configure a read-write community-string? [yes]' prompt to configure a read-write community.

Enter the name of the community that you want to configure as read-write (for example, private) at the 'Enter read-write community (no spaces, case-sensitive):' prompt.

The script displays, 'Setting read-write community to private.'

15. Press Return at the 'Configure a new SNMPv1 Trap Destination? [yes]' prompt to configure a trap destination.

- a. Enter a SNMPv1 Trap destination at the prompt 'Enter a SNMPv1 Trap destination [no spaces, case-sensitive] (none):'. The trap destination can be a valid host-name, IPv4 address, or IPv6 address. For example, aview.ca.com, ea2f:fe90:abcd:0000:230:a2f:200:ad01, or 130.10.100.101.

- b. Enter the trap community, or press Return to configure public as a trap community at the 'Enter SNMPv1 Trap community [public]:' prompt.

The script displays, 'Setting a SNMPv1 Trap destination to aview.ca.com with community public'.

The installation stops and restarts SystemEDGE.

Once the installation is successful, 'CA eHealth SystemEDGE Agent Installation complete' displays.

Install CA eHealth SystemEDGE in a Non-Default Directory for HP-UX

By default, the CA eHealth SystemEDGE agent for HP-UX is installed in the /opt/EMPSysedge directory. You can install CA eHealth SystemEDGE in a different directory other than /opt/EMPSysedge.

To install CA eHealth SystemEDGE in a different directory for HP-UX

1. Log in as root.
2. Stop CA eHealth SystemEDGE if it is running.
3. Download or copy the CA eHealth SystemEDGE HP-UX installation software package to /tmp/CASysedge.depot.
4. Run the HP-UX swinstall utility to install CA eHealth SystemEDGE by entering one of the following as appropriate:

- New installation: For new installations of CA eHealth SystemEDGE (clean installations), enter the following:

```
host% swinstall -s /tmp/CASysedge.depot EMPSysedge,l=<new-directory>
```

Note: Ensure that the depot file name (-s option) is specified with an absolute file path

- Reinstall or upgrade installation: For existing installations of CA eHealth SystemEDGE (reinstalls and upgrades), enter the following:

```
host% swinstall -s /tmp/CASysedge.depot -x reinstall=true  
EMPSysedge,l=<new-directory>
```

Note: You can use this command for a clean installation as well.

The above swinstall commands install the software in the <new-directory> that you specified.

5. Follow the rest of the steps described in Install the Software on HP-UX.

Installing CA eHealth SystemEDGE on Linux Systems

This section describes how to install the CA eHealth SystemEDGE agent on Linux systems. For a list of supported systems and versions, see the *Release Notes*.

Note: Before you begin installing the agent, verify that your system meets the system requirements in the *Release Notes*.

Install the Software on Linux

The CA eHealth SystemEDGE agent for Linux is distributed as a tar file.

To install the software on Linux

1. Log in as root.
2. Stop CA eHealth SystemEDGE if it is running.
3. Create the home directory for the CA eHealth SystemEDGE agent by entering the following:

```
mkdir /opt/EMPSysedge
```

The recommended default installation directory is /opt/EMPSysedge. You can use another directory, but the examples throughout this guide assume that you are installing the CA eHealth SystemEDGE agent in the /opt/EMPSysedge directory.

4. Download or copy the CA eHealth SystemEDGE Linux installation tar package to /opt/EMPSysedge/CASysedge.tar.
5. Change the directory to the agent's home directory, and enter the following commands, one at a time:

```
cd /opt/EMPSysedge
tar xvf CASysedge.tar
```

6. Execute the installation script from the directory where you have extracted the CA eHealth SystemEDGE installation package. For example, enter the following if you extracted the installation files to the /opt/EMPSysedge directory:

```
cd /opt/EMPSysedge
./Install
```

When you run the installation script, you must enter valid values for all prompts. If you enter invalid values or press Enter at a prompt that requires a value (and does not offer a default value), the script will not complete properly. If the script generates an error message or is unable to complete, you must run the script again and specify valid values for all prompts.

You can change the configuration information after the installation if required; for more information, see the chapter, "Configuring the CA eHealth SystemEDGE Agent."

Note: For a detailed example of the installation script, see Installing CA eHealth SystemEDGE on HP-UX Systems. For each prompt, the default value is shown in parentheses (). You can press Return to accept the default, or you can enter another value and then press Return.

Install CA eHealth SystemEDGE in a Non-Default Directory for Linux

By default, the CA eHealth SystemEDGE agent for Linux is installed in the /opt/EMPSysedge directory. You can install CA eHealth SystemEDGE in a different directory other than /opt/EMPSysedge.

To install CA eHealth SystemEDGE in a different directory for Linux, create a directory where you want to install CA eHealth SystemEDGE, and extract the Linux CA eHealth SystemEDGE distribution into this directory:

```
mkdir /usr/joedoe  
  
cd /usr/joedoe  
  
tar xvf CASysedge.tar
```

For detailed information, see [Install the Software on Linux](#) and extract the tar file to the directory you want to install CA eHealth SystemEDGE in.

Installing CA eHealth SystemEDGE on AIX Systems

This section describes how to install the CA eHealth SystemEDGE agent on AIX systems. For a list of supported systems and versions, see the *Release Notes*.

Note: Before you begin installing the agent, verify that your system meets the system requirements in the *Release Notes*.

Install the Software on AIX

The CA eHealth SystemEDGE agent software distribution for AIX is formatted as an AIX product. This distribution uses the smit utility to install the CA eHealth SystemEDGE agent product.

Important! By default, the CA eHealth SystemEDGE agent for AIX is installed in the /usr/lpp/EMPSysedge directory. You *cannot* install CA eHealth SystemEDGE in any directory other than /usr/lpp/EMPSysedge for AIX.

To install the software on AIX

1. Log in as root.
2. Stop CA eHealth SystemEDGE if it is running.
3. Download or copy the CA eHealth SystemEDGE AIX installation software package to /tmp/CASysedge.bff:

```
cp CASysedge.bff /tmp
```

4. Change the directory to /tmp/ and remove /tmp/.toc:

```
cd /tmp  
rm -f .toc
```

5. Run the smit utility by entering the following:

```
smit install_latest
```

- a. Specify the current directory (".") as the input device.
- b. Select EMPsysedge.rte as the software to install.
- c. If you want to reinstall or upgrade the CA eHealth SystemEDGE agent, change the option 'AUTOMATICALLY install requisite software?' to 'no' and the option 'OVERWRITE same or newer versions' to 'yes'.
- d. Select OK (if using graphical version of smit) or press Return (if using text based version of smit) to install the software.

The CA eHealth SystemEDGE agent will be installed in the default directory /usr/lpp/EMPsysedge.

6. Execute the installation script by entering the following commands:

```
cd usr/lpp/EMPsysedge  
./Install
```

When you run the installation script, you must enter valid values for all prompts. If you enter invalid values or press Enter at a prompt that requires a value (and does not offer a default value), the script will not complete properly. If the script generates an error message or is unable to complete, you must run the script again and specify valid values for all prompts.

The installation script displays a message that post-installation is starting.

This script enables you to configure the CA eHealth SystemEDGE agent. You can change the configuration information after the installation if required; for more information, see the chapter, "Configuring the CA eHealth SystemEDGE Agent."

Note: For a detailed example of the installation script, see Installing SystemEDGE on HP-UX Systems. For each prompt, the default value is shown in parentheses (). You can press Return to accept the default, or you can enter another value and then press Return.

Installing CA eHealth SystemEDGE on Tru64 UNIX Systems

This section describes how to install the CA eHealth SystemEDGE agent on Tru64 UNIX systems. For a list of supported systems and versions, see the *Release Notes*.

Note: Before you begin installing the agent, verify that your system meets the system requirements in the *Release Notes*.

Install the Software on Tru64 UNIX

The CA eHealth SystemEDGE agent software distribution for Tru64 UNIX is formatted as a product kit. Tru64 UNIX defines a product kit as a group of related file subsets. This distribution uses the `setld` utility to install the CA eHealth SystemEDGE agent package.

Important! By default, the CA eHealth SystemEDGE agent for Tru64 UNIX is installed in the `/usr/opt/EMPsysedge` directory. You *cannot* install CA eHealth SystemEDGE in any directory other than `/usr/opt/EMPsysedge` for Tru64 UNIX.

To install the software on Tru64 UNIX

1. Log in as root.
2. Stop CA eHealth SystemEDGE if it is running.
3. Download or copy the CA eHealth SystemEDGE installation software package to `/tmp/CASysedge.tar`.
4. Run the `setld` utility by entering the following commands, one at a time:

```
cd /tmp
tar xvof CASysedge.tar
setld -l /tmp EMPSYSEGE
```

The CA eHealth SystemEDGE agent will be installed in the default directory `/usr/opt/EMPsysedge`.

5. Execute the installation script by entering the following commands:

```
cd /usr/opt/EMPsysedge
./Install
```

When you run the installation script, you must enter valid values for all prompts. If you enter invalid values or press Enter at a prompt that requires a value (and does not offer a default value), the script will not complete properly. If the script generates an error message or is unable to complete, you must run the script again and specify valid values for all prompts.

The installation script displays a message that post-installation is starting.

This script enables you to configure the CA eHealth SystemEDGE agent. You can change the configuration information after the installation if necessary; for more information, see the chapter, "Configuring the CA eHealth SystemEDGE Agent."

Note: For a detailed example of the installation script, see Installing CA eHealth SystemEDGE on HP-UX Systems. For each prompt, the default value is shown in parentheses (). You can press Return to accept the default, or you can enter another value and then press Return.

Reviewing the Configuration Files

When the CA eHealth SystemEDGE agent first starts, it reads the following configuration text files to determine configuration settings:

- sysedge.cf
- sysedgeV3.cf
- sysedge.mon

The CA eHealth SystemEDGE agent installation automatically installs sysedge.cf and sysedge.mon in the /etc (UNIX) or %SystemRoot%\system32 (Windows) directories during the installation process, unless you are upgrading from a previous version of the CA eHealth SystemEDGE agent. If you are upgrading, CA eHealth SystemEDGE does *not* overwrite your existing files, but copies the new files to the config subdirectories to enable you to compare the new files with the existing files.

The following chapters contain more information about the sysedge.cf configuration file:

- "Configuring the CA eHealth SystemEDGE Agent"
- "Starting the CA eHealth SystemEDGE Agent"
- "Configuring Threshold Monitoring"
- "Configuring Process and Service Monitoring"
- "Configuring Process Group Monitoring"
- "Configuring Log File Monitoring"
- "Configuring Windows Event Monitoring"
- "Configuring History Collection"
- "Adding Custom MIB Objects"
- "Adding Windows Registry and Performance MIB Objects"

Note: CA recommends that you use the sysedge.cf file (rather than the sysedge.mon file) for manually adding entries to the monitoring tables and configuring the agent. The sysedge.mon file is a backing store for the agent's self-monitoring tables. The two files interact, and entries in sysedge.cf take precedence over entries in sysedge.mon. The sysedge.cf file is static; if edited remotely, you must restart the agent for the changes to take effect. The sysedge.mon file is not static, so CA eHealth AdvantEDGE View and other management software can update this file through SNMP Sets.

The following appendix contains more information about the SNMPv3 configuration file sysedgeV3.cf configuration file:

- "SNMPv3 in CA eHealth SystemEDGE"

The following chapters and appendix contain more information about the sysedge.mon configuration file:

- Chapter, "Configuring Threshold Monitoring"
- Chapter, "Configuring Process and Service Monitoring"
- Chapter, "Configuring Process Group Monitoring"
- Chapter, "Configuring Log File Monitoring"
- Chapter, "Configuring Windows Event Monitoring"
- Chapter, "Configuring History Collection"
- Appendix, "Adding Self-Monitoring Entries to the sysedge.mon File"

Configuration Files for UNIX Systems

By default, the CA eHealth SystemEDGE agent looks for the sysedge.cf and sysedge.mon configuration files in the /etc directory and the sysedgeV3.cf file in the config subdirectory of the agent's installation. The CA eHealth SystemEDGE agent installation script installs these files into these directories automatically during the installation, unless the directories already include files with those names (that is, if you are performing an upgrade instead of a new installation). Edit these files to match your local requirements.

Note: If you have the CA eHealth SystemEDGE agent already installed, during an upgrade, review the new files in the config subdirectory of the agent's installation to identify the latest features, and then integrate them with your current configuration files.

You can also instruct the CA eHealth SystemEDGE agent to look for these files in another directory by specifying an alternate directory through command-line arguments. For more information, see the chapter "Starting the CA eHealth SystemEDGE Agent."

Configuration Files for Windows Systems

By default, the CA eHealth SystemEDGE agent looks for the sysedge.cf and sysedge.mon configuration files in the %SystemRoot%\system32\ directory and the sysedgeV3.cf file in the config subdirectory of the agent's installation. The CA eHealth SystemEDGE agent setup program installs these files into these directories automatically during the installation. Edit these files to match your local requirements.

Note: If you have the CA eHealth SystemEDGE agent already installed, during an upgrade, review the new files in C:\sysedge\config to identify the latest features, and then integrate them with your current configuration files.

Uninstalling the CA eHealth SystemEDGE Agent

This section explains how to remove the files and subdirectories associated with the CA eHealth SystemEDGE agent.

Note: The steps also remove the agent's configuration file (sysedge.cf) and Monitor table configuration file (sysedge.mon) from the /etc (UNIX) or %SystemRoot%\system32\ (Windows) directory. To save these files, copy them to another directory before you run the Remove utility.

Uninstall CA eHealth SystemEDGE for Windows Systems

The uninstall program removes CA eHealth SystemEDGE. It also removes the sysedge.dll, sysedge.cf, and sysedge.mon files from the %SystemRoot%\system32\ directory and the sysedgeV3.cf file from the config subdirectory of the agent's installation.

Note: If you want to save these files, copy them to another directory before you run the uninstall utility.

If you have installed the agent with the InstallShield installer, see Uninstall the Agent with InstallShield. If you installed the agent from the command line, see Uninstall the Agent from the Command Line.

Uninstall the Agent with InstallShield

If you used the Windows InstallShield installer to install the CA eHealth SystemEDGE agent, you can also use it to remove the agent.

To uninstall the agent with InstallShield

1. Log on to the Windows system as Administrator.
2. Select Start, Settings, Control Panel, Add or Remove Programs.
The Add or Remove Programs dialog appears.
3. Click CA eHealth SystemEDGE, click Change/Remove, select Remove, and click Next.
The Confirm Uninstall prompt appears.
4. Click OK.
A prompt appears asking whether to remove the configuration files.
5. Click Yes to continue with the uninstallation, or click No to stop the uninstallation.
A prompt appears asking if you want to save a copy of the log file.
6. Click Yes to save a copy of the log file, or No if you do not want to save the log file.
The InstallShield Wizard Complete wizard page appears.
7. Click Finish.

Uninstall the Agent from the Command Line

If you installed the agent using the command line (non-graphical) installer, you can remove it by running the setup utility with the -x argument. The -x argument instructs the setup utility to remove all CA eHealth SystemEDGE files including the configuration files from the %SystemRoot%\system32 directory.

Note: If you want to save the configuration files, copy them to another directory before you run the CA eHealth SystemEDGE agent setup utility.

To remove the installation of the CA eHealth SystemEDGE agent for Windows

1. Log on to the Windows system as Administrator.
2. Change to the CA eHealth SystemEDGE installation directory by entering the following, where C:\sysedge is the directory where you installed the agent:

C:\sysedge

3. Enter the following to remove the software:

```
setup -x
```

Uninstall CA eHealth SystemEDGE for UNIX Systems

Starting with CA eHealth SystemEDGE r4.3.0, you can use the Remove script from the agent's installation directory to uninstall the agent on UNIX systems.

To uninstall CA eHealth SystemEDGE for UNIX Systems

1. Switch to the CA eHealth SystemEDGE installation directory and enter the following command:

```
./Remove
```

The CA eHealth Advanced Encryption package is removed if it is installed.

2. Select y to continue with the uninstallation.

You are asked whether you want to remove the configuration files.

3. Enter y to remove the configuration files.

When the uninstallation finishes, the 'CA eHealth SystemEDGE Agent has been uninstalled' message displays.

Chapter 3: Configuring the CA eHealth SystemEDGE Agent

This chapter describes how to set configuration parameters that the CA eHealth SystemEDGE agent reads on startup. These configuration parameters are defined in the configuration files `sysedge.cf` and `sysedgeV3.cf`.

This section contains the following topics:

[Configuration Files](#) (see page 63)

[Interactions Between `sysedge.cf` and `sysedge.mon`](#) (see page 64)

[Configuring the Agent During the Installation Procedure](#) (see page 64)

[Before You Begin](#) (see page 65)

[Configuration Using `sysedge.cf`](#) (see page 65)

[Using the SystemEDGE Control Panel for Windows](#) (see page 83)

Configuration Files

You define configuration parameters for CA eHealth SystemEDGE using the `sysedge.cf` and `sysedgeV3.cf` configuration files.

The `sysedge.cf` configuration file, a plain text file viewable through a text editor, specifies local system values such as the following:

- System description, community strings, and trap communities
- Agent behavior for reporting or not reporting security-related information and running or not running action scripts
- Entries for the agent's self-monitoring tables

The `sysedge.cf` file is located in the `/etc` (UNIX) or `%SystemRoot%\system32` (Windows) directory. On Windows systems, you can also access `sysedge.cf` through the CA eHealth SystemEDGE Control Panel by selecting Start, Settings, Control Panel, eHealth SystemEDGE, and then clicking `sysedge.cf` on the CA eHealth SystemEDGE Control Panel.

Note: When you modify the `sysedge.cf` file, you must stop and restart the CA eHealth SystemEDGE agent for your changes to take effect.

The sysedgeV3.cf configuration file is typically an encrypted (if encrypted by the se_enc utility), unviewable file. It contains SNMPv3 information such as the following:

- SNMPv3 user configuration
- SNMPv2c and SNMPv3 trap destination configuration

The sysedgeV3.cf file is located in <SystemEDGE-Installation-Directory>/config/ for UNIX and Windows. For more information about configuring SNMPv3 for the CA eHealth SystemEDGE agent, see the appendix "SNMPv3 in CA eHealth SystemEDGE".

Note: When you modify the sysedgeV3.cf file, you must stop and restart the CA eHealth SystemEDGE agent for your changes to take effect.

Interactions Between sysedge.cf and sysedge.mon

The CA eHealth SystemEDGE agent uses the sysedge.mon file as a backing store for the agent's self-monitoring tables. The sysedge.cf and sysedge.mon files interact, and entries in sysedge.cf take precedence over entries in sysedge.mon. The sysedge.cf file is static and cannot be edited remotely. The sysedge.mon file is not static, so CA eHealth AdvantEDGE View and other management software can update this file through SNMP Sets.

If you remove a monitoring entry from sysedge.cf and that entry also exists in sysedge.mon, you must also remove it from sysedge.mon to prevent the agent from using it. When you add configuration entries to the agent through CA eHealth AdvantEDGE View, the entries are stored in sysedge.mon and are *not* added to sysedge.cf. For more information about using sysedge.mon, see the appendix "Adding Self-Monitoring Entries to the sysedge.mon File."

Configuring the Agent During the Installation Procedure

The CA eHealth SystemEDGE installation lets you perform some configuration tasks when you install the agent. For example, you can configure system description and location, read-only and read-write communities, and SNMPv1 trap destinations during the installation process. You can later modify any of those settings by editing the sysedge.cf file, as described in this chapter.

Note: SNMPv3 users are not configured during installation. You must manually edit the sysedgeV3.cf configuration file. For more information about configuring SNMPv3 for the CA eHealth SystemEDGE agent, see the appendix "SNMPv3 in CA eHealth SystemEDGE".

Before You Begin

Make sure that your installation has copied the `sysedge.cf` file to the `/etc` (UNIX) or `%SystemRoot%\system32` (Windows) directory. Keep in mind that if you performed an upgrade instead of a clean installation, the installation does not overwrite your existing `sysedge.cf`, `sysedge.lic`, or `sysedge.mon` files. You can compare the existing versions with the new versions, which are installed in the `config` subdirectories, modify the new versions, save any information that you want to keep from the existing files, and then copy the new files into the `/etc` (for UNIX) and `%SystemRoot%\system32` (for Windows) directories.

To copy the file manually

For UNIX systems, enter the following:

```
cp <SystemEDGE-Installation-Directory>/config/sysedge.cf /etc/
```

For Windows systems, enter the following:

```
copy <SystemEDGE-Installation-Directory>\config\sysedge.cf %SystemRoot%\system32\
```

Configuration Using `sysedge.cf`

The following sections describe how to configure the CA eHealth SystemEDGE Agent after installation using the `sysedge.cf` configuration file.

Configuring System Information

The CA eHealth SystemEDGE installation lets you define system location and contact. If you want to modify those values, you can do so manually in the `sysedge.cf` file.

You can update the `syscontact` and `syslocation` fields as follows:

- Replace *System contact unknown* with the name of the person who is the contact for this system.
- Replace *System location unknown* with a short description of the system's physical location. For example, specify QA Lab.

Configuring Access Communities

The CA eHealth SystemEDGE installation lets you define read-only and read-write communities. You can modify those communities or define additional communities manually in the sysedge.cf file. The configuration file defines access communities using the following format:

```
community community-name permissions access-list
```

community-name

Specifies any octet string.

permissions

Specifies what level of permissions to grant, either read-only or read-write.

access-list

Specifies a space-separated list of IP addresses (in dotted decimal notation) that defines the systems that have access using the given community string. Access lists are not totally secure because systems can still spoof IP addresses. Access lists do, however, provide the ability to restrict legitimate use. You can provide IPv4 or IPv6 addresses as access lists.

You can use any ASCII characters for the community name.

In the following example, CA eHealth SystemEDGE permits read-write access using the community-string private only to systems with one of the following IP addresses: 45.0.4.10, 45.0.8.12, 198.130.5.7, 0rea2f:fe90:abcd:0000:230:a2f:200:ad01. CA eHealth SystemEDGE treats any other system that attempts to use private as an authentication failure:

```
community private read-write 45.0.4.10 45.0.8.12 198.130.5.7  
ea2f:fe90:abcd:0000:230:a2f:200:ad01
```

Note: The community string of private is used here only as an example. Do not use this value for a read-write community string. Instead, use something like eLtHakSoR97.

Use the examples in this chapter as guidelines for editing the sysedge.cf file to define your own access communities.

Specifying the Access List for SNMPv1 and SNMPv2c Communities

If the access list is empty, CA eHealth SystemEDGE grants access to any system that uses this community string. The following restrictions apply to the access list:

- The access lists specified in sysedge.cf are used for SNMPv1 and SNMPv2c during communication only.
- The SNMPv3 access list is configured separately in the SNMPv3 configuration file sysedgeV3.cf. For more information on how to define access lists for SNMPv3 users, see the appendix "SNMPv3 in CA eHealth SystemEDGE".
- IP addresses must be separated by a space character; you cannot use any other characters, including the newline.
- The maximum length of a community string statement (including any access list) is 1024 characters, which provides enough space for about 60 IPv4 addresses and even less for IPv6 addresses. To configure longer access lists, define separate communities.
- The CA eHealth SystemEDGE agent software distribution includes the edgewatch, edgemon, and emphistory command-line utilities, which act as manager systems, sending requests to the agent. If you are using any of these utilities on the same system on which the agent is installed, include that system's IP address in the access list.

Default Settings

When the CA eHealth SystemEDGE agent is installed, sysedge.cf defines a single access community named public, which provides read-only access to MIB objects. The definition appears as follows:

```
community public read-only
```

Note: Common practice permits read-only access using the community name public.

To modify the values of MIB objects (through SNMP Set operations), you must define a community that has read-write access permissions. For example, you can add a definition like the following to the sysedge.cf file:

```
community private read-write
```

Configuring SNMPv1 Traps

The sysedge.cf file contains definitions for SNMPv1 trap communities, which tell the CA eHealth SystemEDGE agent where to send SNMPv1 trap messages. You can configure the agent to send traps to any number of management systems. The CA eHealth SystemEDGE installation lets you define SNMPv1 trap communities. You can define additional communities as described in this section.

For each management system to which you want to send SNMPv1 traps, add a line specifying either the IP address or host name:

```
trap_community community-name [ip address | hostname] [port-number]
```

You can also specify a port number to send the trap to. If a port number is not specified, the default trap port 162 is used.

For example, add the following lines to send traps with a community-name of mycommunity to two systems, one with the IP address 10.16.5.26 and the other with the hostname atlanta-noc and port number 1692:

```
trap_community mycommunity 10.16.5.26
trap_community mycommunity atlanta-noc 1692
```

Note: sysedge.cf only defines SNMPv1 Trap communities. For information about configuring SNMPv2c and/or SNMPv3 traps, see Configuring SNMPv2c/SNMPv3 Traps in the appendix "SNMPv3 in CA eHealth SystemEDGE".

Specify a SNMPv1 Trap Source

Optionally, you can specify a SNMPv1 trap source. This parameter enables you to specify the IP address as a source of origin in CA eHealth SystemEDGE Trap protocol description units (PDUs). By default, CA eHealth SystemEDGE uses the value returned by the gethostbyname function call. Specify trap_source to override the default behavior.

The agent has only one trap source parameter, so you set this value only once in sysedge.cf, and then all traps in all communities take the specified address. The address you specify must be a valid IP address, but the CA eHealth SystemEDGE agent does not perform error checking to determine whether you have specified a valid address for the specific system you are using as the trap source.

To specify a trap source, add a line in one of the following formats to the sysedge.cf file to specify the source of the trap:

- `trap_source ip-address`
- `trap_source hostname`

For example, to set the trap source to a system with an IP address of 10.0.7.73 and a hostname of system1.empire.com, you can do either of the following:

- To use the IP address, add the following line to sysedge.cf:

```
trap_source 10.0.7.73
```

- To use the hostname, add the following line to sysedge.cf:

```
trap_source system1.empire.com
```

Configuring Authentication Failure Traps

You can configure the CA eHealth SystemEDGE agent to send an Authentication Failure trap whenever it receives an SNMP message whose community name does not match one of the communities recognized by the agent.

By default, the agent does *not* send Authentication Failure traps (no_authen_traps). To configure the agent to send Authentication Failure traps, comment out the no_authen_traps directive in sysedge.cf by putting a pound sign (#) character at the beginning of the line as follows:

```
# no_authen_traps
```

Note: This option applies for authentication failure traps for trap destinations set for SNMPv1 traps (configured in sysedge.cf), SNMPv2c traps (configured in sysedgeV3.cf), and SNMPv3 (configured in sysedgeV3.cf) traps.

Agent Addresses of Traps from SystemEDGE

The source addresses of the traps sent from the CA eHealth SystemEDGE agent will be the address that the agent is bound to. By default, the CA eHealth SystemEDGE agent binds to all of the network interfaces, so the traps sent from the agent will use its first successful IP address.

If SystemEDGE is configured to send the traps to a trap receiver (such as xtrapmon) running on the same local server as the agent, the source address will most likely be a loop back address (127.0.0.1 (for IPv4) or ::1 (for IPv6)).

Configuring Support for Who Table Information

By default, the CA eHealth SystemEDGE agent supports the Who Table, which provides information about users who are currently logged in to a system. For more information about this table, see the chapter “Systems Management MIB.” The disclosure of this type of information can pose a potential security risk; you may want to disable the agent's support for this information.

To disable support for the Who Table, uncomment the following line in sysedge.cf by removing the pound sign (#) character from the beginning of the following line:

```
# no_who_table
```

Configuring Support for User and Group Information

By default, the CA eHealth SystemEDGE agent supports the User table and the Group table, which provide information about the user accounts and user groups that have been configured for the system. The type of information in these tables is similar to the information in the /etc/passwd and /etc/group directories. For more information, see the chapter “Systems Management MIB.”

You might want to disable support for User and Group information in the following cases:

- Your organization considers the disclosure of user and group information to be a potential security issue.
- You have a distributed system with large numbers of users or groups. Because the agent periodically caches this information internally, storing user and group information could consume a significant amount of resources.

To disable support for the User and Group tables, uncomment the following line in sysedge.cf by removing the pound sign (#) character from the following line:

```
# no_usergroup_table
```

Configuring Support for Remote Shell Capability

By default, the CA eHealth SystemEDGE agent supports the Remote Shell group, which permits management systems to remotely instruct the agent to run shell scripts and programs on the system on which the agent is running.

Note: For more information, see the chapter "Systems Management MIB."

The disclosure of this type of information can pose a potential security risk; you may want to disable the agent's support for this information. To disable support for the Remote Shell Group, uncomment the following line in sysedge.cf by removing the pound sign (#) character:

```
# no_remoteshell_group
```

Configuring Alternative Syslog Facilities (UNIX Only)

The CA eHealth SystemEDGE agent, by default, logs all syslog messages to the LOG_DAEMON syslog facility. You can specify a different facility in the configuration file.

Note: Windows does not support the syslog facility. On Windows systems, all syslog output is logged by default to the %SystemRoot%\system32\sysedge.log file. If you are using Windows, see Configuring Alternative Syslog Facilities (Windows Only).

Accepted values are the following:

- kern (LOG_KERN)
- user (LOG_USER)
- mail (LOG_MAIL)
- daemon (LOG_DAEMON)
- auth (LOG_AUTH)
- syslog (LOG_SYSLOG)
- lpr (LOG_LPR)
- local0 (LOG_LOCAL0) to local7 (LOG_LOCAL7)

The following example sends the CA eHealth SystemEDGE agent's syslog messages to the local1 facility. Add this entry near the top of the sysedge.cf file to redirect any syslog messages which are generated by errors in sysedge.cf:

```
syslog_facility local1
```

Configuring Alternative Syslog Facilities (Windows Only)

On Windows systems, the CA eHealth SystemEDGE agent logs all syslog messages to the %SystemRoot%\system32\sysedge.log file by default. That file size is unlimited. However, you can use the syslog_logfile directive to specify an alternative sysedge.log location (and file name), and to put limits on the size of that log file and the number of old log files that the agent saves.

Use the syslog_logfile directive as follows:

```
syslog_logfile filename size number
```

The variables are defined as follows:

filename

Specifies the complete path to the desired log file.

size

Specifies the maximum file size in KB.

number

Specifies the number of log files to preserve for historical purposes. A minimum of two files is recommended.

For example, to instruct the CA eHealth SystemEDGE agent to log messages to the file C:\sysedge\sysedge.log, creating up to two log files with a maximum size of 20 KB each, add the following to the sysedge.cf file:

```
syslog_logfile c:\sysedge\sysedge.log 20 2
```

Configuring Support for Agent Debugging

By default, the CA eHealth SystemEDGE agent logs all syslog messages of priority LOG_INFO or lower. (Lower priority levels signify greater importance.) While UNIX-based agents can change this log-level through the command-line -d option, the CA eHealth SystemEDGE agent on Windows cannot, because no command-line options are available to it. To use the configuration file option to instruct the CA eHealth SystemEDGE agent (on both UNIX and Windows systems) to log messages of priority LOG_DEBUG or lower, add this entry at the top of the sysedge.cf file:

```
sysedge_debug
```

For more information about the syslog facility, see the appendix "Using the syslog Facility".

Note: On Windows systems, all syslog output is logged to the file %SystemRoot%\system32\sysedge.log. For more information, see Configuring Alternative Syslog Facilities (Windows Only) in this chapter.

Configuring Support for Floppy Status Checking

By default, the CA eHealth SystemEDGE agent automatically determines the status of all floppy devices on the system as part of its support for the hrDeviceTable from the Host Resources MIB. On some UNIX systems, however, when you check the status (with the stat command) of a floppy device that contains no media, the console and some system log files display warning messages. To circumvent these warning messages, configure the agent to not check status of floppy drives. To do so, uncomment the following line in sysedge.cf by removing the pound sign (#) character:

```
# no_stat_floppy
```

Configuring Support for Serial Port Status Checking

By default, the CA eHealth SystemEDGE agent automatically determines the status of all serial devices on the host system as part of its support for the hrDeviceTable from the Host Resources MIB.

Some serial applications, however, encounter problems because they cannot handle the opening and closing of serial devices (which are necessary for determining status) by any process other than themselves. For example, some tty and serial applications become confused when another application briefly opens and closes a serial port device.

To circumvent these problematic serial applications, use the no_serial_status configuration option to configure the agent to check only the keyboard and mouse. In this case, the agent returns unknown(1) for the status of serial ports when it is queried by management systems.

To inhibit serial port status checking, uncomment the following line in sysedge.cf by removing the pound sign (#) character from the following line:

```
# no_serial_status
```

Configuring Support for Disk Probing

By default, the CA eHealth SystemEDGE agent automatically determines size, capacity, description, and other properties of disks and CD-ROMs that may be installed on the underlying system. The agent usually uses I/O control functions (for example, UNIX `ioctl`s) to obtain this information. However, on some older UNIX systems (for example, HP-UX), probing of disk devices may cause the agent to block while the driver waits on status information.

You can use the `no_probe_disks` option to avoid potentially lengthy agent blocking. To inhibit disk probing, add the following line to `sysedge.cf`:

```
no_probe_disks
```

Note: If this option is enabled, the agent may be unable to provide disk statistics, capacity information, device descriptions, and status information.

Configuring Support for Actions

By default, the CA eHealth SystemEDGE agent permits the execution of action commands with the self-monitoring tables. The capability to run action commands and scripts can be a potential security issue because the command and scripts can run commands as the root or administrator users. Depending on the local security policies in effect at your site, you may want to disable the agent's support for executing action commands.

For more information about actions, see the following sections:

- Monitor Table Actions in the chapter, "Configuring Threshold Monitoring"
- Process Monitor Table Actions in the chapter "Configuring Process and Service Monitoring"
- Process Group Monitor Table Actions in the chapter "Configuring Process Group Monitoring"
- Log Monitor Table Actions in the chapter "Configuring Log Monitor Table Actions"
- NT Event Monitor Table Actions in the chapter "Configuring Windows Event Monitoring"

To disable support for action execution, uncomment the following line in `sysedge.cf` by removing the pound sign (`#`) character from the following line:

```
# no_actions
```

Disabling Support for Remote File System Checking (UNIX Only)

By default, the CA eHealth SystemEDGE agent makes data about all file systems (local and remote) available through the Systems Management MIB. However, on some UNIX systems, CA eHealth SystemEDGE can be blocked if a file system is mounted from a remote file server that is no longer available (either if it is down or if the network connection between the server and client is down). Unfortunately, there is no way for the CA eHealth SystemEDGE agent to unblock in this situation. To circumvent this blocking, configure the CA eHealth SystemEDGE agent to avoid checking status on remote file systems. To do so, uncomment the following line in sysedge.cf by removing the pound sign (#) character from the following line:

```
# no_stat_nfs_filesystems
```

Note: If you uncomment this line in the sysedge.cf file, it disables the checking of all remote file systems, not just NFS file systems.

Configuring Support for Threshold Monitoring

The CA eHealth SystemEDGE agent includes support for threshold monitoring of MIB objects, including file systems, interfaces, processors, and so on. You can use SNMP Set requests to add entries for threshold monitoring dynamically while the agent is running, or you can define them in the sysedge.cf configuration file that the CA eHealth SystemEDGE agent reads when it starts.

Note: For more information about creating entries in the Monitor table, see the chapter “Configuring Threshold Monitoring.”

Configuring Support for Process Monitoring

By default, the CA eHealth SystemEDGE agent permits SNMP Gets and Sets to the Systems Management Process Monitor table and the Host Resources Running Software table, assuming that queries use a valid community with read-only or read-write permissions (respectively) for SNMPv1 and SNMPv2c communications or valid SNMPv3 user credentials for SNMPv3 communication. For more information on how to configure SNMPv3 for CA eHealth SystemEDGE, see the appendix “SNMPv3 in CA eHealth SystemEDGE”.

Performing SNMP Sets in those tables may be a potential security issue: SNMP Sets can send UNIX processes signals (for example, KILL) and can terminate processes on Windows systems. Depending on the local security policies in effect at your site, you may want to disable the agent's support for SNMP Sets in these tables. For more information about these tables, see the chapters “Systems Management MIB” and “Host Resources MIB.”

To disable support for SNMP Sets to the Process Monitor table, uncomment the following line in sysedge.cf by removing the pound sign (#) character from the following line:

```
# no_process_sets
```

Performing SNMP Gets in those tables can also be a potential security issue: SNMP Gets can discover the processes running on the underlying system. Depending on the local security policies in effect at your site, you may want to disable the agent's support for SNMP Gets in these tables. For more information about these tables, see the chapters "Systems Management MIB" and "Host Resources MIB".

To disable support for SNMP Gets and Sets to the Process Monitor table, add the following line to sysedge.cf:

```
no_process_table
```

Monitoring Applications, Processes, and Services

The CA eHealth SystemEDGE agent can also monitor applications, processes, and Windows services by creating entries in the Process Monitor table. You can dynamically add entries through an SNMP Set request while the agent is running, or you can define them in the sysedge.cf configuration file that the agent reads when it starts.

Monitoring Process Attributes

The watch process configuration file directive automatically configures the agent to monitor a process attribute that you specify. You identify the process to be monitored using regular expressions to match the process name and the attribute of the process that you want to monitor. The CA eHealth SystemEDGE agent automatically determines the process ID for the specified process and then creates the appropriate entry in the agent's Process Monitor table. When you use the watch process directive, you need not know the process ID or to use SNMP Set requests to add an entry to the Process Monitor table.

Note: For more information about creating entries in the Process Monitor table, see the chapter "Configuring Process and Service Monitoring."

Monitoring Windows Services

The watch ntservice configuration file directive automatically configures the agent to monitor a Windows service to verify that it is running. You identify the Windows service that you want to monitor, and the CA eHealth SystemEDGE agent automatically determines the service index from the NT Service MIB table and creates the appropriate entry in the agent's Process Monitor table.

Note: For more information about the NT Service MIB table, see the chapter "Systems Management MIB."

Configuring Support for Process Group Monitoring

The flexible Process Group Monitor table of the Systems Management MIB enables you to dynamically configure the CA eHealth SystemEDGE agent to monitor groups of processes running on the underlying system. You select the process group, regular expression, and interval, and the agent uses that information to monitor those process groups. For example, the agent can determine what processes exist in each group and whether the group membership changes. If components of an application start or fail, or if members leave a group or are added to a group, the CA eHealth SystemEDGE agent can automatically notify the NMS.

Note: For more information about creating entries in the Process Monitor table, see the chapter "Configuring Process Group Monitoring."

Configuring Support for Log File Monitoring

The CA eHealth SystemEDGE agent includes a log file monitoring capability that lets you instruct the agent to monitor log files continuously for the appearance of user-specified regular expressions, and to notify the management system with a trap message if the agent finds a match. You can specify entries for log file monitoring dynamically (through SNMP Set requests) while the agent is running, or you can define them in the sysedge.cf configuration file.

Note: For more information about creating entries in the Process Monitor table, see the chapter "Configuring Log File Monitoring".

Configuring Support for Windows Event Log Monitoring (Windows Only)

The CA eHealth SystemEDGE agent includes a Windows Event Monitoring capability that lets you instruct the agent to continuously monitor Windows event logs in much the same way that it monitors textual log files. When a matching event is generated on the system, the agent notifies the management system with a trap message and can run an action command to immediately handle the event. Because Windows events include several identifying characteristics in addition to the textual message, this monitoring capability enables you to specify more sophisticated types of matches.

Note: For more information about creating entries in the Process Monitor table, see the chapter "Configuring Windows Event Monitoring".

Configuring History Collection

The CA eHealth SystemEDGE agent can track the value of various integer-based MIB objects (counters, gauges, and so on) over time and can store them for later retrieval. You can define entries for history collection in the sysedge.cf configuration file that the CA eHealth SystemEDGE agent reads when it starts.

Note: For more information about creating entries in the Process Monitor table, see the chapter "Configuring History Collection".

Configuring User and Group Permissions for Subprograms (UNIX Only)

By default, the CA eHealth SystemEDGE agent runs subprograms (for example, remote shell, action, and extension object invocations) with its effective user and group permissions--normally root. Depending on the local security policies in effect at your site, you may want to set the agent to use actions and extension objects that run with different user and group permissions.

To run subprograms with the effective user and group permissions of a user other than root, add the following statements to your CA eHealth SystemEDGE agent configuration file:

```
subprogram_user_name concord
subprogram_group_name sysmgmt
```

In these examples, all subprograms run with the effective permissions of the user concord and group sysmgmt. The user and group names that you specify *must be valid* on the underlying system.

Note: If either the user name or group name is incorrect, CA eHealth SystemEDGE disables all subprogram functionality. That is, the agent does not support actions, extension MIB objects, and remote-shell capabilities.

Configuring the SNMP Bind Address

By default, CA eHealth SystemEDGE binds to all interfaces (*/*UDP-161). You can bind CA eHealth SystemEDGE to a specific interface by using the `bind_address` token as follows:

```
bind_address ip-address
```

The `bind_address` token accepts IPv4 and IPv6 addresses.

Examples

Entering the following line in the `sysedge.cf` file binds to the 10.1.0.202 address only:

```
bind_address 10.1.0.202
```

Entering the following line in the `sysedge.cf` file binds to the ea2f:fe90:abcd:0000:230:a2f:200:ad01 address only:

```
bind_address ea2f:fe90:abcd:0000:230:a2f:200:ad01
```

Configuring IP Family for SNMP User Datagram Protocol Communications

Hosts can have multiple IP family source sockets and multiple family destination addresses. The CA eHealth SystemEDGE `sysedge_ip_family` options lets you set the preferred method for SNMP User Datagram Protocol (UDP) communications.

The `sysedge_ip_family` option has the following possible values:

1

Specifies that the agent tries to use only IPv4 as the preferred SNMP UDP communication method.

2

Specifies that the agent tries to use only IPv6 as the preferred SNMP UDP communication method.

3

Specifies that the agent tries to use both IPv4 and IPv6 as the preferred SNMP UDP communication methods. This is the default method.

For example, modify the line pertaining to `sysedge_ip_family` in the `sysedge.cf` file as follows to use only IPv4 mode for SNMP UDP communications:

```
sysedge_ip_family 1
```

Note: If the `sysedge_ip_family` option is set to a mode not configured on the host, CA eHealth SystemEDGE overrides the `sysedge_ip_family` setting and uses either IPv4 or IPv6 (whichever is configured).

Enabling Federal Information Processing Standard Mode

You can configure how the CA eHealth SystemEDGE agent handles encryption using the following `sysedge_fips_mode` options:

0

Enables non-FIPS mode. This is the default if `sysedge_fips_mode` is not configured.

1

Specifies that the agent operates in FIPS co-existence mode.

2

Specifies that the agent operates in FIPS only mode.

For example, modify the line pertaining to `sysedge_fips_mode` in the `sysedge.cf` file as follows to run the agent using only FIPS-certified protocols and FIPS-certified libraries:

```
sysedge_fips_mode 2
```

For detailed information about how to enable FIPS mode, see the appendix "Using FIPS 140-2 Encryption".

Configuring Support for CA eHealth AIMs

The CA eHealth SystemEDGE agent provides a plug-in architecture through which you can load optional CA eHealth AIMs at initialization. These CA eHealth AIMs provide an extensible and flexible approach to supporting application-specific MIB variables.

By default, the CA eHealth SystemEDGE agent does not load any AIMs at initialization time. You can edit the `sysedge.cf` file to specify which AIMs the agent should load. You must specify absolute paths for the CA eHealth SystemEDGE agent to find the AIM to load.

Note: The Top Processes AIM is included with the CA eHealth SystemEDGE distribution on every platform. You can set the agent to load the Top Processes AIM during the CA eHealth SystemEDGE agent installation.

To load the AIM for Top Processes from the standard Solaris distribution directory, for example, enter the following in the sysedge.cf file:

```
sysedge_plugin /opt/EMPSysedge/plugins/topprocs/topprocs-sol32bit.so
```

To load the AIM for Top Processes module from the standard Windows distribution directory, enter the following in the sysedge.cf file:

```
sysedge_plugin c:\sysedge\plugins\topprocs\topprocs.dll
```

Note: If you selected the option for configuring Top Processes during the CA eHealth SystemEDGE installation, this line is automatically added to the sysedge.cf file.

For information about enabling other AIMs, see the documentation for that AIM.

Configuring Support for the Monitored Windows AIM

The Monitored Windows AIM is used for time-based control of the activity of CA eHealth SystemEDGE monitoring. For more information about this AIM, see the appendix "Using the Monitored Windows AIM".

Configuring Support for Linux Free Memory

Linux free memory is calculated as total physical memory less memory in use. By default, memory in use includes system buffers and disk cache.

Cache and system buffers can be reclaimed by the operating system if memory is needed for processes. For this reason, some choose to view free memory as including memory which is used by the operating system for caching or system buffers. The `linux_freemem_include` directive includes these values in the Linux free memory calculation. The directive supports the following two options:

- *buffers* include system buffers in the free memory calculation.
- *cached* include cached memory in the free memory calculation.

At least one of the previous options must be specified. The options may be specified in any order and must be separated by one or more spaces.

For example, to include both buffers and cached memory in free memory add the following line to sysedge.cf:

```
linux_freemem_include buffers cached
```

Recommendations for Configuring Security

Following are the recommended configuration options for implementing security:

no_who_table

See Configuring Support for Who Table Information.

no_usergroup_table

See Configuring Support for User and Group Information.

no_remoteshell_group

See Configuring Support for Remote Shell Capability.

no_process_sets

See Configuring Support for Process Monitoring.

Following are additional configuration options that can help implement security:

no_actions

See Configuring Support for Actions.

no_process_table

See Configuring Support for Process Monitoring.

subprogram_user_name

See Configuring User and Group Permissions for Subprograms (UNIX Only).

subprogram_group_name

See Configuring User and Group Permissions for Subprograms (UNIX Only).

Using the SystemEDGE Control Panel for Windows

As a standalone Windows service, CA eHealth SystemEDGE has its own SystemEDGE Control Panel. To view the SystemEDGE Control Panel, open the Control Panel dialog and double-click CA eHealth SystemEDGE.

You can use the SystemEDGE Control Panel to perform the following tasks:

- Start and stop the agent.
- View community strings and trap destinations.
- Open the configuration and license files.
- View the CA eHealth SystemEDGE log file.
- Run the diagsysedge utility.
- View the *User Guide* and *Release Notes*.

Chapter 4: Starting the CA eHealth SystemEDGE Agent

This chapter explains how to start the CA eHealth SystemEDGE agent. Before you do, you must configure the agent for your environment. For more information, see the chapter "Configuring the CA eHealth SystemEDGE Agent."

To start the agent manually, see [Starting the Agent Manually](#). To start the agent at system boot, see [Starting the Agent Automatically at System Boot](#).

Note: After you start the agent as described in this chapter, you can use the `diagsysedge.exe` program to verify that your agent is running. For more information, see [Using diagsysedge.exe](#) in the chapter "Troubleshooting and Usage Suggestions."

This section contains the following topics:

[Starting the Agent Manually](#) (see page 85)

[Starting the Agent Automatically at System Boot](#) (see page 89)

[Logging Agent Operation Messages](#) (see page 91)

Starting the Agent Manually

You can start the agent manually for both Windows and UNIX systems.

Start CA eHealth SystemEDGE on Windows Systems

For Windows systems, you can manually start CA eHealth SystemEDGE from the command line, the Services Control Panel, or the SystemEDGE Control Panel.

To start CA eHealth SystemEDGE from the command line, enter the following:

```
net start sysedge
```

The SystemEDGE agent starts as a service in the background using the port that you specified during the installation and the default configuration files `/etc/sysedge.cf`, `/etc/sysedge.mon`, and `<install-directory>/sysedge/config/sysedgeV3.cf`.

You can also run SystemEDGE in the foreground using non-default options or non-default configuration files. For more information, see [Command Line Options for Windows Systems](#).

To start CA eHealth SystemEDGE from the Services Control Panel

1. Select Start, Settings, Control Panel, Administrative Tools, Services.
2. Right-click SystemEDGE, and select Start.

To start CA eHealth SystemEDGE from the SystemEDGE Control Panel

1. Select Start, Settings, Control Panel.
2. Double-click eHealth SystemEDGE, and click Start Agent.

Note: In the Windows XP Category View, the SystemEDGE control panel is under "Network and Internet Connections." In the Windows XP Classic View, the SystemEDGE control panel icon is at the main level of the display.

Command Line Options for Windows Systems

This section describes the command line options for starting the CA eHealth SystemEDGE agent. To start the CA eHealth SystemEDGE agent in the foreground from the Windows command line, change directory to the CA eHealth SystemEDGE installation directory, and then enter the following:

```
sysedge [-e SNMPV3 config file] [-f config file] [-l license file] [-m monitor file] [-p port] [-h] [-t]
```

-e *SNMPV3 config file*

Specifies the path name to use for the SNMPv3 configuration file instead of the default \sysedge\config\sysedgeV3.cf file.

-f *config file*

Specifies the path name to use for the sysedge.cf file instead of the default %SYSTEM32%\sysedge.cf file.

-l *license file*

Reads license related information from the license file instead of the default %SYSTEM32%\sysedge.lic file. This option is intended for the agent's internal use only.

-m *monitorfile*

Specifies the path name to use for the monitor table configuration file instead of the default %SYSTEM32%\sysedge.mon file.

-p *port*

Specifies the port to listen for the incoming SNMP messages on instead of the standard SNMP port 161. Port UDP/1691 is reserved for use as an alternate port for running the CA eHealth SystemEDGE agent.

-h

Displays help for the 'sysedge' command. This option lists the available command line options for starting 'sysedge'.

-t

Runs the agent in packet-trace mode. This option causes the agent to write a packet dump of each SNMP PDU received or transmitted by the agent to standard output (file descriptor 1).

Start CA eHealth SystemEDGE on UNIX Systems

For UNIX systems, you can start the CA eHealth SystemEDGE agent using command line options or using an automated service startup script.

Command Line Options for UNIX Systems

This section describes the command line options for starting the CA eHealth SystemEDGE agent. The usage options are as follows:

```
sysedge [-b] [-d] [-f config file] [-e SNMPv3 config file] [-l license file] [-m  
monitor file] [-p port] [-h] [-t]
```

-b

Runs the agent in the background. The CA eHealth SystemEDGE agent runs as a daemon process and disconnects from the controlling terminal. Use this flag when starting the CA eHealth SystemEDGE agent from a startup script.

-d

Runs the agent in debug mode. This option causes the agent to log debug-level messages with the syslog facility. All of the critical events are logged in syslog, and all of the SNMP messages that come to the agent are logged to the `sysedge_snmp.log` file. For more information about syslog, see the appendix "Using the syslog Facility."

-f *configfile*

Reads configuration files from the *config file* instead of the default `/etc/sysedge.cf` file.

-e *SNMPV3 config file*

Reads SNMPv3 configuration from the SNMPV3 config file instead of the default `<sysedge-install-dir>/sysedge/config/sysedgeV3.cf` file.

-l *license file*

Reads license related information from the license file instead of the default `/etc/sysedge.lic` file. This option is intended for the agent's internal use only.

-m *monitorfile*

Uses the monitor table configuration file *monitor file* instead of the default `/etc/sysedge.mon` file.

-p *port*

Listens for incoming SNMP messages on *port* instead of on the standard SNMP port 161. Port UDP/1691 is reserved for use as an alternate port for running the CA eHealth SystemEDGE agent.

-h

Displays help for the command. This option lists the available command line options.

-t

Runs the agent in packet-trace mode. This option causes the agent to write a packet dump of each SNMP PDU received or transmitted by the agent to standard output (file descriptor 1).

Service Startup Script for UNIX Systems

Using CA eHealth SystemEDGE service startup scripts, you can start the agent automatically. It starts the agent on the port number that was configured during the installation.

Following is the service script name for each UNIX platform:

Solaris (SPARC/Intel/AM64)

/etc/init.d/sysedge

Linux x86/IA64

/etc/init.d/sysedge

HP-UX PA-RISC/IA64

/sbin/init.d/sysedge

HP - Tru64 (DEC)

/sbin/init.d/sysedge

AIX

/etc/rc.d/sysedge

The following arguments are accepted for the startup scripts:

start

Starts the CA eHealth SystemEDGE agent.

stop

Stops the CA eHealth SystemEDGE agent.

restart

Stops and restarts the CA eHealth SystemEDGE agent.

status

Checks if the CA eHealth SystemEDGE agent is started.

Examples

```
/etc/init.d/sysedge start
```

```
/sbin/init.d/sysedge stop
```

```
/etc/rc.d/sysedge status
```

Starting the Agent Automatically at System Boot

The CA eHealth SystemEDGE installation automatically copies the scripts to the system startup area so that they can be used to start the agent automatically whenever the system is booted.

Some systems may require additional steps to make the agent start automatically whenever the system is booted.

This section includes instructions for the following operating systems:

- Solaris
- Windows
- HP-UX
- Linux
- AIX
- Tru64 UNIX

Starting the Agent Automatically for Solaris Systems

For Solaris 2.x systems, the CA eHealth SystemEDGE agent's config subdirectory contains a script named S99sysedge. The installation process puts this script in /etc/rc2.d to start the agent automatically at boot time.

If you install the agent in a directory other than the default (/opt/EMPsysedge), you must edit the script file to include the correct installation directory. In addition, you must also edit the service startup script file (/etc/init.d/sysedge) to include the correct directory for the agent's configuration files (sysedge.lic, sysedge.cf, and sysedge.mon).

Starting the Agent Automatically for Windows Systems

The CA eHealth SystemEDGE service starts by default at system boot time. To verify that the agent is set to start automatically, open the Services Control Panel, right-click SystemEDGE, and select Properties. On the General tab, select Automatic from the Startup type list, and then click OK.

Starting the Agent Automatically for HP-UX Systems

For HP-UX systems, the CA eHealth SystemEDGE agent's config subdirectory contains a service startup script named sysedge. The installation process puts this script in /sbin/init.d to start the agent automatically at boot time.

If you have installed the agent in a directory other than the default (/opt/EMPsysedge), you must edit the script file to include the correct installation directory. In addition, you must also edit the service startup script file (/sbin/init.d/sysedge) to include the correct directory for the agent's configuration files (sysedge.lic, sysedge.cf, and sysedge.mon) if you installed them in a directory other than the default (/etc).

To disable the native HP-UX agent, edit the following statements in the script file SnmpMaster in /etc/rc.config.d/ as described below:

- 'SNMP_MASTER_START=1' to 'SNMP_MASTER_START=0'
- 'SNMP_HPUNIX_START=1' to 'SNMP_HPUNIX_START=0'
- 'SNMP_MIB2_START=1' to 'SNMP_MIB2_START=0'

Starting the Agent Automatically for Linux Systems

For Linux systems, the CA eHealth SystemEDGE agent's config subdirectory contains a service startup script named sysedge. The installation process puts this script in /etc/init.d to start the agent automatically at boot time.

If you have installed the agent in a directory other than the default (/opt/EMPsysedge), you must edit the service startup script file (/etc/init.d/sysedge) to include the correct installation directory. In addition, you must also edit the script file to include the correct directory for the agent's configuration files (sysedge.lic, sysedge.cf, and sysedge.mon) if you installed them in a directory other than the default (/etc).

To stop the native Linux SNMP agent, edit /etc/rc.d/init.d/snmpd by placing "exit 0" (no quotes) at the first line of execution in the file.

Starting the Agent Automatically for AIX Systems

For AIX systems, the CA eHealth SystemEDGE agent's config subdirectory contains a service startup script file named `sysedge`. The installation process copies the script as `/etc/rc.d/rc2.d/S990sysedge` to enable the agent to be started automatically at boot time.

To start the agent automatically for AIX systems, you must add the commands for starting the agent to the local boot file `/etc/rc.tcpip`.

To add the commands for starting the agent and to include the `sysedge` agent process, enter the following:

```
/usr/lpp/EMPsysedge/bin/sysedge -b
```

When you enter this command, you must include the full directory path and file name for `sysedge` on your system.

Starting the Agent Automatically for Tru64 UNIX Systems

For Tru64 UNIX systems, the CA eHealth SystemEDGE agent's config subdirectory contains a service startup script file named `sysedge`. The installation process puts this script in `/sbin/rc3.d` to start the agent automatically at boot time.

If you have installed the agent in a directory other than the default (`/usr/opt/EMPsysedge`), you must edit the script file to include the correct installation directory. In addition, you must also edit the script file to include the correct directory for the agent's configuration files (`sysedge.lic`, `sysedge.cf`, and `sysedge.mon`) if you installed them in a directory other than the default (`/etc`).

To disable the native Tru64 UNIX agent, edit `/sbin/init.d/snmpd` by entering an `exit 0` at the first line of execution in the file.

Logging Agent Operation Messages

This section explains how to log agent operation messages.

Logging Messages for UNIX

The CA eHealth SystemEDGE agent uses the UNIX syslog facility to log informational messages and error conditions that it may encounter during operation. The syslog daemon typically logs these messages to the `/var/adm/messages` text file, depending on how the syslog daemon is configured on your system. By default, the agent daemon uses syslog to log messages of priority levels informational through emergency. If you are running the agent in debug mode through the `-d` runtime command line option, the agent also logs messages of priority level debug.

For information about configuring the syslog daemon on your system to log messages from daemon processes like `sysedge` to the `/var/adm/daemon-log` text file, see the appendix "Using the syslog Facility". For more information about syslog, see the Man pages: `syslog(3)`, `syslog.conf(5)`, and `syslogd(8)`.

If you are running the agent in debug mode through the `-d` runtime command line option, the agent logs all of the SNMP messages (critical as well as informational) to the log file `sysedge_snmp.log` in the 'bin' sub-directory of the CA eHealth SystemEDGE agent's installation.

Logging Messages for Windows

The CA eHealth SystemEDGE agent logs informational messages and error conditions that it may encounter during operation in the `%SystemRoot%\system32\sysedge.log` file. For information about possible issues that may arise during CA eHealth SystemEDGE agent operation, see the `sysedge.log` file.

Chapter 5: Using the CA eHealth SystemEDGE Agent with Other SNMP Agents

This chapter describes how to use the CA eHealth SystemEDGE agent with other SNMP agents.

This section contains the following topics:

[Supporting Multiple SNMP Agents](#) (see page 93)

[Using the CA eHealth SystemEDGE Agent with the Solaris Solstice Enterprise Agent](#) (see page 95)

[Using the CA eHealth SystemEDGE Agent with the Microsoft Windows SNMP Agent](#) (see page 96)

[Using the CA eHealth SystemEDGE Agent with the HP SNMP Agent](#) (see page 97)

[Using the CA eHealth SystemEDGE Agent with the AIX SNMP Agent](#) (see page 97)

[Using the CA eHealth SystemEDGE Agent with the Tru64 UNIX SNMP Agent](#) (see page 98)

Supporting Multiple SNMP Agents

By default, SNMP agents attempt to use UDP port 161, but only a single process can bind to a given port at any one time. That is, only one SNMP agent can use UDP port 161 at a time. If you are running multiple SNMP agents simultaneously, you must develop strategies for their coexistence.

The following are potential solutions for enabling several SNMP agents to share UDP port 161:

- Multiplex UDP port 161 among several SNMP agents, usually in a master/slave relationship.

Note: UDP port 1691 is reserved for use by the CA eHealth SystemEDGE agent. You can configure the agent to use that port instead of UDP/161. For more information about starting the CA eHealth SystemEDGE agent on an alternate port, see the chapter "Starting the CA eHealth SystemEDGE Agent".
- Run agents simultaneously if each binds to a separate UDP port and management software is configured to communicate with them individually.

- Code agents to conform to every existing proprietary master/subagent API. This solution has an exceedingly high cost in terms of coding, testing, and licensing the numerous proprietary APIs.

The most common notion of agent multiplexing involves a master agent bound to UDP port 161 that communicates with separate subagents or slave agents that implement different MIBs. The master and subagents communicate using a defined protocol for registering the subagent and exchanging messages between them. Several protocols exist for governing such a communication protocol between Master and subagent.

Agent Multiplexing

The two most common multiplexing protocols are SNMP-Distributed Protocol Interface (DPI) and SNMP multiplexing (SMUX). SNMP-DPI is an extension to SNMP agents that permits users to dynamically add, delete, or replace management variables in the local MIB through a subagent without having to recompile the SNMP agent. SMUX is a session-management protocol that provides a lightweight communication channel from the underlying transport layer to the application layer by multiplexing data streams on top of a reliable stream-oriented transport.

Note: For more information about SNMP-DPI, see RFC 1592. For more information about SMUX, see RFC 1227.

Neither protocol is standard or dominant in the marketplace. The IETF is currently working on a standardized approach based on SNMP-DPI named AgentX. These RFCs define only the protocol used between Master and subagent. They do *not* define a set of APIs. Because the specifications for DPI and SMUX are publicly available, an agent developer can implement a master or subagent that conforms to either specification without encountering licensing and royalty fees when using them with another vendor's master or subagent. However, because neither protocol is standard, an agent developer has to support both protocols to make sure that the greatest amount of master-slave agent multiplexing interoperability.

Several proprietary, non-standard, agent-multiplexing solutions exist, although they are not technically true agent multiplexing. These solutions are based on the notion of monolithic agents, whereby subagents are actually linked into the Master agent's executable binary or its address space while they are running. This task can be accomplished without Master agent source code, as long as the subagents adhere to the proprietary API.

There are several drawbacks to this approach:

- The agent developer may have to support multiple code bases because the proprietary APIs are often incompatible with those used within other agent source bases.
- Because the API is proprietary, these subagents cannot communicate with other, more open specifications, such as DPI and SMUX.
- The development and deployment of subagents usually involve licensing and royalty fees, which are usually charged on a per-CPU basis. That means that the more subagents you deploy, the more the user must pay in licensing and royalty fees. In addition, because this solution is proprietary, every developer of the Master agent and subagents must purchase those APIs to make the pieces interoperate.

Using the CA eHealth SystemEDGE Agent with the Solaris Solstice Enterprise Agent

When you run the CA eHealth SystemEDGE agent on UDP port 1691, you can use it as a subagent under the Solstice Enterprise Agent (SEA). The CA eHealth SystemEDGE agent installation package installs the appropriate configuration file. Following are the SEA configuration files:

sysedge.reg

Specifies the MIB branches to query the CA eHealth SystemEDGE agent and the UDP port number to send SNMP queries.

sysedge.rsrc

Specifies the CA eHealth SystemEDGE agent type and the location of its corresponding registration file (by default, /etc/snmp/conf/sysedge.reg).

To configure the CA eHealth SystemEDGE agent to run as a subagent under the SEA

1. Install the SEA packages that correspond to your version of Solaris and the underlying hardware architecture.
2. Copy the CA eHealth SystemEDGE agent SEA configuration file into the SEA configuration directory, /etc/snmp, by entering the following commands from the directory where you installed the CA eHealth SystemEDGE agent:

```
cp config/sysedge.reg /etc/snmp/conf
cp config/sysedge.rsrc /etc/snmp/conf
```

3. Restart the CA eHealth SystemEDGE agent so that it will use port UDP/1691 instead of the default UDP/161.

The S99sysedge script attempts to determine whether to use the CA eHealth SystemEDGE agent with the SEA. It does so by checking for the existence of the /etc/snmp/conf directory and the /etc/rc2.d/K76snmpdx file. If the script determines that it should use both agents, it instructs the CA eHealth SystemEDGE agent to use port UDP/1691.

4. Restart the SEA multiplexor by entering the following commands:

```
/etc/rc2.d/K76snmpdx stop  
/etc/rc2.d/K76snmpdx start
```

Note: Although the SEA multiplexes SNMP Get, GetNext, and Set operations, it does not correctly support SNMP row-creation operations across subagents because they occur as side effects of Set operations on non-existent rows. Consequently, when you are using utilities such as edgwatch, edgemon, and emphistory to create rows in their respective SNMP tables, query the CA eHealth SystemEDGE agent directly (not the SEA master agent). You can instruct those utilities to query UDP port 1691 by appending: 1691 to the hostname or IP address when you invoke the utilities.

Using the CA eHealth SystemEDGE Agent with the Microsoft Windows SNMP Agent

The Microsoft Windows SNMP Agent is a master agent that multiplexes among subagents linked to its address space at run-time. Subagents are implemented as Windows DLLs. Upon initialization, the Microsoft master agent uses registry settings to determine which subagents (and corresponding DLLs) to load. Upon initialization, the subagents inform the Master agent which MIB branches they implement.

The CA eHealth SystemEDGE agent is no longer a subagent of the Microsoft SNMP agent. It can coexist with the Microsoft Windows extensible agent.

Take either of the following coexistence approaches:

- Run the Microsoft and CA eHealth SystemEDGE agents together on the system, running the CA eHealth SystemEDGE agent on port 1691 and the Microsoft agent on port 161.
- Turn off or disable the Microsoft Master agent, and run the CA eHealth SystemEDGE agent on port 161.

Using the CA eHealth SystemEDGE Agent with the HP SNMP Agent

HP-UX systems ship with an extensible agent and subagents that provide support for MIB-II and the HP (rather limited) private-enterprise MIB. The Master/subagent interface is based on a proprietary API that the CA eHealth SystemEDGE agent does not support.

Take either of the following coexistence approaches:

- Run the HP and CA eHealth SystemEDGE agents together on the system, running the CA eHealth SystemEDGE agent on port UDP/1691.
- Turn off or disable the HP master or subagent and run the CA eHealth SystemEDGE agent on port UDP/161.

Using the CA eHealth SystemEDGE Agent with the AIX SNMP Agent

AIX Release 4.1 and later systems ship with a MIB-II SNMP agent that also supports SMUX. You can run the CA eHealth SystemEDGE agent in addition to or in place of the AIX agent.

Take either of the following coexistence approaches:

- Run the CA eHealth SystemEDGE and AIX agents together. To do so, make sure that the CA eHealth SystemEDGE agent is invoked with the `-p 1691` option in the AIX TCP/IP startup script, `/etc/rc.tcpip`.
- Run the CA eHealth SystemEDGE agent instead of the AIX agent. To do so, comment out the AIX agent's invocation in `/etc/rc.tcpip` by adding a pound sign (`#`) at the beginning of the following lines:

```
# Start up the Simple Network Management Protocol (SNMP) daemon  
# start /usr/sbin/snmpd "$src_running"
```

Using the CA eHealth SystemEDGE Agent with the Tru64 UNIX SNMP Agent

Tru64 UNIX systems ship with an extensible agent and subagents that provide support for MIB-II and for the Tru64 UNIX implementation of the Host Resources MIB. The Master/subagent interface is based on a proprietary API that the CA eHealth SystemEDGE agent does not support.

Take either of the following coexistence approaches:

- Run the agents together on the system, running the CA eHealth SystemEDGE agent on port UDP/1691.
- Turn off or disable the Tru64 UNIX Master/subagent, and run the CA eHealth SystemEDGE agent on port UDP/161.

Using the CA eHealth SystemEDGE Agent with the Compaq Insight Manager

You can run the CA eHealth SystemEDGE agent and the Compaq Insight Manager (CIM) on the same system. When you are deploying CA eHealth SystemEDGE agents from CA eHealth AdvantEDGE View to a system that includes CIM, CA eHealth AdvantEDGE View stops the following CIM services:

- Compaq Foundation Agent
- Compaq Web Agent
- Compaq Storage Agent
- Compaq Server Agent
- Compaq NIC Agent

After the CA eHealth SystemEDGE agent and CA eHealth AIMs are deployed, CA eHealth AdvantEDGE View restarts all of these services.

Chapter 6: Systems Management MIB

This chapter gives examples of management information available through the enterprise-specific Systems Management MIB.

This section contains the following topics:

[Systems Management MIB Information](#) (see page 100)

[Host System Information](#) (see page 101)

[Mounted Devices](#) (see page 101)

[Kernel Configuration](#) (see page 103)

[Boot Configuration](#) (see page 104)

[Streams Group](#) (see page 104)

[User Information](#) (see page 106)

[Group Information](#) (see page 107)

[Process Information](#) (see page 108)

[Who Table Information](#) (see page 110)

[Remote Command Execution](#) (see page 111)

[Kernel Performance Statistics](#) (see page 111)

[Interprocess Communication: Queues, Shared Memory, and Semaphores](#)
(see page 113)

[Message Buffer Allocation and Usage Statistics](#) (see page 115)

[Stream Buffers](#) (see page 116)

[I/O Buffer Cache](#) (see page 117)

[RPC Group](#) (see page 118)

[NFS Group](#) (see page 118)

[Windows-Specific Groups](#) (see page 120)

[Monitor Table](#) (see page 134)

[Process Monitor Table](#) (see page 135)

[Process Group Monitor Table](#) (see page 135)

[Log Monitor Table](#) (see page 136)

[History Table](#) (see page 136)

[Disk Statistics Group](#) (see page 139)

[CPU Statistics Group](#) (see page 141)

[Extension Group](#) (see page 142)

Systems Management MIB Information

The Systems Management MIB modules defines a collection of objects for managing host systems. The MIB is organized into section for the following types of information:

- Host configuration
- Kernel configuration
- Mounted devices
- Users and groups
- Remote command execution
- Processes
- Streams
- Performance monitoring
- Interprocess communications (IPC)
- NFS and RPC statistics
- Buffer statistics for network buffers
- Streams buffers
- I/O buffer cache

The MIB sections are described in more detail in the following sections. The illustrations show sample output from the CA eHealth SystemEDGE agent daemon, `sysedge`, running on a Solaris SPARC server. For a complete description of the MIB objects, see the `empire.asn1` file in the 'doc' subdirectory of the CA eHealth SystemEDGE agent installation. For information about platform-specific support, see the *Release Notes*.

Host System Information

You can retrieve values of the objects in the Host System group to determine the following types of information:

- Hostname
- Operating system version and release number
- CPU type
- Amount of memory
- Version of the CA eHealth SystemEDGE agent

The following list provides sample values for each MIB object:

- nodename(1): pluto
- cpu(2): sun4u
- memory(3): 16280
- hostid(4): 57212aab
- osyer(5): Generic_103640-31
- osrel(6): 5.8
- agentVersion(8): SystemEDGE Agent Release 4.3

Mounted Devices

For information about the devices and file systems mounted on the host, retrieve the Mounted Devices table. Each row in the table represents a currently mounted device and contains columns that represent the following types of information:

- Device's mount point
- Block size
- Total number of blocks
- Total number of free blocks
- Total number of files
- Total number of free files
- Percent capacity (percent used)

The following illustration shows a sample Mounted Devices table:

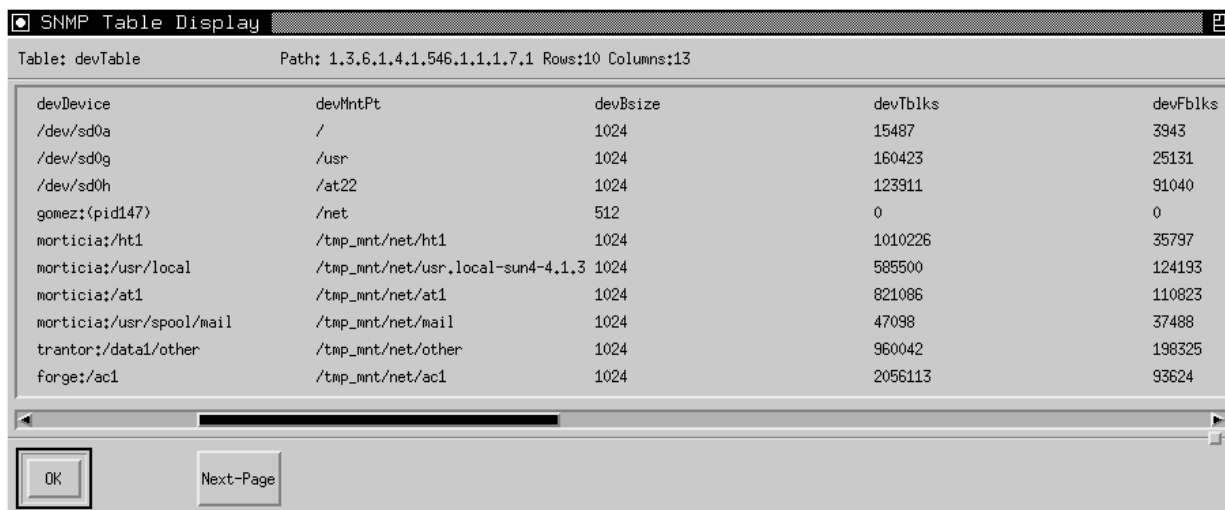


Table: devTable Path: 1.3.6.1.4.1.546.1.1.1.7.1 Rows:10 Columns:13

| devDevice | devMntPt | devBsize | devTbks | devFbks |
|--------------------------|-----------------------------------|----------|---------|---------|
| /dev/sd0a | / | 1024 | 15487 | 3943 |
| /dev/sd0g | /usr | 1024 | 160423 | 25131 |
| /dev/sd0h | /at22 | 1024 | 123911 | 91040 |
| gomez:(pid147) | /net | 512 | 0 | 0 |
| morticia:/ht1 | /tmp_mnt/net/ht1 | 1024 | 1010226 | 35797 |
| morticia:/usr/local | /tmp_mnt/net/usr.local-sun4-4.1.3 | 1024 | 585500 | 124193 |
| morticia:/at1 | /tmp_mnt/net/at1 | 1024 | 821086 | 110823 |
| morticia:/usr/spool/mail | /tmp_mnt/net/mail | 1024 | 47098 | 37488 |
| trantor:/data1/other | /tmp_mnt/net/other | 1024 | 960042 | 198325 |
| forge:/ac1 | /tmp_mnt/net/ac1 | 1024 | 2056113 | 93624 |

OK Next-Page

File System Space Monitoring

The devCapacity column, which shows the percentage of file system space being used, is ideal for checking for file systems at risk of becoming too full. You can use the Monitor table to instruct the agent to monitor the devCapacity values for you and to send a trap to the management system if the value goes above the threshold value that you select. For example, you can have the agent monitor the /usr directory and send a trap if it becomes greater than 90% full. For more information about setting thresholds, see the chapter “Configuring Threshold Monitoring.”

Unmount a Mounted Device

You can unmount a mounted device by setting the devUnmount column for that device to the value delete(1). When you do so, the agent unmounts the device and removes its entry from the Mounted Devices table.

Note: When you are unmounting devices, use a community name that grants you read-write access.

Kernel Configuration

You can identify the version of the kernel running on the system and determine how the kernel is configured by retrieving objects in the Kernel Configuration group. Kernel configuration parameters include the following:

- Maximum number of processes that can run concurrently
- Number of CPUs
- Clock rate
- Amount of virtual memory
- Maximum number of i-nodes, open files, and clists
- Amount of system swap space
- Maximum memory and open files allowed per process
- Kernel version description string

Following are sample values for each MIB object in the Kernel Configuration group:

- maxProcs(1): 138
- serialNumber(2): 11931680
- romVersion(3): 2.6
- numCpu(4): 1
- clockHZ(5): 100
- kernelVers(6): Generic_103640-31
- virtualMem(7): 68596 (KB)
- maxInode(8): 322
- maxFiles(9): 582
- maxClist(10): 228
- maxMemPerProc(11): 13852 (KB)
- totalSwap(12): 65516 (KB)
- openMaxPerProc(13): 256
- posixJobCtrl(14): true(1)
- posixVersion(15): 198808
- pageSize(16): 4096

Boot Configuration

You can use the MIB objects in the Boot Configuration group to determine which device and partition the system uses for the root file system, the dump file system, swap space, and the number of blocks contained by each.

Following are sample values for the MIB objects of the Boot Configuration group:

- rootName(1): sd0a
- rootFSType(2): 4.2
- rootBlocks(3): 0
- dumpName(4): sd0b
- dumpFSType(5): Spec
- dumpBlocks(6): 131040
- swapName(7): sd0b
- swapFSType(8): Spec
- swapSize(9): 131040

Streams Group

The Streams I/O subsystem provides a data transit/processing path between applications in user space and drivers in kernel space in the host. You can monitor the health of the Streams subsystem by retrieving objects that provide the following types of information:

- Maximum stream message size
- Number of streams in use
- Maximum number of streams
- Number of stream allocation failures
- Number of stream queues in use
- Number of stream queue allocation failures
- Statistics for stream message blocks and data blocks

Following are the MIB objects in the Streams group and sample values for each:

maxmsgSize(1)

Specifies the maximum streams message size in bytes.

Value: 4096 bytes

maxNumPush(2)

Specifies the maximum number of stream modules that can be pushed at one time.

Value: 9

numMuxLinks(3)

Specifies the number of streams multiplexor links.

Value: 87

streamUse(4)

Specifies the current number of open streams.

Value: 15

streamMaxs(5)

Specifies the greatest number of open streams recorded.

Value: 17

streamFails(6)

Specifies the number of stream allocation failures.

Value: 0

queueUse(7)

Specifies the number of streams queues currently in use.

Value: 54

queueMaxs(8)

Specifies the greatest number of open streams recorded.

Value: 62

queueFails(9)

Specifies the number of streams queue allocation failures.

Value: 0

mblockUse(10)

Specifies the number of streams message blocks in use.

Value: 26

mblockMaxs(11)

Specifies the greatest number of message blocks in use at one time.

Value: 187

mblockFails(12)

Specifies the number of streams message block allocation failures.

Value: 0

dblockUse(13)

Specifies the number of streams data blocks currently in use.

Value: 26

dblockMaxs(14)

Specifies the greatest number of data blocks in use at one time.

Value: 187

dblockFails(15)

Specifies the number of streams data block allocation failures.

Value: 0

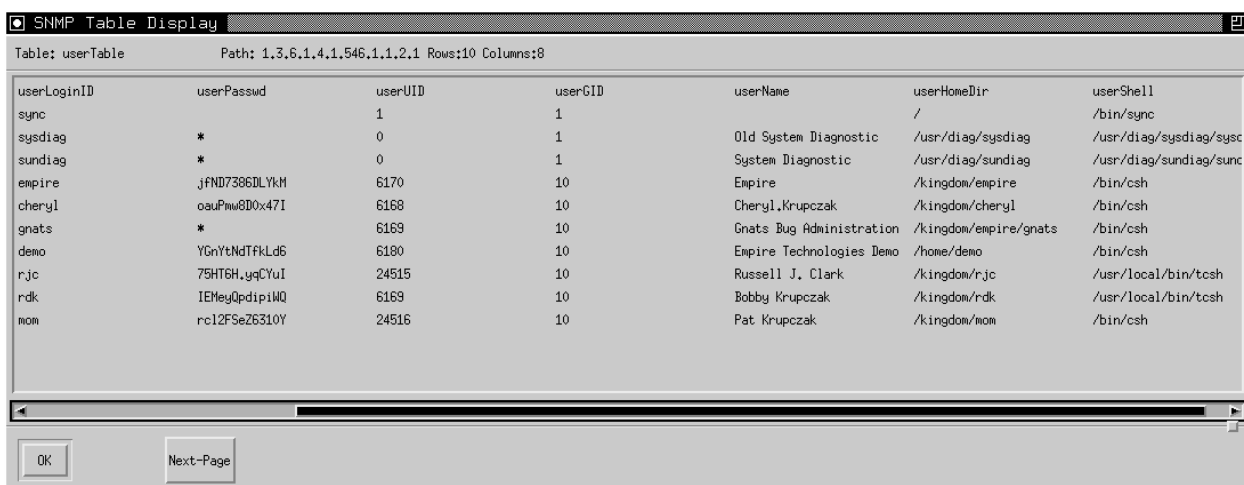
User Information

Use the User table to retrieve information about the user accounts that have been created on the system. Each row in this table represents a user account and contains columns that represent the following:

- User's login name
- Password
- User ID
- Group ID
- User name
- User home directory
- Login shell

Note: Depending on your local security policies, you may want to disable support for this table. For more information about disabling this support, see *Configuring Support for User and Group Information* in the chapter "Configuring the CA eHealth SystemEDGE Agent."

The following illustration shows the user account information in a sample User table:



The image shows a window titled "SNMP Table Display" with a table of user account information. The table has 8 columns: userLoginID, userPasswd, userUID, userGID, userName, userHomeDir, and userShell. The table contains 10 rows of data. Below the table are "OK" and "Next-Page" buttons.

| userLoginID | userPasswd | userUID | userGID | userName | userHomeDir | userShell |
|-------------|---------------|---------|---------|--------------------------|-----------------------|------------------------|
| sync | | 1 | 1 | | / | /bin/sync |
| sysdiag | * | 0 | 1 | Old System Diagnostic | /usr/diag/sysdiag | /usr/diag/sysdiag/sysc |
| sundiag | * | 0 | 1 | System Diagnostic | /usr/diag/sundiag | /usr/diag/sundiag/sunc |
| empire | jfND7386DLyK | 6170 | 10 | Empire | /kingdom/empire | /bin/csh |
| cheryl | oauPmw8D0x47I | 6168 | 10 | Cheryl Krupczak | /kingdom/cheryl | /bin/csh |
| gnats | * | 6169 | 10 | Gnats Bug Administration | /kingdom/empire/gnats | /bin/csh |
| demo | YGnYtNdTfkLd6 | 6180 | 10 | Empire Technologies Demo | /home/demo | /bin/csh |
| rjc | 75HT6H.yqCYuI | 24515 | 10 | Russell J. Clark | /kingdom/rjc | /usr/local/bin/tcsh |
| rdk | lEMegQdpip1WQ | 6169 | 10 | Bobby Krupczak | /kingdom/rdk | /usr/local/bin/tcsh |
| mom | rc12FSeZ6310Y | 24516 | 10 | Pat Krupczak | /kingdom/mom | /bin/csh |

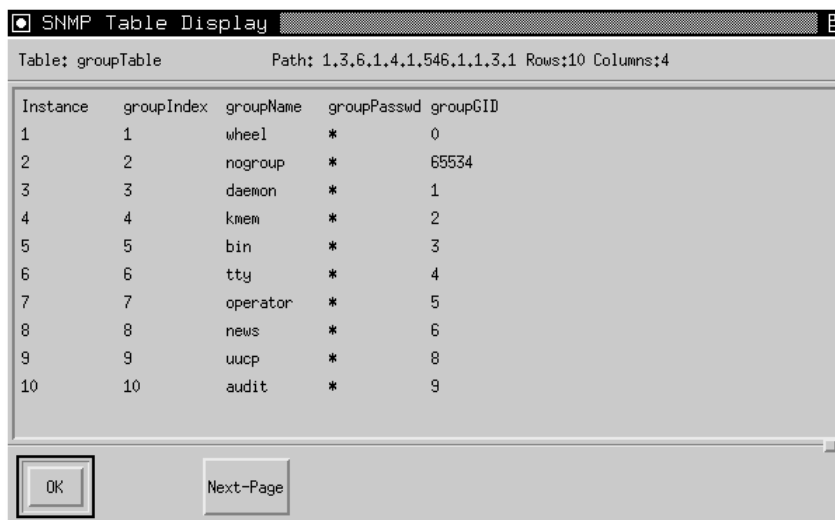
Group Information

Use the Group table to retrieve information about the user groups that have been created on the host system. Each row in the table represents a group defined in the `/etc/group` file, including the following information for each group:

- Group name
- Group password
- Group ID

Note: Depending on your local security policies, you may want to disable support for this table. For more information, see [Configuring Support for User and Group Information](#) in the chapter "Configuring the CA eHealth SystemEDGE Agent."

The following illustration shows a sample Group table:



The image shows a window titled "SNMP Table Display". Below the title bar, it says "Table: groupTable" and "Path: 1.3.6.1.4.1.546.1.1.3.1 Rows:10 Columns:4". The table has the following data:

| Instance | groupIndex | groupName | groupPasswd | groupGID |
|----------|------------|-----------|-------------|----------|
| 1 | 1 | wheel | * | 0 |
| 2 | 2 | nogroup | * | 65534 |
| 3 | 3 | daemon | * | 1 |
| 4 | 4 | knmem | * | 2 |
| 5 | 5 | bin | * | 3 |
| 6 | 6 | tty | * | 4 |
| 7 | 7 | operator | * | 5 |
| 8 | 8 | news | * | 6 |
| 9 | 9 | uucp | * | 8 |
| 10 | 10 | audit | * | 9 |

At the bottom of the window are two buttons: "OK" and "Next-Page".

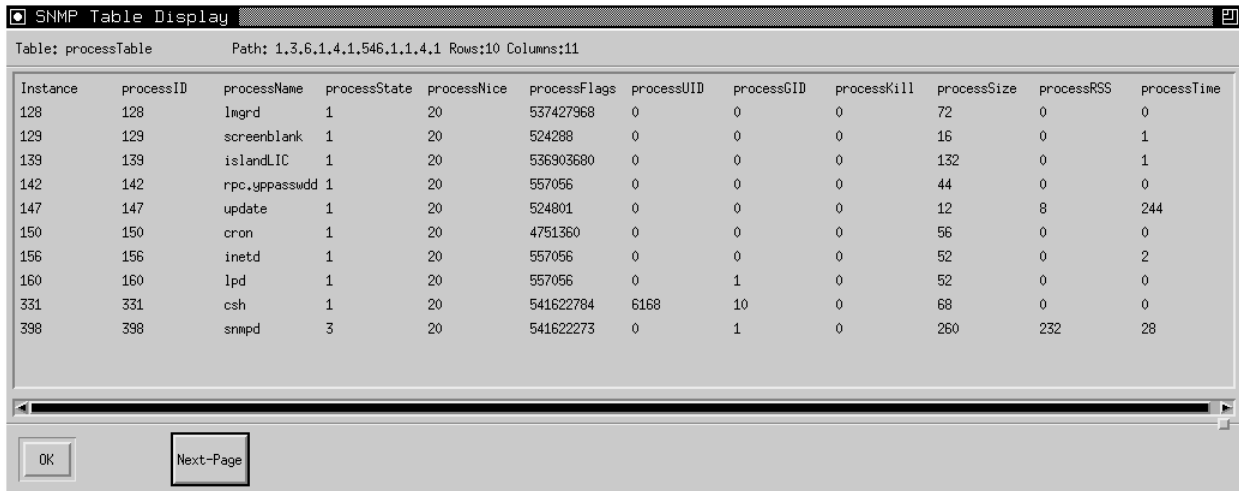
Process Information

Use the Process table to determine what processes are currently running on the system. Each row in the table represents a process and contains columns that represent the following:

- Process ID
- Name
- State
- Flags
- Owner user ID (UID)
- Owner group ID (GID)
- Scheduling priority
- Amount of memory and CPU time that the process is using

You can also control processes through the Process table. For example, you can increase or decrease a process's scheduling priority by setting the desired priority value in the process's nice column. You can also send a signal to a process by setting the kill column to the desired signal value. For example, from a remote network management station console, you can terminate an unauthorized process by setting the value of its processKill column to 9, the number that represents the UNIX SIGKILL signal, which can kill a process.

The following illustration shows a sample Process table:



SNMP Table Display

Table: processTable Path: 1.3.6.1.4.1.546.1.1.4.1 Rows:10 Columns:11

| Instance | processID | processName | processState | processNice | processFlags | processUID | processGID | processKill | processSize | processRSS | processTime |
|----------|-----------|---------------|--------------|-------------|--------------|------------|------------|-------------|-------------|------------|-------------|
| 128 | 128 | lmgd | 1 | 20 | 537427968 | 0 | 0 | 0 | 72 | 0 | 0 |
| 129 | 129 | screenblank | 1 | 20 | 524288 | 0 | 0 | 0 | 16 | 0 | 1 |
| 139 | 139 | islandLIC | 1 | 20 | 536903680 | 0 | 0 | 0 | 132 | 0 | 1 |
| 142 | 142 | rpc.yppassudd | 1 | 20 | 557056 | 0 | 0 | 0 | 44 | 0 | 0 |
| 147 | 147 | update | 1 | 20 | 524801 | 0 | 0 | 0 | 12 | 8 | 244 |
| 150 | 150 | cron | 1 | 20 | 4751360 | 0 | 0 | 0 | 56 | 0 | 0 |
| 156 | 156 | inetd | 1 | 20 | 557056 | 0 | 0 | 0 | 52 | 0 | 2 |
| 160 | 160 | lpd | 1 | 20 | 557056 | 0 | 1 | 0 | 52 | 0 | 0 |
| 331 | 331 | csd | 1 | 20 | 541622784 | 6168 | 10 | 0 | 68 | 0 | 0 |
| 398 | 398 | snmpd | 3 | 20 | 541622273 | 0 | 1 | 0 | 260 | 232 | 28 |

OK Next-Page

Change the nice Value of a Process (UNIX only)

You can change the nice value for a specific process.

To change the nice value of a process

1. Find the row that represents the selected process.
2. Set the processNice column in that row to the desired nice value (priority).
For a list of values, see the UNIX nice(1) man page.

Send a Signal to a Process

You can send a signal to a process.

When you are performing SNMP Set operations, use a community name that grants you read-write access.

To send a signal to a process

1. Find the row that represents the selected process.
2. Set the processKill column in that row to the number corresponding to the desired signal. For more information, see the UNIX signal(3V) man page.

Who Table Information

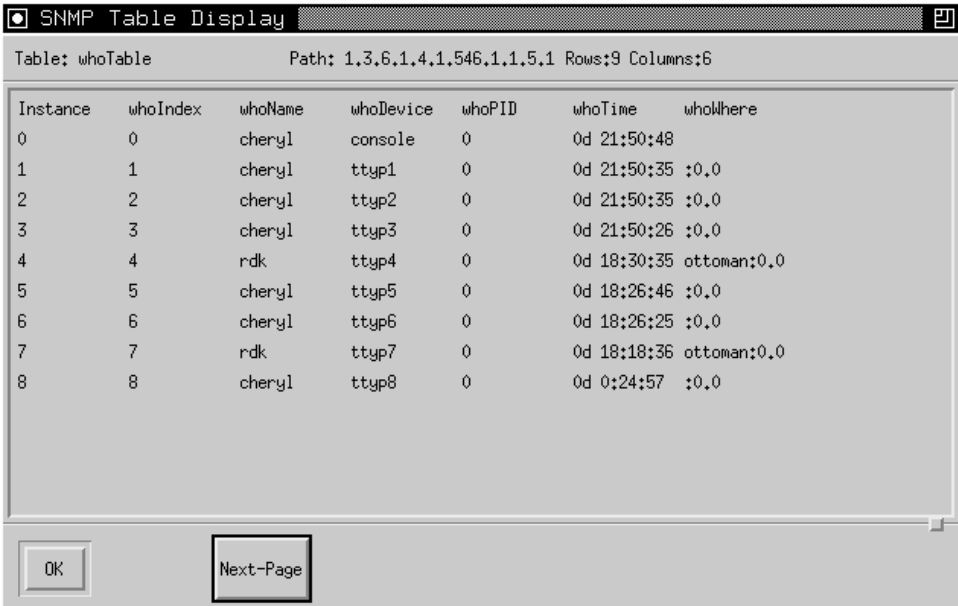
Use the Who table to find out which users are currently logged on to the host system. Each row in this table represents a current user and contains columns that represent the following:

- User's name
- Login device
- Login PID
- Login time
- Location from where the user is logged in

The Who table can help you monitor who is using this system at a particular time.

Depending on your local security policies, you may want to disable support for this table. For more information about disabling this support, see *Configuring Support for Who Information* in the chapter "Configuring the CA eHealth SystemEDGE Agent."

The following illustration shows a sample Who table, which lists the users who are currently logged on:



The image shows a window titled "SNMP Table Display" with a table titled "Table: whoTable" and "Path: 1.3.6.1.4.1.546.1.1.5.1". The table has 9 rows and 6 columns. The columns are labeled: Instance, whoIndex, whoName, whoDevice, whoPID, and whoTime. The rows show users cheryl and rdk logging in from various devices (console, tty) at different times.

| Instance | whoIndex | whoName | whoDevice | whoPID | whoTime |
|----------|----------|---------|-----------|--------|-------------------------|
| 0 | 0 | cheryl | console | 0 | 0d 21:50:48 |
| 1 | 1 | cheryl | ttyp1 | 0 | 0d 21:50:35 :0,0 |
| 2 | 2 | cheryl | ttyp2 | 0 | 0d 21:50:35 :0,0 |
| 3 | 3 | cheryl | ttyp3 | 0 | 0d 21:50:26 :0,0 |
| 4 | 4 | rdk | ttyp4 | 0 | 0d 18:30:35 ottoman;0,0 |
| 5 | 5 | cheryl | ttyp5 | 0 | 0d 18:26:46 :0,0 |
| 6 | 6 | cheryl | ttyp6 | 0 | 0d 18:26:25 :0,0 |
| 7 | 7 | rdk | ttyp7 | 0 | 0d 18:18:36 ottoman;0,0 |
| 8 | 8 | cheryl | ttyp8 | 0 | 0d 0:24:57 :0,0 |

Remote Command Execution

Use the Remote Shell Group to run shell scripts and programs on the remote host system. MIB variables in the Remote Shell group let you specify the command, its arguments, and the name of a file where the output will be written. You can specify the command and arguments by setting the remoteShell MIB variable; specify the output file by setting shellOutput.

When you perform an SNMP Set on the shellCmd MIB variable, the agent fork/execs the specified command with standard output (stdout) and standard error (stderr) redirected to the file named by the shellOutput MIB variable. When the command finishes executing, the agent puts the command's exit status in shellExitStat.

Execute a Remote Command

When you are performing SNMP Set operations, use a community name that grants you read-write access.

Enter the following at a command prompt to run a remote command:

```
SET shellOutput.0 = output filename
SET shellCmd.0 = command
```

Note: Depending on your local security policies, you may want to disable support for this table. For more information about disabling this support, see *Configuring Support for Remote Shell Capability* in the chapter “Configuring the CA eHealth SystemEDGE Agent.”

Kernel Performance Statistics

Use the Kernel Performance group to track the health and performance of the host's operating system. Statistics that you can monitor include the following:

- Number of jobs waiting on disk I/O and page I/O
- Number of jobs in the scheduler's run queue
- Number of active jobs that are swapped out, and the number that are sleeping
- Number of current processes and open files
- Statistics on paging, context switching, interrupts, and page faults

This group also includes a running percentage over 1-, 5-, and 15-minute intervals of the time that the processor was *not* in an idle state. For example, you can detect that the host system is overloaded if the 15-minute average is continuously high, and you can detect peak periods of CPU utilization by monitoring the 1-minute and 5-minute averages.

The following are the Kernel Performance Statistics MIB objects:

- cpu1Min(1)
- cpu5Min(2)
- cpu15Min(3)
- runQLen(4)
- diskWaitNum(5)
- pageWaitNum(6)
- swapActive(7)
- sleepActive(8)
- memInUse(9)
- activeMem(10)
- numProcs(11)
- numOpenFiles(12)
- swapInUse(13)
- numSwitches(14)
- numTraps(15)
- numSyscalls(16)
- numInterrupts(17)
- numPageSwapIn(18)
- numPageSwapOut(19)
- numSwapIn(20)
- numSwapOut(21)
- numPageIn(22)
- numPageOut(23)
- numPageReclaims(24)
- numPageFaults(25)
- loadAverage1Min(26)
- loadAverage5Min(27)
- loadAverage15Min(28)

- totalSwapSpace(29)
- swapCapacity(30)
- memCapacity(31)
- memInUseCapacity(32)
- pageScans(33)

For descriptions of these MIB objects, see the `empire.asn1` file in the `/doc` subdirectory of the CA eHealth SystemEDGE agent distribution.

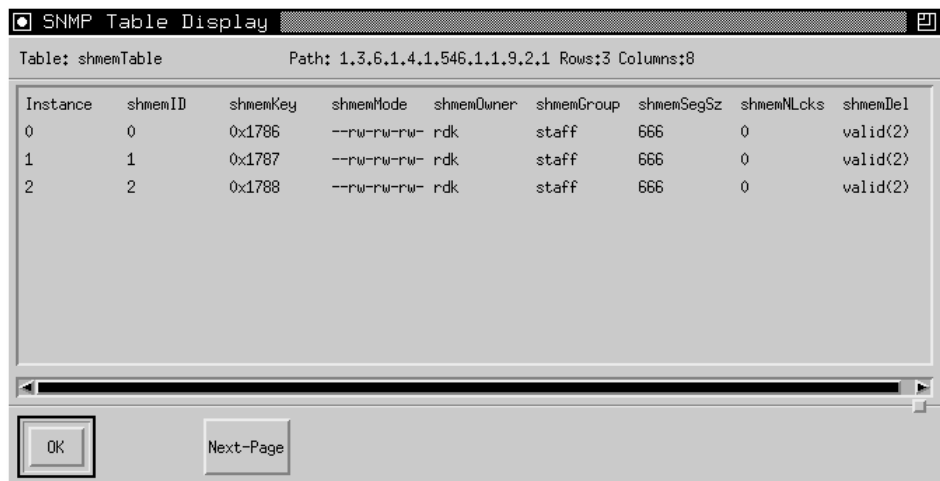
Interprocess Communication: Queues, Shared Memory, and Semaphores

You can track the IPC mechanisms in use on the system by retrieving the appropriate MIB table:

- Message Queue table
- Shared Memory table
- Semaphore table

Note: A semaphore is a value in operating system or kernel storage that a process can check and change to coordinate activities in which multiple process compete for the same operating system resources.

A row in each of these tables represents one instance of IPC usage. The columns differ for each table. The columns of the Message Queue table provide the message queue ID, key, mode, owner, group, and size. The columns of the Shared Memory table provide the shared memory segment ID, key, mode, owner, group, and size. The columns of the Semaphore table provide the semaphore ID, key, mode, and owner. The following illustration shows a sample Shared Memory table.



The image shows a window titled "SNMP Table Display". Inside, it says "Table: shmemTable" and "Path: 1.3.6.1.4.1.546.1.1.9.2.1 Rows:3 Columns:8". Below this is a table with 8 columns: Instance, shmemID, shmemKey, shmemMode, shmemOwner, shmemGroup, shmemSegSz, shmemNLcks, and shmemDel. There are 3 rows of data. At the bottom of the window are "OK" and "Next-Page" buttons.

| Instance | shmemID | shmemKey | shmemMode | shmemOwner | shmemGroup | shmemSegSz | shmemNLcks | shmemDel |
|----------|---------|----------|-------------|------------|------------|------------|------------|----------|
| 0 | 0 | 0x1786 | --rw-rw-rw- | rdk | staff | 666 | 0 | valid(2) |
| 1 | 1 | 0x1787 | --rw-rw-rw- | rdk | staff | 666 | 0 | valid(2) |
| 2 | 2 | 0x1788 | --rw-rw-rw- | rdk | staff | 666 | 0 | valid(2) |

Deleting an Interprocess Communication

When you are performing SNMP Set operations, use a community name that grants you read-write access.

The IPC tables provide you with the power to delete message queues, shared memory segments, and semaphores using an SNMP Set operation. Each of these tables contains a column that acts as a Delete button; setting the column to the value of delete(1) causes the agent to destroy that instance of IPC usage.

The Delete button objects are named queDel (for the Message Queue table), shmemDel (for the Shared Memory table), and semDel (for the Semaphore table).

Message Buffer Allocation and Usage Statistics

Use the Message Buffer Allocation table to discover how your system is using message buffers (mbufs) and how many buffers have been allocated for each use. In addition, you can obtain statistics for the number of times mbuf requests were denied or delayed, which can help you track down message buffer shortages. Following are the MIB objects of the Message Buffer Group and sample values for each object:

numMbufs(1)

Specifies the total number of message buffers in the message buffer pool.

Value: 608

numClusters(2)

Specifies the total number of logical pages or clusters obtained from the page pool.

Value: 28

freeClusters(2)

Specifies the number of free clusters.

Value: 28

numDrops(4)

Specifies the number of times requests for message buffers were denied.

Value: 0

numWait(5)

Specifies the number of times requests for message buffers were delayed.

Value: 0

numDrain(6)

Specifies the number of calls to protocol drain routes.

Value: 0

The following illustration shows a sample Message Buffer Allocation table:

SNMP Table Display

Table: mbufAllocTable Path: 1.3.6.1.4.1.546.1.1.10.1.1 Rows:10 Columns:3

| Instance | mbufType | mbufDesc | mbufAlloc |
|----------|----------|---------------------------|-----------|
| 0 | 0 | Free list | 81 |
| 1 | 1 | Dynamic (data) allocation | 1 |
| 2 | 2 | Packet header | 6 |
| 3 | 3 | Socket structure | 165 |
| 4 | 4 | Protocol control block | 203 |
| 5 | 5 | Routing tables | 3 |
| 6 | 6 | IMP host tables | 0 |
| 7 | 7 | Address resolution tables | 0 |
| 8 | 8 | Socket name | 18 |
| 9 | 9 | Zombie proc status | 0 |

OK Next-Page

Stream Buffers

Use the Streams Buffer Allocation table to monitor buffer allocation and usage statistics for buffers used by the Streams subsystem. The following illustration shows a sample Streams Buffer Allocation table:

SNMP Table Display

Table: strbufAllocTable Path: 1.3.6.1.4.1.546.1.1.10.2.1.1 Rows:8 Columns:6

| Instance | strbufAllocIndex | strbufAllocSize | strbufAllocCurrent | strbufAllocMax | strbufAllocTotal | strbufAllocFail |
|----------|------------------|-----------------|--------------------|----------------|------------------|-----------------|
| 0 | 0 | 16 | 0 | 0 | 0 | 0 |
| 1 | 1 | 32 | 0 | 16 | 294714 | 0 |
| 2 | 2 | 128 | 5 | 30 | 117340 | 0 |
| 3 | 3 | 512 | 0 | 4 | 18373 | 0 |
| 4 | 4 | 1024 | 22 | 163 | 120473 | 0 |
| 5 | 5 | 2048 | 0 | 4 | 1665 | 0 |
| 6 | 6 | 8192 | 0 | 1 | 483 | 0 |
| 7 | 7 | 16777215 | 0 | 4 | 8 | 0 |

OK Next-Page

I/O Buffer Cache

Use the I/O Buffer Cache group to track I/O buffer allocation and usage for basic disk I/O. You can also graph the values of these counters to detect peak periods of I/O buffer activity. Following are the I/O Buffer Cache group MIB objects and sample values for each object:

numBreadRequests(1)

Specifies the total number of buffer read calls that were made.

Value: 1121570

numBreadHits(2)

Specifies the number of kernel buffer cache hits.

Value: 1047985

numBufSleeps(3)

Specifies the total number of times a kernel had to sleep for a buffer.

Value: 0

numAgeAlloc(4)

Specifies the total number of times an aged buffer was allocated.

Value: 15423

numLRUAlloc(5)

Specifies the total number of times an LRU buffer was allocated.

Value: 88217

minNumBufHdrs(6)

Specifies the minimum number of buffer headers allocated.

Value: 30

numAllocBuf(7)

Specifies the current number of allocated buffers.

Value: 30

ioBufferHitRate(8)

Specifies the percentage of buffer read requests that result in a 'hit'.

Value: 90

RPC Group

Use the Remote Procedure Call (RPC) Group to track statistics and counters that relate to the use of the kernel's RPC facilities. You can graph the values of these counters to detect peak periods of RPC activity. Statistics and counters are divided according to their applicability towards client and server side protocol operations. For more information about RPC, see RFC 1057. The following are the RPC group MIB objects:

- clientRPCCalls(1)
- clientRPCBadcalls(2)
- clientRPCRetrans(3)
- clientRPCBadxids(4)
- clientRPCTimeouts(5)
- clientRPCWaits(6)
- clientRPCNewcreds(7)
- clientRPCTimers(8)
- serverRPCCalls(9)
- serverRPCBadcalls(10)
- serverRPCNullrecvs(11)
- serverRPCBadlens(12)
- serverRPCXdrcalls(13)

For descriptions of these MIB objects, see the empire.asn1 file in the /doc subdirectory of the CA eHealth SystemEDGE agent distribution.

NFS Group

Use the NFS group to track statistics and counters related to the use of the kernel's NFS facilities. You can graph the values of these counters to detect peak periods of NFS activity. Statistics and counters are divided according to their applicability towards client and server side protocol operations. For more information about NFS, see RFC 1094. The following are the NFS group MIB objects:

- clientNFSCalls(1)
- clientNFSBadcalls(2)
- clientNFSNclgets(3)
- clientNFSNclsleeps(4)
- clientNFSNulls(5)

- clientNFSGetattrs(6)
- clientNFSSetattrs(7)
- clientNFSRoots(8)
- clientNFSLookups(9)
- clientNFSReadlinks(10)
- clientNFSReads(11)
- clientNFSWrcaches(12)
- clientNFSWrites(13)
- clientNFSCreates(14)
- clientNFSRemoves(15)
- clientNFSRenames(16)
- clientNFSLinks(17)
- clientNFSSymlinks(18)
- clientNFSMkdirs(19)
- clientNFSRmdirs(20)
- clientNFSReaddirs(21)
- clientNFSFsstats(22)
- serverNFSCalls(23)
- serverNFSBadcalls(24)
- serverNFSNulls(25)
- serverNFSGetattrs(26)
- serverNFSSetattrs(27)
- serverNFSRoots(28)
- serverNFSLookups(29)
- serverNFSReadlinks(30)
- serverNFSReads(31)
- serverNFSWrcaches(32)
- serverNFSWrites(33)
- serverNFSCreates(34)
- serverNFSRemoves(35)
- serverNFSRenames(36)
- serverNFSLinks(37)
- serverNFSSymlinks(38)

- serverNFSMkdirs(39)
- serverNFSRmdir(40)
- serverNFSReaddir(41)
- serverNFSFsstats(42)

For descriptions of these MIB objects, see the `empire.asn1` file in the `/doc` subdirectory of the CA eHealth SystemEDGE agent distribution.

Windows-Specific Groups

In most cases, the CA eHealth SystemEDGE agent supports the same MIB objects for Windows and UNIX. However, the underlying Windows operating system does not support some of the MIB objects, and therefore cannot be implemented by the CA eHealth SystemEDGE agent for Windows. For a list of the MIB objects not supported by these Windows operating systems, see [Unsupported MIB Objects on Windows](#) in this chapter. The following sections define groups that have been specifically designed for Windows systems.

NT System Group

Use the NT System group to determine the following:

- Operating system version, build, and service-pack numbers
- Kernel configuration parameters
- Cluster information
- Other pertinent system information about the Windows system

Following are the NT System group MIB objects:

- ntSystemVersion(1)
- ntBuildNumber(2)
- ntServicePackNumber(3)
- ntWorkstationOrServer(4)
- ntfsDisable8dot3NameCreation(5)
- ntWin31FileSystem(6)
- ntCriticalSectTimeout(7)
- ntGlobalFlag(8)
- ntIoPageLockLimit(9)
- ntLargeSystemCache(10)

- ntPagedPoolSize(11)
- ntNonPagedPoolSize(12)
- ntPagingFiles(13)
- ntSystemPages(14)
- ntOptionalSubsystem(15)
- ntCmdlineOptions(16)
- ntLPTTimeout(17)
- ntDosMemSize(18)
- ntWowCmdline(19)
- ntWowSize(20)
- ntUserFullScreen(21)
- ntHistoryBufferSize(22)
- ntNumberHistoryBuffers(23)
- ntQuickEdit(24)
- ntScreenBufferSize(25)
- ntWindowSize(26)
- ntWindowsAppInitDLLs(27)
- ntWindowsDeviceNotSelectedTimeout(28)
- ntWindowsSpooler(29)
- ntWindowsSwapDisk(30)
- ntWindowsXmitRetryTimeout(31)
- ntSystemRoot(32)
- ntBuildType(33)
- ntSysStartOptions(34)
- ntSysBiosDate(35)
- ntSysBiosVersion(36)
- ntVideoResolution(37)
- ntCrashDumpEnabled(38)
- ntLogEvent(39)
- ntOverwrite(40)
- ntSendAlert(41)
- ntIsClustered(42)
- ntClusterName(43)

- ntClusterMembers(44)
- ntClusterIsActive(45)
- ntClusterActiveNode(46)

For descriptions of these MIB objects, see the empire.asn1 file in the /doc subdirectory of the CA eHealth SystemEDGE agent distribution.

NT Thread Group

In the Windows operating system, the kernel schedules threads to run on the processors. A thread is a part of a process that runs program code. A process may contain one or many threads.

Use the NT Thread table to obtain detailed status information for each thread executing on the system, including the following:

- Process to which the thread belongs
- Number of seconds the thread has been running
- Percentage of time that the thread has run in user and privileged modes
- Current state of the thread (ready, running, wait, terminated, and so on)
- Reason a thread is waiting if it is in a wait state

Following are the NT Thread group MIB objects:

ntThreadPID

Specifies the process to which the thread belongs.

ntThreadNumber

Specifies the number for the thread within the process.

ntThreadPrivTime

Specifies the total elapsed time (in centiseconds) that this thread has executed in privileged mode.

ntThreadProcTime

Specifies the total elapsed time (in centiseconds) that the thread has used the processor to execute instructions.

ntThreadUserTime

Specifies the total elapsed time (in centiseconds) that the thread has executed in user mode.

ntThreadContextSwitches

Specifies the number of context switches.

ntThreadElapsedTime

Specifies the total elapsed time (in seconds) that the thread has been running.

ntThreadPriorityBase

Specifies the base priority level of the thread.

ntThreadPriority

Specifies the current priority level of the thread.

ntThreadWaitReason

Specifies the reason that the thread is waiting: executive, free page, page in, virtual memory, user request, and so on.

ntThreadStartAddr

Specifies the starting virtual address.

ntThreadState

Specifies the current state: ready, running, wait, terminated, and so on.

ntThreadID

Specifies the system-wide unique ID number for the thread.

NT Registry Group

The NT Registry group enables you to query the Windows registry. You can determine if your registry is in danger of growing larger than the size limit set for it by querying the following objects:

ntRegistryCurrentSize

Provides the current size of the registry in MB.

ntRegistrySizeLimit

Provides the size limit (in MB) defined for the registry.

Note: The Windows registry is a database repository for information about the system's configuration for hardware, user accounts, and applications.

NT Service Group

Windows provides several programs that run as Windows services. The NT Service group provides detailed status information for each configured service on the system including the following:

- Service name
- Path to service executable
- Start type
- Parameters
- State
- Object name

Following are the NT Service group MIB objects:

ntServiceIndex

Specifies the index of this service entry.

ntServiceName

Specifies the name of the Windows service.

ntServicePathName

Specifies the path name of the executables.

ntServiceStartType

Specifies the service start type (for example, Automatic).

ntServiceParameters

Specifies the start parameters passed to this service when it is invoked.

ntServiceState

Specifies the current state of the service.

ntServiceObjectName

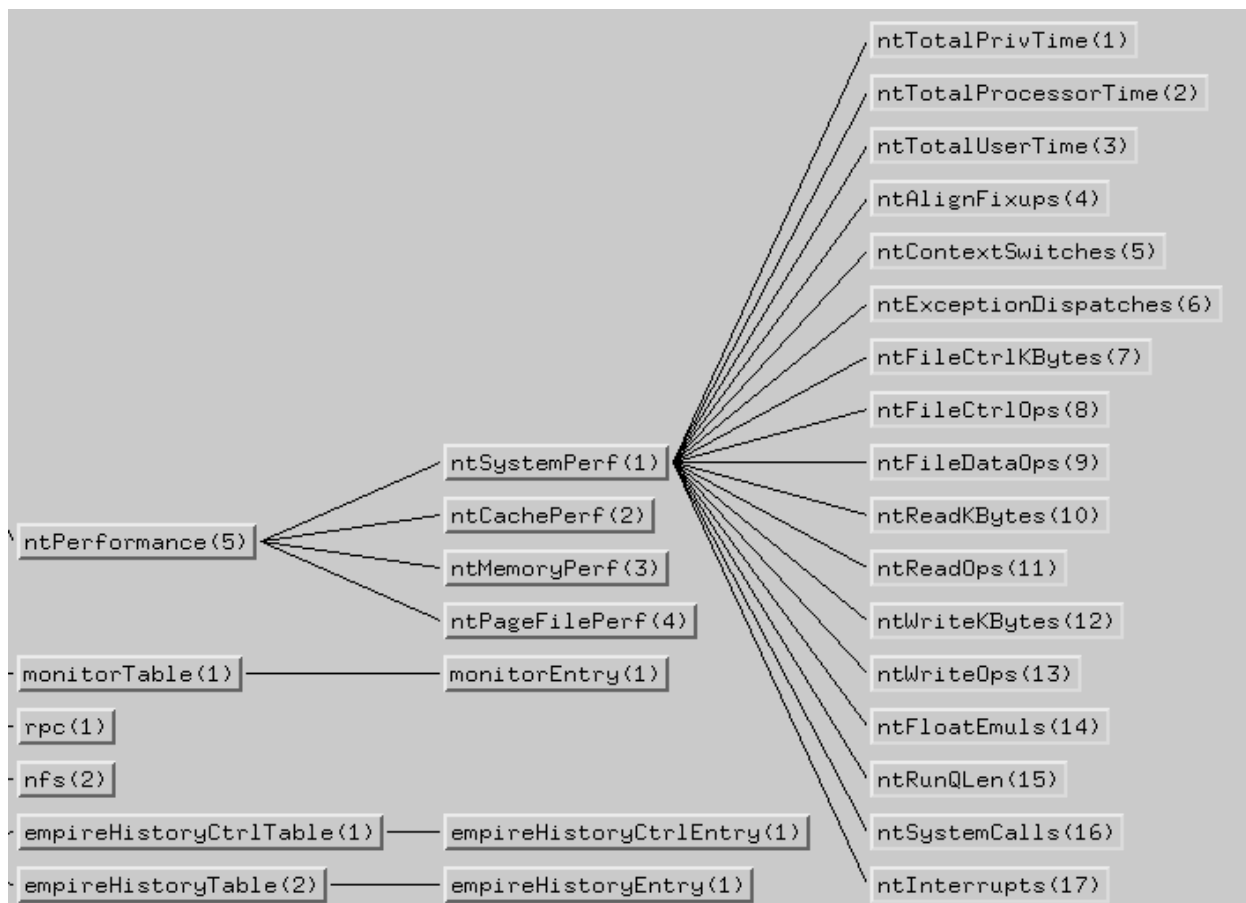
Specifies the current user name that the service is running as.

NT System Performance Group

The NT System Performance group provides statistics that enable you to track the health and performance of the system's Windows operating system. Specifically, performance counters in the System Performance group let you see the following:

- How much time the processors spent in user and privileged modes
- Number of context switches that have taken place
- Number of system calls and interrupts
- Number of jobs in the scheduler's run queue
- Performance counters for file system operations, such as the number of read, write, and control operations, and the total number of KB for each

The following illustration shows the organization of the NT System Performance group:



Detecting a Heavily Loaded System

A heavily loaded system typically has a high level of system activity with many process threads competing for CPU time and jobs queuing up as they wait to run. You can track the level of system activity by monitoring the `ntContextSwitches` and `ntRunQLen` MIB objects.

`ntContextSwitches` counts the number of context switches that have occurred. A context switch occurs each time a thread gives up the CPU and another takes its place. A high rate of context switching indicates a high system load. `ntRunQLen` indicates whether there is a backup of threads waiting to run.

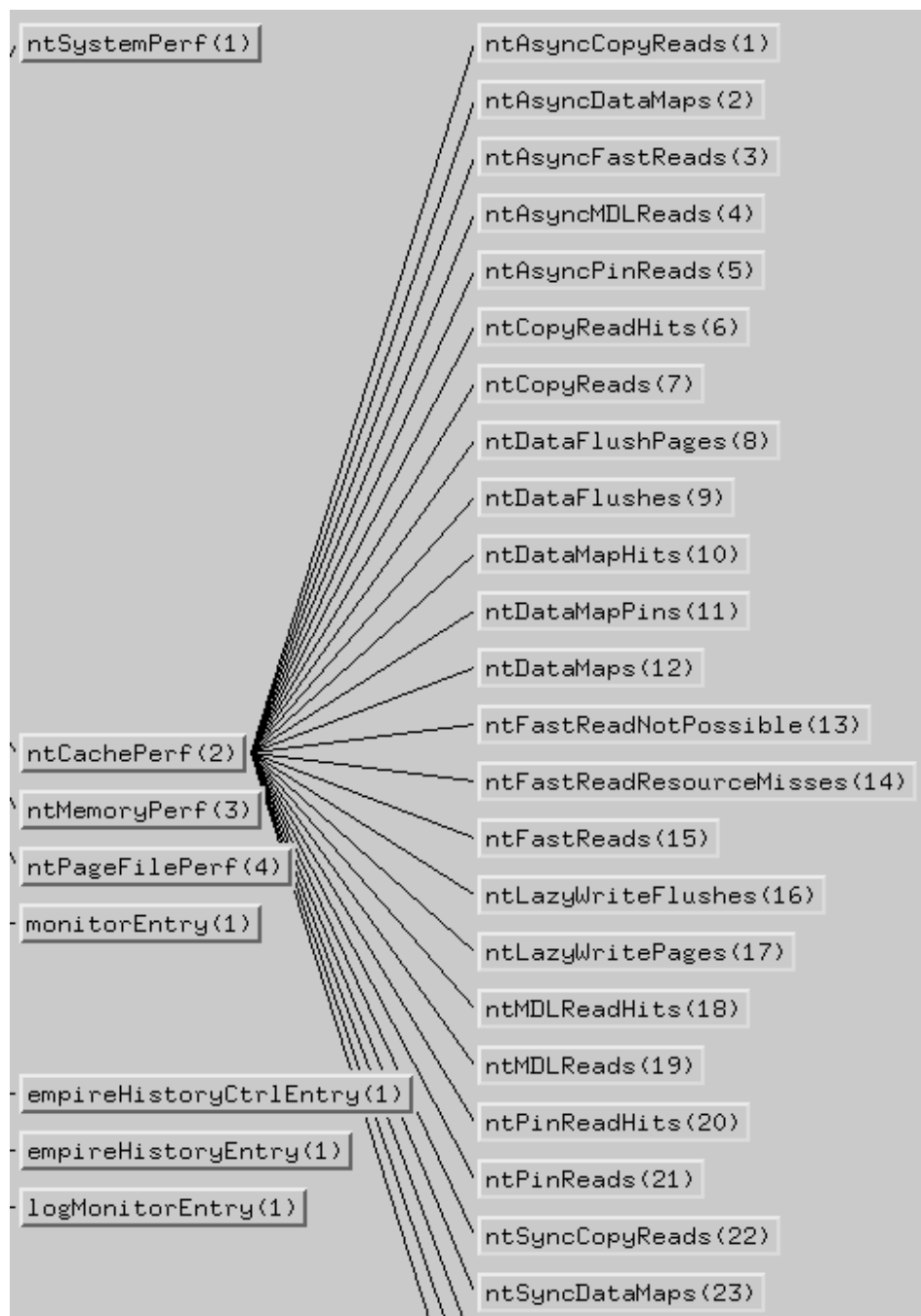
NT Cache Performance Group

The NT Cache Performance group contains system-wide buffer and filesystem cache statistics that let you determine the effectiveness of a system's caching mechanisms.

When an application requests data, the data is first mapped into the cache and then copied into memory from the cache. Later, data changed by the application is written from the cache to disk by the Lazy Writer system thread or by a write-through call from the application.

Monitor the cache performance objects in this group to see if the cache is performing poorly (for example, if the system has a low cache hit ratio). If so, your system may need additional memory.

The following illustration shows the organization of the NT Cache Performance group:



NT Memory Performance Group

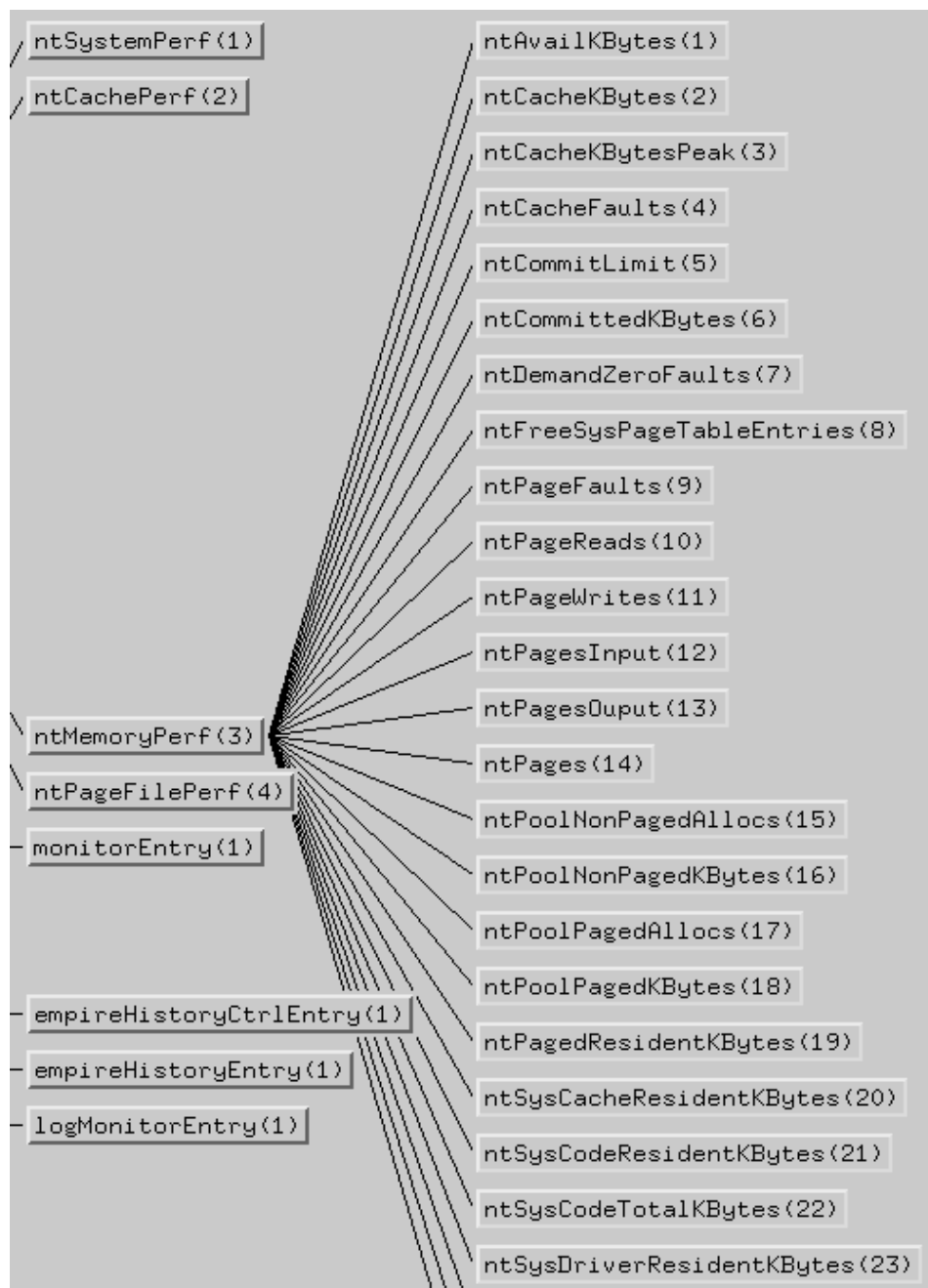
The NT Memory Performance group contains memory and paging performance counters and statistics that enable you to track the memory usage of your system.

You can determine, for example, the following:

- Amount of available virtual memory
- Number of KB being used by the system cache
- Number of page faults, page reads, and page writes
- Many other memory and paging-related statistics

For more information about the specific MIB objects, see the `empire.asn1` file in the 'doc' subdirectory of the CA eHealth SystemEDGE agent distribution.

The following illustration shows the organization of the NT Memory Performance Group:



NT Page File Performance Group

Windows uses the Pagefile.sys paging file to handle any extra demands for memory. The paging file is a block of disk space reserved by the operating system so that the memory manager can free space in memory when necessary. It does so by writing pages not often referenced to the paging file on disk. You can monitor the size of the paging file to determine whether you should add more memory to your system.

The NT Page File Performance group includes the following MIB objects:

ntPageFileUsage

Provides the percentage of the page file space currently being used.

ntPageFilePeakUsage

Provides the maximum percentage of the page files that were used when memory demand was at its peak.

NT Event Monitor Group

Windows uses event logs to capture important system and application status messages. You can use the CA eHealth SystemEDGE agent Windows event monitoring capability to instruct the agent to continuously monitor Windows event logs for specific events that you specify. Whenever a matching event is generated on the system, the agent notifies the management systems with a trap message. The agent can also run an action command to immediately handle the event.

The NT Event table provides detailed status information for each monitor entry. For more information about using Windows event monitoring, see the chapter "Configuring Windows Event Monitoring."

Following are the NT Event Monitor group MIB objects:

ntEventMonIndex

Specifies the row index.

ntEventMonLog

Specifies the number identifying the type of event log to monitor.

ntEventMonTime

Specifies the time of the last matching event.

ntEventMonMatches

Specifies the number of matches so far.

ntEventMonTypeLastMatch

Specifies the event type of the last matching event.

ntEventMonTypeFilter

Specifies the number identifying the event types to match.

ntEventMonSrcLastMatch

Specifies the source name of the event log that last matched this entry.

ntEventMonSrcFilter

Specifies the regular expression to match the source name in the event log.

ntEventLastMatch

Specifies the event description of the last match entry.

ntEventMonDescFilter

Specifies the regular expression to match the description in the event log.

ntEventMonStatus

Specifies the row status.

ntEventMonDescr

Specifies a description of the event.

ntEventMonAction

Specifies the action to perform if a match is found.

ntEventMonFlags

Specifies the flags that indicate additional behavior.

NT Registry and Performance Extension Group

The CA eHealth SystemEDGE agent provides a powerful mechanism for extending the Systems Management MIB to include information from the Windows registry and performance counters. This includes both configuration data (typically viewed using regedit) and performance data (typically viewed using perfmon).

Using this feature, you can customize CA eHealth SystemEDGE to return additional configuration and performance information for systems and applications. For instance, many applications provide registry entries that specify the configuration of the application. The CA eHealth SystemEDGE agent can make these entries available through SNMP using the CA eHealth SystemEDGE agent. In addition, the CA eHealth SystemEDGE agent can also access statistics that many applications provide.

This support is provided in the ntRegPerfGroup of the Systems Management MIB. This group contains 128 unspecified scalar MIB variables that you can configure. In response to a SNMP Get request for one of these variables, the CA eHealth SystemEDGE agent reads the Windows Registry and returns the value. For more information about using and configuring MIB objects in the NT Registry Performance group, see the chapter “Adding Windows Registry and Performance MIB Objects.”

Unsupported MIB Objects on Windows

The CA eHealth SystemEDGE agent for Windows supports the Systems Management MIB, but some of the MIB objects within this MIB module are not supported by the underlying Windows operating system. In addition, some of the UNIX-specific MIB objects are not applicable to Windows. Those objects, therefore, cannot be implemented by the Windows version of the agent.

The following Systems Management MIB objects are not supported by Windows:

- system.hostid
- devTable.devTfiles
- devTable.devFfiles
- devTable.devMaxNameLen
- devTable.devFstr
- devTable.devInodeCapacity
- kernelConfig.serialNumber
- kernelConfig.clockHZ
- kernelConfig.maxInode
- kernelConfig.maxFiles
- kernelConfig.maxClist
- kernelConfig.maxMemPerProc
- kernelConfig.openMaxPerProc
- kernelConfig.posixJobCtrl
- kernelConfig.posixVersion
- bootconf
- streams
- userTable.userUID
- userTable.userGID
- userTable.userShell

- processTable.processFlags
- processTable.processUID
- processTable.processGID
- processTable.processInBlks
- processTable.processOutBlks
- processTable.processMsgsSent
- processTable.processMsgsRecv
- processTable.processSysCalls
- processTable.processMinorPgFaults
- processTable.processNumSwaps
- processTable.processVolCtx
- processTable.process.InvolCtx
- performance (except kernelperf)
- kernelperf.diskWaitNum
- kernelperf.pageWaitNum
- kernelperf.swapActive
- kernelperf.sleepActive
- kernelperf.numTraps
- kernelperf.numPageSwapIns
- kernelperf.numPageSwapOuts
- kernelper.numSwapIns
- kernelper.numSwapOuts
- kernelperf.numPageReclaims
- kernelperf.pageScans
- errorTable
- ipc
- buffers.mbuf
- buffers.strbuf
- ioBufferCache.numBufSleeps
- ioBufferCache.numAgeAllocs
- ioBufferCache.numLRUAllocs
- ioBufferCache.numBufHdrs
- ioBufferCache.numAllocBuff

- dnld
- ntRegistry.ntRegistryCurrentSize
- rpc
- nfs

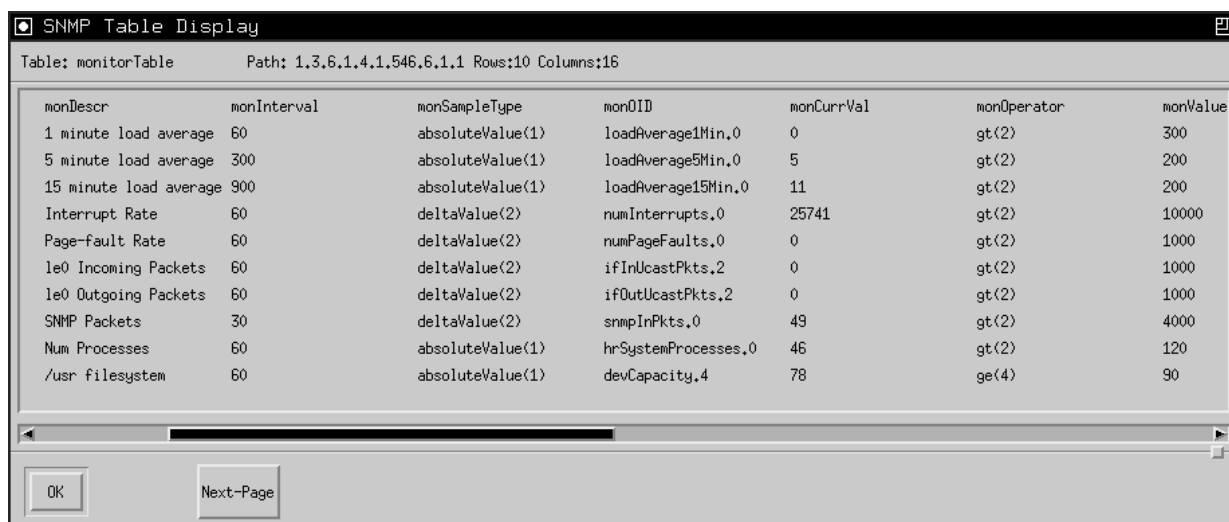
Monitor Table

The Monitor table enables you to dynamically configure the agent's self-monitoring capability to monitor any integer-based MIB variable under its control; you select the polling interval, comparison operator (greater than, less than, equal to, and so on), and threshold value. The agent automatically monitors the MIB variable for you and notifies the management system with a trap message if an exception occurs.

Use the Monitor table to specify an entry for any MIB variable within MIB-II, the Host Resources MIB, and the Systems Management MIB. As the agent is running, the entries in the Monitor table also show information such as the following:

- Time at which the variable specified by the entry was last sampled
- Value of the variable
- Lowest and highest values observed
- Number of times that a trap has been sent for the entry

The following illustration shows examples of some conditions that you can configure the agent to monitor:



SNMP Table Display

Table: monitorTable Path: 1.3.6.1.4.1.546.6.1.1 Rows:10 Columns:16

| monDescr | monInterval | monSampleType | monOID | monCurrVal | monOperator | monValue |
|------------------------|-------------|------------------|---------------------|------------|-------------|----------|
| 1 minute load average | 60 | absoluteValue(1) | loadAverage1Min.0 | 0 | gt(2) | 300 |
| 5 minute load average | 300 | absoluteValue(1) | loadAverage5Min.0 | 5 | gt(2) | 200 |
| 15 minute load average | 900 | absoluteValue(1) | loadAverage15Min.0 | 11 | gt(2) | 200 |
| Interrupt Rate | 60 | deltaValue(2) | numInterrupts.0 | 25741 | gt(2) | 10000 |
| Page-fault Rate | 60 | deltaValue(2) | numPageFaults.0 | 0 | gt(2) | 1000 |
| le0 Incoming Packets | 60 | deltaValue(2) | ifInUcastPkts.2 | 0 | gt(2) | 1000 |
| le0 Outgoing Packets | 60 | deltaValue(2) | ifOutUcastPkts.2 | 0 | gt(2) | 1000 |
| SNMP Packets | 30 | deltaValue(2) | snmpInPkts.0 | 49 | gt(2) | 4000 |
| Num Processes | 60 | absoluteValue(1) | hrSystemProcesses.0 | 46 | gt(2) | 120 |
| /usr filesystem | 60 | absoluteValue(1) | devCapacity.4 | 78 | ge(4) | 90 |

OK Next-Page

For more information about using the Monitor table and CA eHealth SystemEDGE threshold monitoring, see the chapter “Configuring Threshold Monitoring.”

Process Monitor Table

The Process Monitor table provides a powerful and flexible mechanism for monitoring applications and processes. You can monitor various attributes of key processes and applications and send SNMP traps when certain thresholds have been exceeded. The following illustration shows a sample Process Monitor table:

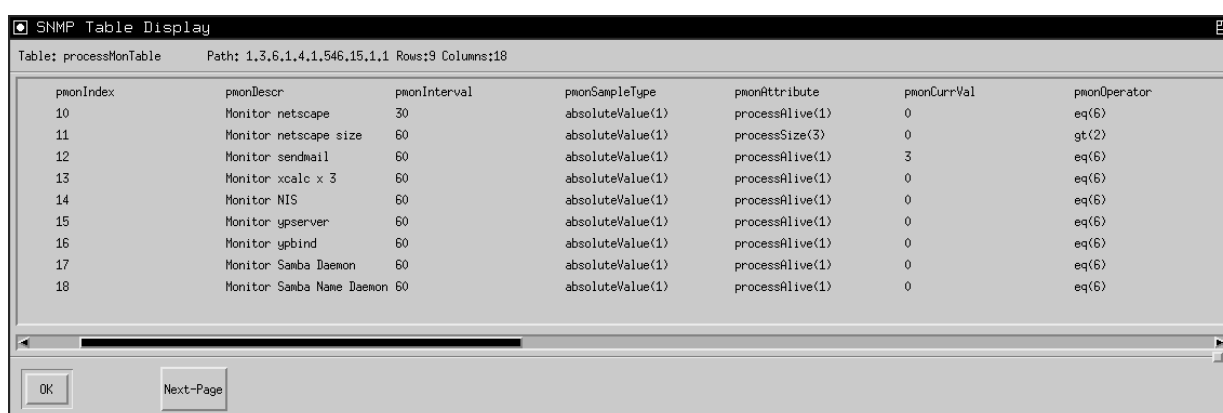


Table: processMonTable Path: 1.3.6.1.4.1.546.15.1.1 Rows:9 Columns:18

| pmIndex | pmDescr | pmInterval | pmSampleType | pmAttribute | pmCurrVal | pmOperator |
|---------|---------------------------|------------|------------------|-----------------|-----------|------------|
| 10 | Monitor netscape | 30 | absoluteValue(1) | processAlive(1) | 0 | eq(6) |
| 11 | Monitor netscape size | 60 | absoluteValue(1) | processSize(3) | 0 | gt(2) |
| 12 | Monitor sendmail | 60 | absoluteValue(1) | processAlive(1) | 3 | eq(6) |
| 13 | Monitor xcalc x 3 | 60 | absoluteValue(1) | processAlive(1) | 0 | eq(6) |
| 14 | Monitor NIS | 60 | absoluteValue(1) | processAlive(1) | 0 | eq(6) |
| 15 | Monitor ypserver | 60 | absoluteValue(1) | processAlive(1) | 0 | eq(6) |
| 16 | Monitor ypbind | 60 | absoluteValue(1) | processAlive(1) | 0 | eq(6) |
| 17 | Monitor Samba Daemon | 60 | absoluteValue(1) | processAlive(1) | 0 | eq(6) |
| 18 | Monitor Samba Name Daemon | 60 | absoluteValue(1) | processAlive(1) | 0 | eq(6) |

For more information about using the Process Monitor table, see the chapter “Configuring Process and Service Monitoring.”

Process Group Monitor Table

The Process Group Monitor table provides a powerful and flexible mechanism for monitoring groups of processes. The following illustration shows a sample Process Group Monitor table.

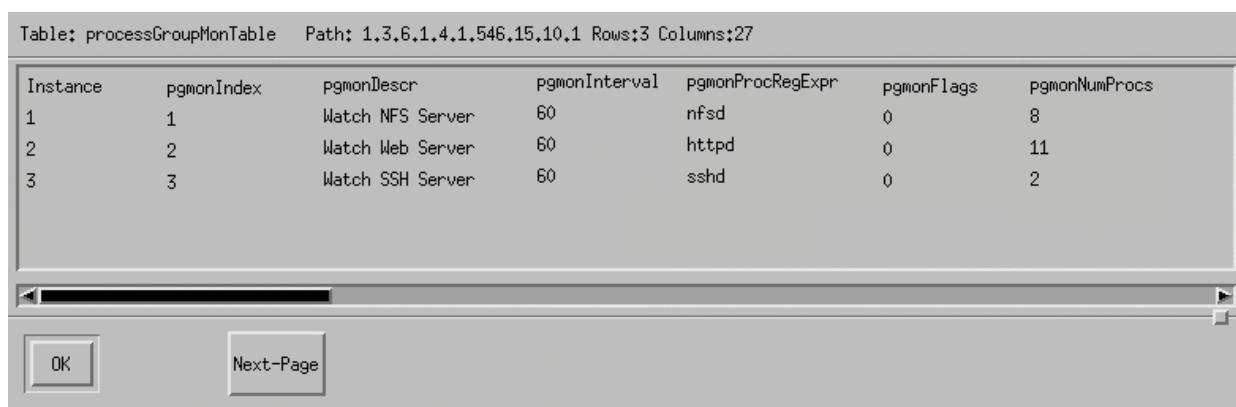


Table: processGroupMonTable Path: 1.3.6.1.4.1.546.15.10.1 Rows:3 Columns:27

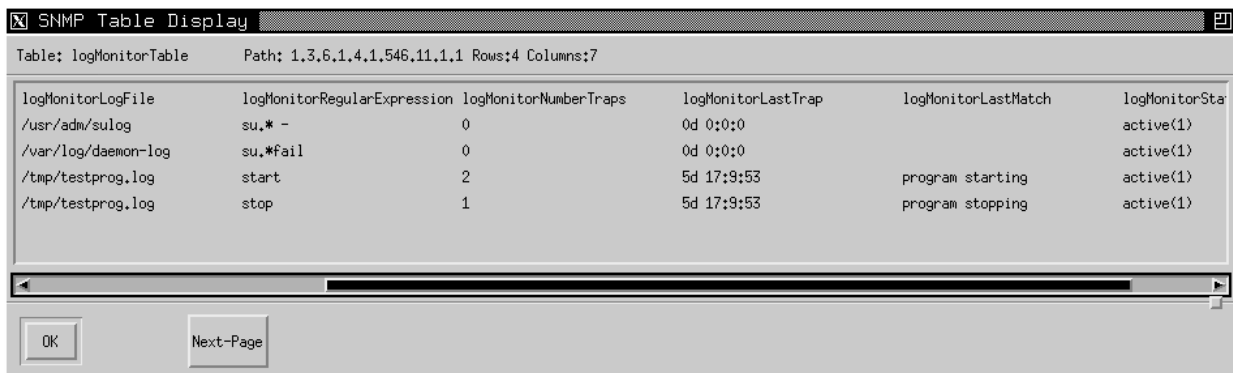
| Instance | pgmonIndex | pgmonDescr | pgmonInterval | pgmonProcRegExpr | pgmonFlags | pgmonNumProcs |
|----------|------------|------------------|---------------|------------------|------------|---------------|
| 1 | 1 | Watch NFS Server | 60 | nfsd | 0 | 8 |
| 2 | 2 | Watch Web Server | 60 | httpd | 0 | 11 |
| 3 | 3 | Watch SSH Server | 60 | sshd | 0 | 2 |

For more information about using the Process Group Monitor table, see the chapter “Configuring Process Group Monitoring.”

Log Monitor Table

The Log Monitor table enables you to dynamically configure the agent to monitor system log files for regular expressions. For example, you can configure the agent to monitor the `/var/adm/sulog` log file for the regular expression `su.*fail`. Whenever the agent finds a match, it sends an SNMP trap to the configured management systems. For more information about formatting the `logMonMatch` Trap PDU, see `logMonMatch` Trap in the chapter “Private Enterprise Traps.”

Each row of the Log Monitor table represents the monitoring of a log file for a particular regular expression. The following illustration shows example entries for the Log Monitoring table:



The image shows a window titled "SNMP Table Display" with a table titled "Table: logMonitorTable" and "Path: 1.3.6.1.4.1.546.11.1.1 Rows:4 Columns:7". The table has the following data:

| logMonitorLogFile | logMonitorRegularExpression | logMonitorNumberTraps | logMonitorLastTrap | logMonitorLastMatch | logMonitorSta |
|---------------------|-----------------------------|-----------------------|--------------------|---------------------|---------------|
| /usr/adm/sulog | su.* - | 0 | 0d 0:0:0 | | active(1) |
| /var/log/daemon-log | su.*fail | 0 | 0d 0:0:0 | | active(1) |
| /tmp/testprog.log | start | 2 | 5d 17:9:53 | program starting | active(1) |
| /tmp/testprog.log | stop | 1 | 5d 17:9:53 | program stopping | active(1) |

At the bottom of the window are "OK" and "Next-Page" buttons.

For more information about configuring CA eHealth SystemEDGE log file monitoring, see the chapter “Configuring Log File Monitoring.”

History Table

The CA eHealth SystemEDGE agent can track the values of various integer-based MIB objects (counters, gauges, and so on) over time and store them for later retrieval. This functionality, commonly referred to as history sampling, can greatly reduce the amount of management station polling across the network. The agent provides this functionality through two SNMP MIB tables:

- History Control table
- History table

The History Control table contains parameters that describe the data that will be sampled and stored in the History table. Each row in the control table defines a specific data-collection function by assigning values to the parameters (column objects) of the table. One or more rows (stored samples) in the History table are associated with that single control row.

Each control table row is assigned a unique index value (empireHistoryCtrlIndex). A row defines the data-collection function by specifying the object-instance to be sampled, how often to sample (in multiples of 30 seconds), and the number of samples to keep (buckets). Associated with each data-collection function (row of the control table) is a set of rows of the History table. Each row of the History table, also named bucket, holds the value of the specified MIB object gathered during one sampling interval.

As each sampling interval occurs, the agent adds a new row to the History table with the same empireHistoryIndex value as other rows for this data-collection function. This new row corresponds to the single row in the History Control table and has an empireHistorySampleIndex value one greater than the SampleIndex of the previous sample.

History Sampling Examples

The following table shows sample entries in the History Control table:

| Index | Desc | Int | ObjID | Type | Bckts-Req | Bckts-Grant | Last-Call | Create-Time | Status |
|-------|------------------------|-----|-------------------|------|-----------|-------------|------------|-------------|-----------|
| 1 | 1 Minute CPU Load Avg. | 60 | loadAverage1Min.0 | 2 | 5 | 5 | 0d 6:30:00 | 0d 0:0:0 | active(1) |
| 2 | Memory In Use | 30 | memInUse.0 | 2 | 10 | 10 | 0d 6:30:30 | 0d 0:0:0 | active(1) |
| 3 | Swap Capacity | 900 | swapCapacity.0 | 2 | 48 | 48 | 0d 6:30:00 | 0d 2:00:00 | active(1) |

The History Control table uses the SNMPv2 Structure of Management Information (SMI) Row Status Textual Convention for adding and deleting rows.

In this table, the rows are defined as follows:

- Control row 1 causes the agent to poll the 1-minute load average every minute and to store the most recent 5 samples (5 minutes).
- Control row 2 causes the agent to poll the memoryInUse MIB variable every 30 seconds and to store the most recent 10 samples (5 minutes).
- Control row 3 causes the agent to poll the swapCapacity variable (percentage of swap in use) every 15 minutes and to store the most recent 48 samples. By storing 48 samples, a management system needs to upload these samples only once every 12 hours (that is, 48 samples at 15-minute intervals provide 12 hours of coverage).

The following table shows the history samples that were stored in the History table as a result of the sampling configuration defined in the sample History Control (see the table in History Sampling Examples):

| Index | SampleIndex | StartTime | SampleTime | Value |
|-------|-------------|-----------|------------|-------|
| 1 | 387 | 0d 0:0:0 | 0d 6:26:00 | 2 |
| 1 | 388 | 0d 0:0:0 | 0d 6:27:00 | 0 |
| 1 | 389 | 0d 0:0:0 | 0d 6:28:00 | 1 |
| 1 | 390 | 0d 0:0:0 | 0d 6:29:00 | 2 |
| 1 | 391 | 0d 0:0:0 | 0d 6:30:00 | 2 |
| 2 | 772 | 0d 0:0:0 | 0d 6:26:00 | 32204 |
| 2 | 773 | 0d 0:0:0 | 0d 6:26:30 | 32213 |
| : | : | : | : | : |
| : | : | : | : | : |
| 2 | 781 | 0d 0:0:0 | 0d 6:30:30 | 32224 |
| 3 | 1 | 0d 2:0:0 | 0d 2:00:00 | 70 |
| 3 | 2 | 0d 2:0:0 | 0d 2:15:00 | 64 |
| 3 | 3 | 0d 2:0:0 | 0d 2:30:00 | 66 |
| : | : | : | : | : |
| : | : | : | : | : |
| 3 | 19 | 0d 2:0:0 | 0d 6:30:00 | 80 |

For more information about using the History Control table, see the chapter “Configuring History Collection.”

Disk Statistics Group

The Disk Statistics group provides disk I/O statistics. Each table entry provides the latest disk statistics for one disk. The agent periodically (every 60 seconds) checks the status of the system data structures for each disk and records the values in the table. Following are the Disk Statistics group MIB objects:

diskStatsIndex

Specifies the index in the Disk Statistics table.

diskStatsQueueLength

Specifies the average number of operations waiting.

diskStatsServiceTime

Specifies the average service time in milliseconds.

diskStatsUtilization

Specifies the percentage of utilization.

diskStatsKBytesTransferred

Specifies the total KB transferred to and from this disk.

diskStatsTransfers

Specifies the total number of transfers.

diskStatsReads

Specifies the total number of read operations.

diskStatsWrites

Specifies the total number of write operations.

diskStatsHostmibDevTableIndex

Specifies the index of this disk in the Host Resources MIB Device table.

diskStatsLastUpdate

Specifies the time of the last stats update.

When you are using the Disk Statistics table, see the `diskStatsLastUpdate` column for information about the time (in `TimeTicks`) that this row was last updated. This column lets you know whether the statistics have been updated since you last queried the table. The values for `diskStatsQueueLength`, `diskStatsServiceTime`, and `diskStatsUtilization` are all calculated over the interval between updates.

Enabling Collection of Disk-Performance Statistics

For Windows and AIX systems, you must manually configure the systems to collect disk-performance statistics. The CA eHealth SystemEDGE agent can monitor these statistics only if you instruct the operating system to track them.

Note: Enabling the collection of these statistics decreases the disk throughput.

Configure Disk Performance Statistics Collection for Windows Systems

You can enable and disable the collection of disk statistics for Windows systems.

To enable collection of disk statistics for Windows 2000 and Legacy systems

1. Log in to the system you want to monitor as an Administrator.
2. Enter the following at the command prompt:

```
diskperf -y
```
3. Restart the system.

Note: Disk performance counters are permanently enabled on systems beyond Windows 2000. Both Logical and Physical Disk Performance counters are automatically enabled on demand.

To disable collection of disk statistics for Windows 2000 and Legacy systems

1. Log in to the system you want to monitor as an Administrator.
2. Enter the following at the command prompt:

```
diskperf -n
```
3. Restart the system.

Note: Disk performance counters are permanently enabled on systems beyond Windows 2000. Both Logical and Physical Disk Performance counters are automatically enabled on demand.

Configure Disk Performance Statistics Collection for AIX Systems

You can enable and disable the collection of disk statistics for AIX systems.

To enable collection of disk statistics for AIX systems

1. Log in to the system you want to monitor with root privileges.
2. Enter the following at the command prompt:

```
chdev -l sys0 -a iostat=true
```

To disable collection of disk statistics for AIX systems

1. Log in to the system you want to monitor with root privileges.
2. Enter the following at the command prompt:

```
chdev -l sys0 -a iostat=false
```

CPU Statistics Group

The CPU Statistics group provides performance statistics for each CPU. Each table entry provides the latest statistics for one CPU. The agent periodically (every 60 seconds) checks the status of the system data structures for each CPU and records the values in the table. Following are the CPU Statistics MIB objects:

cpuStatsIndex

Specifies the index in this table.

cpuStatsDescr

Specifies a textual description of the CPU type.

cpuStatsIdle

Specifies the total number of ticks spent in Idle mode.

cpuStatsUser

Specifies the total number of ticks spent in User mode.

cpuStatsSys

Specifies the total number of ticks spent in System mode.

cpuStatsWait

Specifies the total number of ticks spent in Wait mode.

cpuStatsLastUpdate

Specifies the time of the last update.

cpuStatsIdlePercent

Specifies the percentage of sample time spent in Idle mode.

cpuStatsUserPercent

Specifies the percentage of sample time spent in User mode.

cpuStatsSysPercent

Specifies the percentage of sample time spent in Sys mode.

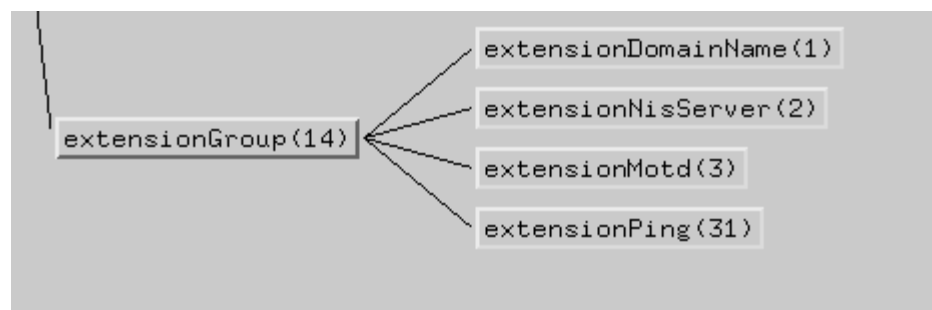
cpuStatsWaitPercent

Specifies the percentage of sample time spent in Wait mode.

When you are using the CPU statistics table, see the `cpuStatsLastUpdate` column for information about the time (in `TimeTicks`) that this row was last updated. This column also lets you know whether the statistics have been updated since you last queried the table. The values for `cpuStatsIdlePercent`, `cpuStatsUserPercent`, `cpuStatsSysPercent`, and `cpuStatsWaitPercent` are all calculated over the interval between updates.

Extension Group

The Extension group provides a powerful mechanism for extending the Systems Management MIB to include a wide range of information about your systems and applications. This group contains 232 unspecified scalar MIB variables that you can configure. In response to an SNMP Get request for one of these variables, the CA eHealth SystemEDGE agent invokes the command that you specify for the variable and returns the value that is returned from the command. Using an SNMP Set operation, you can also pass parameters to a command. The following illustration shows the organization of a sample Extension group:



For more information about using the Extension group variables, see the chapter "Adding Custom MIB Objects."

Chapter 7: Private Enterprise Traps

In addition to supporting the standard MIB-II traps, the CA eHealth SystemEDGE agent supports a number of private-enterprise trap types that have been defined for use with the agent's self-monitoring capabilities. This chapter describes the format of the Trap PDUs that the CA eHealth SystemEDGE agent can send.

This section contains the following topics:

[Format of Trap PDUs](#) (see page 143)

[SNMPv1 Trap Format](#) (see page 156)

Format of Trap PDUs

The Systems Management MIB file, `empire.asn1`, defines enterprise-specific traps for use with the CA eHealth SystemEDGE agent. This section describes the PDU format, including the information in the variable-bindings fields, for each type of trap.

Variable bindings are optional fields in a Trap PDU that associate a particular object instance with its current value.

Note: The text in this chapter is taken from the Systems Management MIB definition, which exists in the `doc` subdirectory of the CA eHealth SystemEDGE agent distribution.

All traps sent by the CA eHealth SystemEDGE agent contain the following enterprise system object identifier (`sysObjectID`) `empire(546).1.1`.

monitor Trap

The following code shows the format of a monitor trap, which the CA eHealth SystemEDGE agent sends to indicate that a monitor event has occurred. The agent sends this trap when the expression *monCurrVal monOperator monValue* evaluates to True.

```
monitorEvent OBJECT IDENTIFIER ::= { traps 1 }

monitorTrap TRAP-TYPE

ENTERPRISE empire

VARIABLES { monDescr, monOID, monCurrVal, monValue,
monRowStatus, monOperator, monIndex, monFlags }

DESCRIPTION
"A Monitor event has occurred. Recall a
monitor event occurs when a Monitor Table row
expression evaluates to true. The expression
is: 'monCurrVal monOperator monValue'."

::= 1
```


monitorEntryNotReady Trap

The following code shows the format of a monitorEntryNotReady trap, which the CA eHealth SystemEDGE agent sends to indicate that the monRowStatus field of a Monitor table entry is set to notReady(3).

```
monitorEntryNotReadyEvent OBJECT IDENTIFIER ::=
{ traps 3 }
```

```
monitorEntryNotReadyTrap TRAP-TYPE
```

```
ENTERPRISE empire
```

```
VARIABLES { monDescr, monOID, monCurrVal, monValue,
monRowStatus, monOperator, monIndex, monFlags }
```

DESCRIPTION

"This trap is sent when a Monitor Table entry's monRowStatus is set to 'notReady(3)'. One reason this may occur is that the object-instance identifier being monitored by its corresponding Monitor Table entry is no longer in existence. For example, when a process is being monitored (via the Empire processTable), and that process should exit, its corresponding entry in the Empire processTable would no longer exist. Consequently, the object-instance identifier (for that process) being monitored would no longer exist and the agent would send a monitorEntryNotReady trap to all properly configured managers."

```
::= 3
```

logMonMatch Trap

The following code shows the format of a logMonMatch trap, which the CA eHealth SystemEDGE agent sends to indicate that the log-file-monitoring subsystem has detected a match in a log file that the agent is currently monitoring.

```
logMonMatchEvent OBJECT IDENTIFIER ::= { traps 4 }
```

```
logMonMatchTrap TRAP-TYPE
```

```
ENTERPRISE empire
```

```
VARIABLES { logMonitorLogFile, logMonitorRegularExpression,  
logMonitorLastTrap, logMonitorLastMatch,  
logMonitorDescr, logMonitorIndex, logMonitorFlags }
```

```
DESCRIPTION
```

```
"This trap is sent when the log monitoring  
subsystem detects a match in a log file it is  
currently monitoring. Periodically, the agent  
stats each log file for changes; if any changes  
have occurred, the agent scans only those changes  
for a pattern match. Pattern matches result in  
logMonMatch events. Changes to log files occur  
when new entries are added by syslogd(1M) or other  
logging daemons."
```

```
::= 4
```

logMonNotReady Trap

The following code shows the format of a logMonNotReady trap, which the CA eHealth SystemEDGE agent sends to indicate that the status of an entry in the Log Monitor table has changed to notReady.

```
logMonNotReadyEvent OBJECT IDENTIFIER ::= { traps 5 }
```

```
logMonNotReadyTrap TRAP-TYPE
```

```
ENTERPRISE empire
```

```
VARIABLES { logMonitorLogFile, logMonitorRegularExpression,  
logMonitorLastTrap, logMonitorLastMatch,  
logMonitorDescr, logMonitorIndex, logMonitorFlags }
```

DESCRIPTION

"This trap is sent when the status of a log monitoring entry becomes 'notReady(3)'. An entry becomes 'notReady(3)' if an error occurs during log file scanning, if the regular expression is syntactically incorrect, or if the log file does not exist. An entry that is 'notReady(3)' will undergo no further evaluation until its status becomes 'active(1)'."

```
::= 5
```

ntEventMonMatch Trap

The following code shows the format of an ntEventMonMatch trap, which the CA eHealth SystemEDGE agent sends to indicate that the event-log-monitoring subsystem has detected a match in a Windows event log file that the agent is currently monitoring.

```
ntEventMonMatchEvent OBJECT IDENTIFIER ::=
{ traps 7 }

ntEventMonMatchTrap TRAP-TYPE

ENTERPRISE empire

VARIABLES { ntEventMonLog, ntEventMonTypeLastMatch,
ntEventMonTime, ntEventMonSrcLastMatch,
ntEventMonDescLastMatch, ntEventMonDescr,
ntEventMonIndex, ntEventMonFlags }

DESCRIPTION
"This trap is sent when the event log monitoring
subsystem detects a match in a log it is
currently monitoring. Periodically, the agent
checks each event log for changes; if any changes
have occurred, the agent scans only those changes
for a pattern match. Pattern matches result in
ntEventMonMatch events."

::= 7
```

ntEventMonNotReady Trap

The following code shows the format of an ntEventMonNotReady trap, which the CA eHealth SystemEDGE agent sends to indicate that the status of a Windows event log monitoring entry has become notReady(3).

```
ntEventMonNotReadyEvent OBJECT IDENTIFIER ::=
{ traps 8 }

ntEventMonNotReadyTrap TRAP-TYPE

ENTERPRISE empire

VARIABLES { ntEventMonLog, ntEventMonTypeFilter,
ntEventMonSrcFilter, ntEventMonDescFilter,
ntEventMonDescr, ntEventMonIndex, ntEventMonFlags }

DESCRIPTION
"This trap is sent when the status of an event log
monitoring entry becomes 'notReady(3)'. An entry
becomes 'notReady(3)' if an error occurs during log
file scanning, if the regular expression is
syntactically incorrect, or if the log file does
not exist. An entry that is 'notReady(3)' will
undergo no further evaluation until its status
becomes 'active(1)'."

 ::= 8
```

monitorClear Trap

The following code shows the format of a monitorClear trap, which the CA eHealth SystemEDGE agent sends to indicate that a condition that previously existed no longer exists. The CA eHealth SystemEDGE agent sends this trap when a Monitor table entry that has a set Clear Trap flag transitions from True to False.

```
monitorClearEvent OBJECT IDENTIFIER ::= { traps 9 }
```

```
monitorClearTrap TRAP-TYPE
```

```
ENTERPRISE sysmgmt
```

```
VARIABLES { monDescr, monOID, monCurrVal, monValue,  
monRowStatus, monOperator, monIndex, monFlags }
```

DESCRIPTION

"This trap is sent when a Monitor Table entry transitions from True to False and is using a clear trap flag. This trap indicates that the condition that previously had existed no longer does. This Trap provides management software the ability to determine that an alarm can be canceled or marked as corrected. This event only occurs when a monitor table entry evaluates to True and then evaluates to False. This Trap is sent each time the entry transitions from True to false."

```
::= 9
```

processStop Trap

The following code shows the format of a processStop trap, which the CA eHealth SystemEDGE agent sends to indicate that a process it was monitoring has either stopped running or is in a state where it cannot run.

```
processStopEvent OBJECT IDENTIFIER ::= { traps 10 }
```

```
processStopTrap TRAP-TYPE
```

```
ENTERPRISE sysmgmt
```

```
VARIABLES { pmonIndex, pmonDescr, pmonAttribute,  
pmonCurrVal, pmonOperator, pmonValue,  
pmonFlags, pmonRegExpr, pmonCurrentPID }
```

DESCRIPTION

"This Trap is sent when using a Process Monitor Table entry to monitor the state of a process. When the processing being monitored dies or transitions into a Zombie (or not runnable state), this Trap is sent. This Trap is sent if the value of pmonFlags does not preclude sending Traps."

```
::= 10
```

processStart Trap

The following code shows the format of a processStart trap, which the CA eHealth SystemEDGE agent sends to indicate that a process has restarted (and has been reacquired by the CA eHealth SystemEDGE agent).

```
processStartEvent OBJECT IDENTIFIER ::= { traps 11 }
```

```
processStartTrap TRAP-TYPE
```

```
ENTERPRISE sysmgmt
```

```
VARIABLES { pmonIndex, pmonDescr, pmonAttribute,  
pmonCurrVal, pmonOperator, pmonValue,  
pmonFlags, pmonRegExpr, pmonCurrentPID }
```

DESCRIPTION

"This Trap is sent when using a Process Monitor Table entry to monitor the state of a process. When a process is re-started, and subsequently re-acquired by the SystemEDGE agent, this Trap is sent. This Trap is sent if the value of pmonFlags specifies that processStart Traps should be sent."

```
::= 11
```


processThreshold Trap

The following code shows the format of a processThreshold trap, which the CA eHealth SystemEDGE agent sends to indicate that an attribute of a process that it is monitoring has reached the specified threshold.

```
processThresholdEvent OBJECT IDENTIFIER ::=
{ traps 12 }
```

```
processThresholdTrap TRAP-TYPE
```

```
ENTERPRISE sysmgmt
```

```
VARIABLES { pmonIndex, pmonDescr, pmonAttribute,
pmonCurrVal, pmonOperator, pmonValue,
pmonFlags, pmonRegExpr, pmonCurrentPID }
```

DESCRIPTION

"This Trap is sent when using a Process Monitor Table entry to monitor some attribute (e.g. memory usage, process size) of a process for some threshold. When a Process Monitor table expression evaluates to True, this Trap is sent. The expression is: 'pmonCurrVal pmonOperator pmonValue'. This Trap is sent if the value of pmonFlags does not preclude the sending of Traps."
 ::= 12

processClear Trap

The following code shows the format of a processClear trap, which the CA eHealth SystemEDGE agent sends to indicate that a process attribute for which it previously sent a processThreshold event is no longer at the threshold level.

```
processClearEvent OBJECT IDENTIFIER ::= { traps 13 }
```

```
processClearTrap TRAP-TYPE
```

```
ENTERPRISE sysmgmt
```

```
VARIABLES { pmonIndex, pmonDescr, pmonAttribute,  
pmonCurrVal, pmonOperator, pmonValue,  
pmonFlags, pmonRegExpr, pmonCurrentPID }
```

DESCRIPTION

"This Trap is sent when using a Process Monitor Table entry to monitor some attribute (e.g. memory usage, process size) of a process for some threshold. When the threshold is crossed, a processThreshold Trap is sent. When the attribute threshold expression first transitions from True to False, this Trap is sent."

```
::= 13
```

license Trap

The following code shows the format of a license trap, which the CA eHealth SystemEDGE agent sends to indicate that it did not find a valid license for itself or one of the CA eHealth AIMs.

```
licenseEvent OBJECT IDENTIFIER ::= { traps 16 }
```

```
licenseTrap TRAP-TYPE
```

```
ENTERPRISE sysmgmt
```

```
VARIABLES { sysedgeLicenseString }
```

DESCRIPTION

"This Trap is sent when SystemEDGE or associated modules failed to find a valid license. It can be used in conjunction with auto-licensing or remote-licensing starting with SystemEDGE 4.0. This Trap contains a single MIB object denoting which product or module failed to find a valid license and a string containing the license information for that product or module."
 ::= 16

addrChangeTrap

The following code shows the format of an addrChange trap, which the CA eHealth SystemEDGE agent sends to indicate that the underlying IP address has changed.

```
addrChangeEvent OBJECT IDENTIFIER ::= { traps 18 }
```

```
addrChangeTrap TRAP-TYPE
```

```
ENTERPRISE sysmgmt
```

```
VARIABLES { nodename sysedgeAddressList }
```

DESCRIPTION

"This Trap is sent when SystemEDGE detects that its underlying IP address has changed perhaps due to DHCP or other administrative means. It includes up to the last 5 IP addresses that this system was configured with. The addresses are ordered with most recently used addresses occurring first in the address list. This Trap may be used on multi-homed systems."
 ::= 18

procGroupChangeTrap

The following code shows the format of a procGroupChange trap, which the CA eHealth SystemEDGE agent sends to indicate that the process group has changed.

```
procGroupChangeEvent OBJECT IDENTIFIER ::=
{ traps 19 }
```

```
procGroupChangeTrap TRAP-TYPE
```

```
ENTERPRISE sysmgmt
```

```
VARIABLES { pgmonIndex, pgmonDescr, pgmonFlags, pgmonNumProcs, pgmonProcRegExpr,
pgmonRowStatus, pgmonPIDList, pgmonStatusList}
```

DESCRIPTION

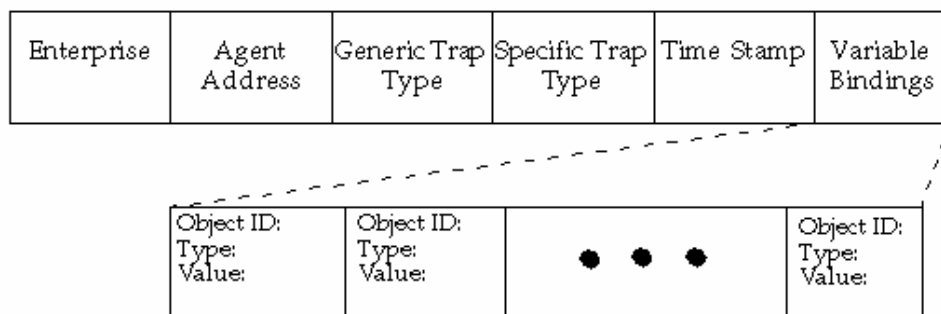
"Membership in a process group has changed. (For example, members have joined or left the group or have changed.)"

```
::= 19
```

SNMPv1 Trap Format

The CA eHealth SystemEDGE agent sends Trap PDUs to the SNMPv1 Trap port (UDP/162). The following illustration shows the structure of a SNMPv1 Trap PDU.

SNMP Trap PDU



Following are the components of the SNMPv1 Trap PDU:

Enterprise

Specifies the System Object ID of the sender: empire(546).1.1.

Source address

Specifies the IP address of the sending host.

Generic Trap Type

Specifies the generic trap type, which can be one of the following:

- coldStart(0)
- warmStart(1)
- linkDown(2)
- linkup(3)
- authenFailure(4)
- egpNeighborloss(5)
- enterpriseSpecific(6)

Specific Trap Type

Specifies the enterprise-specific trap type.

Time Stamp

Specifies the value of sysUptime when the trap was sent. The value is always 0 for sendtrap.

Variable Bindings

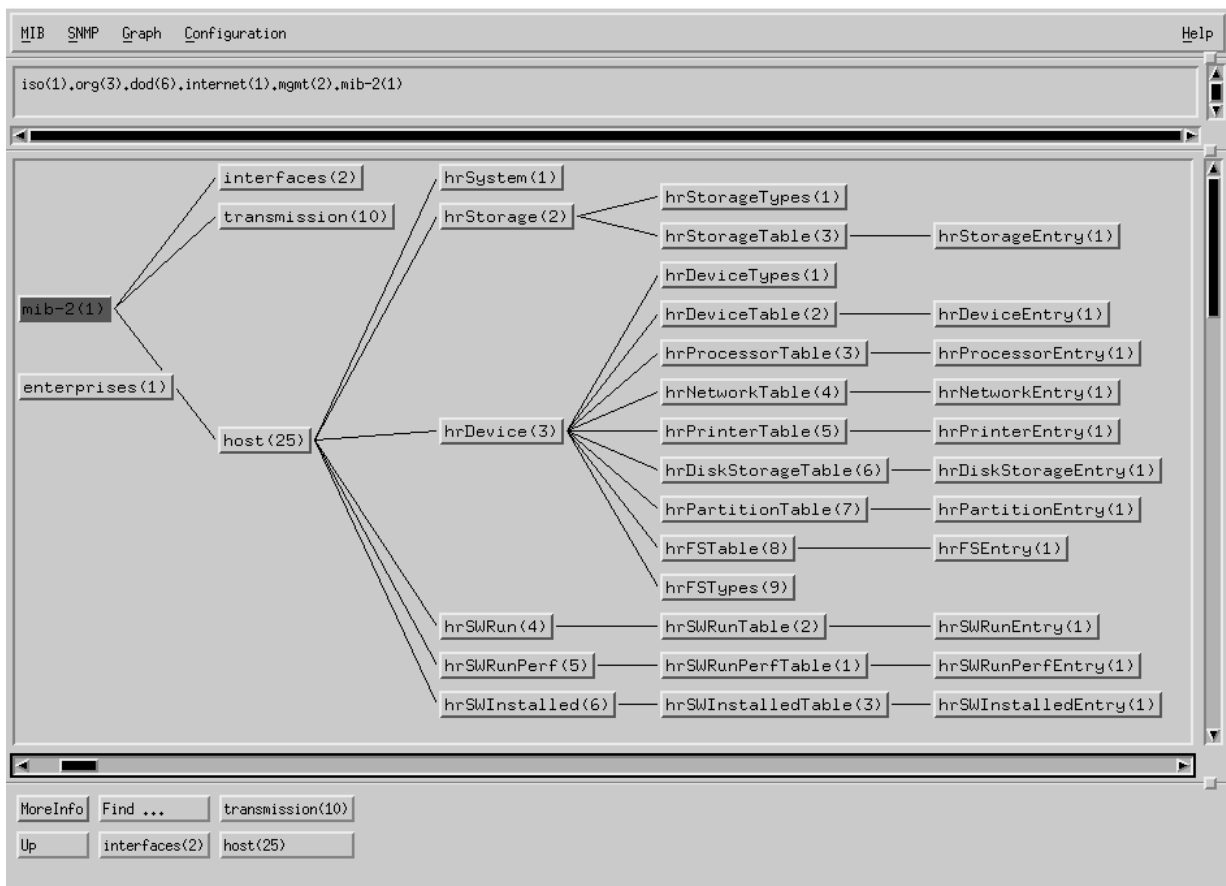
Specifies the array of MIB variables and their values.

Chapter 8: Host Resources MIB

This chapter describes the management information available through the IETF Host Resources MIB (RFC 1514).

Note: The examples in this chapter do not describe the entire Host Resources MIB. For a description of the entire MIB, see the `hostmib.asn1` file, which is included in the CA eHealth SystemEDGE agent distribution.

The Host Resources MIB defines a set of objects that can manage host computers, which are independent of the operating system, network services, or any software application. The objects defined in the Host Resources MIB are common across many computer system architectures. The following illustration shows the overall organization of the Host Resources MIB:



This section contains the following topics:

[Host Resources System Group](#) (see page 160)

[Host Resources Storage Group](#) (see page 161)

[Host Resources Device Group](#) (see page 162)

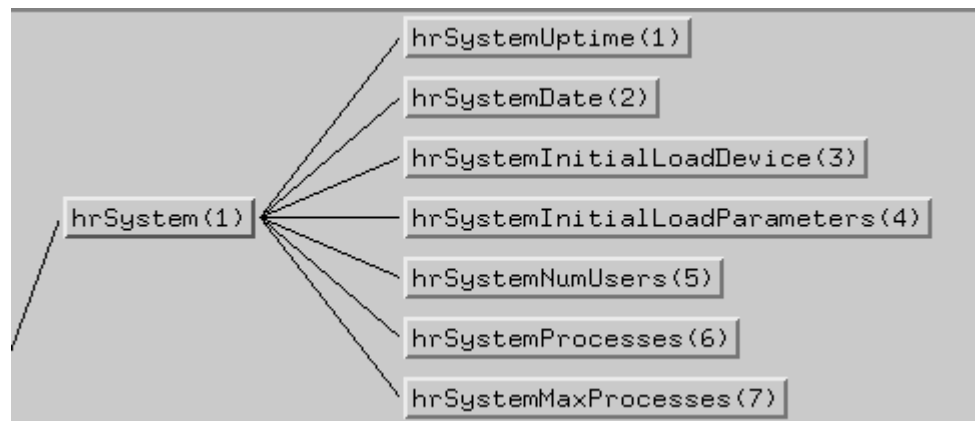
[Host Resources Running Software Group](#) (see page 167)

[Host Resources Installed Software Group](#) (see page 168)

[Unsupported MIB Objects on Windows Systems](#) (see page 169)

Host Resources System Group

The Host Resources System (hrSystem) group provides you with information that pertains to the host system as a whole. The following illustration shows the organization of the hrSystem group:



The following information is available through the Host Resources System group:

hrSystemUptime

Specifies the amount of time since the host was last initialized.

hrSystemDate

Specifies the host's settings for the local date and time of day.

hrSystemInitialLoadDevice

Specifies the device from which the host loads its initial operating system configuration.

hrSystemInitialLoadParameters

Specifies the pathname and parameters applied when loading the initial operating system.

hrSystemNumUsers

Specifies the number of user sessions for which the host is storing state information.

hrSystemProcesses

Specifies the number of processes currently loaded or running on the system.

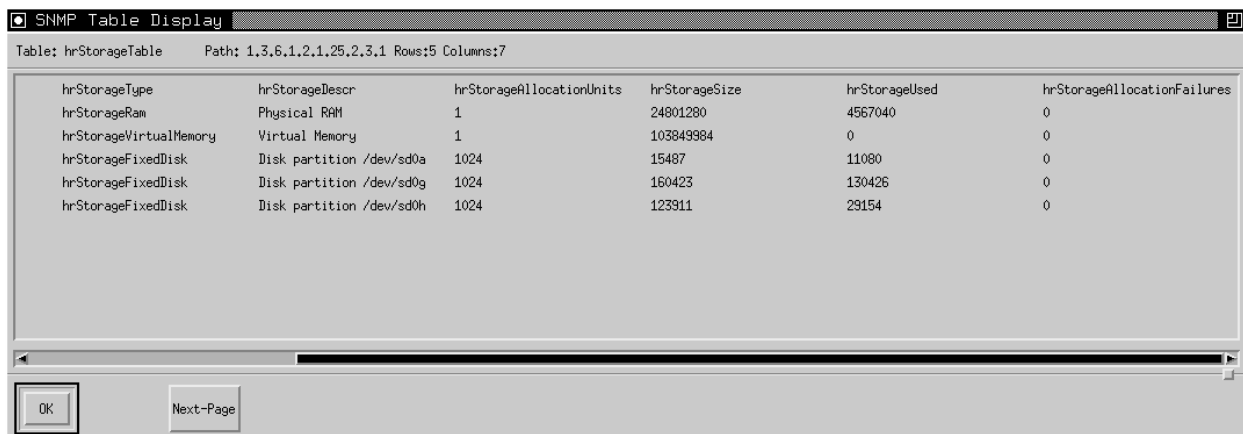
hrSystemMaxProcesses

Specifies the maximum number of processes that can be run on this system.

Host Resources Storage Group

The Host Resources Storage group (hrStorageTable) lists the logical areas of storage allocated on the host system. These storage areas include file systems and disk partitions that might be seen by an application, rather than physical storage such as tapes and floppy drives.

This table is intended to be a useful diagnostic for out-of-memory and out-of-buffers types of failures. In addition, it can be a useful performance-monitoring tool for tracking memory, disk, or buffer usage. The following illustration shows the output of the Host Resources Storage group from a Sun SPARC IPX:



The image shows a window titled "SNMP Table Display" with a table of storage information. The table has 7 columns: hrStorageType, hrStorageDescr, hrStorageAllocationUnits, hrStorageSize, hrStorageUsed, and hrStorageAllocationFailures. The table contains 5 rows of data.

| hrStorageType | hrStorageDescr | hrStorageAllocationUnits | hrStorageSize | hrStorageUsed | hrStorageAllocationFailures |
|------------------------|--------------------------|--------------------------|---------------|---------------|-----------------------------|
| hrStorageRam | Physical RAM | 1 | 24801280 | 4567040 | 0 |
| hrStorageVirtualMemory | Virtual Memory | 1 | 103849984 | 0 | 0 |
| hrStorageFixedDisk | Disk partition /dev/sd0a | 1024 | 15487 | 11080 | 0 |
| hrStorageFixedDisk | Disk partition /dev/sd0g | 1024 | 160423 | 130426 | 0 |
| hrStorageFixedDisk | Disk partition /dev/sd0h | 1024 | 123911 | 29154 | 0 |

At the bottom of the window, there are two buttons: "OK" and "Next-Page".

Host Resources Device Group

The Host Resources Device (hrDevice) group contains several tables that provide information about the devices contained by the host system. The main table in this group is the hrDeviceTable. It lists all of the devices that the host contains. The hrDevice group also includes a number of device-specific tables that provide more detailed information for particular device types.

For example, the group includes device-specific tables for the following:

- Processors
- Networks
- Printers
- Disk storage
- Partitions
- File systems

If the hrDeviceTable shows that the host contains a Printer device, for example, you can retrieve more detailed information about the printer from the hrPrinterTable. The following sections describe the hrDeviceTable and the device-specific tables.

Device Table

The hrDeviceTable lists the devices in the host system. For each device in the hrDeviceTable, the table lists the following:

- Device type
- Description of the device
- Status (for example, running or down)
- Number of errors detected for the device

The following illustration shows an hrDeviceTable for a Sun SPARC IPX:

| hrDeviceType | hrDeviceDescr | hrDeviceID | hrDeviceSt |
|----------------------|---|------------|------------|
| hrDeviceOther | SUNW,SPARCstation-LX, Sun Sparc Workstation | null | running(2) |
| hrDeviceProcessor | TI,TMS390S10 | null | running(2) |
| hrDeviceOther | openprom, PROM monitor configuration interface | null | running(2) |
| hrDeviceSerialPort | zs0, Zilog 8530 SCC Serial Communications Driver | null | running(2) |
| hrDeviceSerialPort | zs1, Zilog 8530 SCC Serial Communications Driver | null | running(2) |
| hrDeviceOther | sbu0, Main Memory and Bus I/O Space | null | running(2) |
| hrDeviceParallelPort | bpp0, Bidirectional parallel port | null | running(2) |
| hrDeviceNetwork | FORE,sba-2000, Fore ATM SBus Adapter | null | unknown(1) |
| hrDeviceNetwork | SUNW,DBRIe0, Dual Basic Rate ISDN Interface and Speaker | null | unknown(1) |
| hrDeviceVideo | cgsix0, Accelerated 8-bit Color Frame Buffer | null | running(2) |

Inventory Tracking and Asset Management

The hrDeviceTable is especially useful for inventory tracking and asset management. Instead of manually opening and inspecting each workstation on your network to determine the number and types of ethernet cards, serial devices, audio devices, disk storage devices, and so on, you can use your central management system to retrieve the hrDeviceTable from each of the systems on your network.

Processor Table

The `hrProcessorTable` lists device-specific information about the system's processors. If the system contains only a single processor, the `hrProcessorTable` includes only one row. The columns of the `hrProcessorTable` provide the product ID of the firmware associated with the processor (if such an ID has been assigned and made available by the processor vendor), and the processor load. The load is the average over the last minute of the percentage of time that the processor was *not* idle.

Tracking Processor Load

By monitoring (graphing) the Processor Load average, you can visually track the load put on the processor. For example, if the graph shows constant, high levels (indicating that the processor was heavily loaded), you may decide that the host system is being overworked with too many users and processes, so you could then take steps to limit the number of users or have some of the processes run on another system.

A temporary spike in a Processor Load graph could be an indication that a processor-intensive process is running. To avoid problems that might result from such a heavy processor load, you can reschedule that process to run at a time when processor load is less heavy.

Disk Storage Table

The hrDiskStorageTable provides information about the host's disk-storage devices. For each disk-storage device, the columns of the table list the following:

- Storage media type (for example, hard disk or floppy disk)
- Whether the disk is removable
- Storage capacity (in KB)
- Whether the disk-storage device permits read-write or read-only access

The following illustration shows an hrDiskStorageTable for a Sun SPARC IPX:

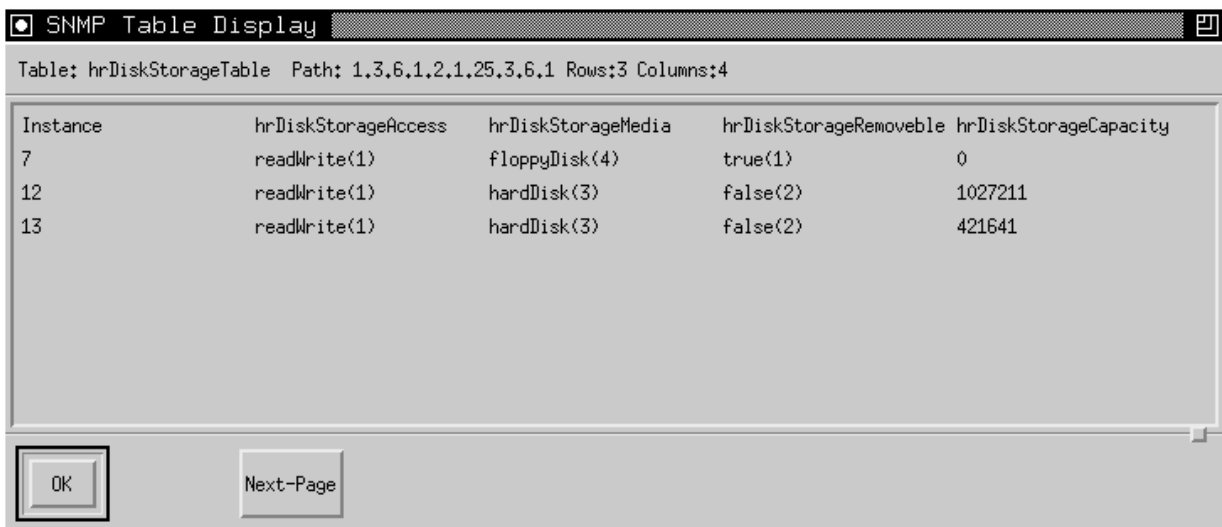


Table: hrDiskStorageTable Path: 1.3.6.1.2.1.25.3.6.1 Rows:3 Columns:4

| Instance | hrDiskStorageAccess | hrDiskStorageMedia | hrDiskStorageRemovable | hrDiskStorageCapacity |
|----------|---------------------|--------------------|------------------------|-----------------------|
| 7 | readWrite(1) | floppyDisk(4) | true(1) | 0 |
| 12 | readWrite(1) | hardDisk(3) | false(2) | 1027211 |
| 13 | readWrite(1) | hardDisk(3) | false(2) | 421641 |

OK Next-Page

Partition Table

The hrPartitionTable shows the partitions that have been configured for the disk-storage devices. For each partition, the table lists the following:

- Textual description of the partition
- Partition ID
- Size of the partition (in KB)
- Index of the file system mounted on that partition

The following illustration shows a sample hrPartitionTable for a Sun SPARC IPX system:

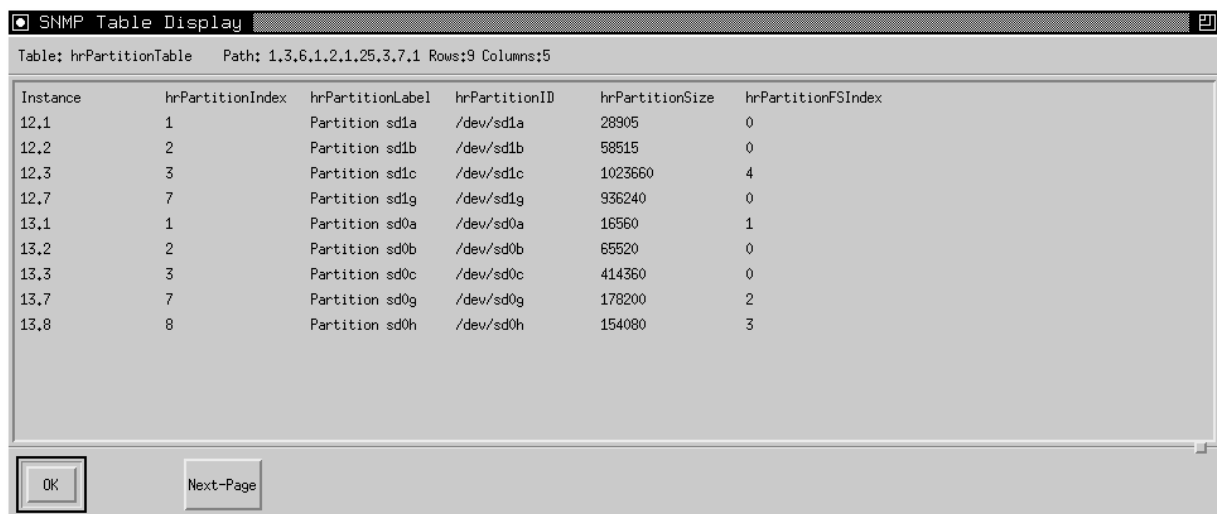


Table: hrPartitionTable Path: 1.3.6.1.2.1.25.3.7.1 Rows:9 Columns:5

| Instance | hrPartitionIndex | hrPartitionLabel | hrPartitionID | hrPartitionSize | hrPartitionFSIndex |
|----------|------------------|------------------|---------------|-----------------|--------------------|
| 12.1 | 1 | Partition sd1a | /dev/sd1a | 28905 | 0 |
| 12.2 | 2 | Partition sd1b | /dev/sd1b | 58515 | 0 |
| 12.3 | 3 | Partition sd1c | /dev/sd1c | 1023660 | 4 |
| 12.7 | 7 | Partition sd1g | /dev/sd1g | 936240 | 0 |
| 13.1 | 1 | Partition sd0a | /dev/sd0a | 16560 | 1 |
| 13.2 | 2 | Partition sd0b | /dev/sd0b | 65520 | 0 |
| 13.3 | 3 | Partition sd0c | /dev/sd0c | 414360 | 0 |
| 13.7 | 7 | Partition sd0g | /dev/sd0g | 178200 | 2 |
| 13.8 | 8 | Partition sd0h | /dev/sd0h | 154080 | 3 |

OK Next-Page

File System Table

The hrFSTable provides information about the host's file systems, both local and remote. For each file system, the table lists the following:

- Local mount point
- Remote mount point (if the file system is being mounted from a remote machine)
- Type of file system (for example, BerkeleyFFS)
- Whether the file system permits read-write or read-only access

The following illustration shows sample output of the hrFSTable for a Sun SPARC IPX system:

Table: hrFSTable Path: 1.3.6.1.2.1.25.3.8.1 Rows:10 Columns:6

| hrFSMountPoint | hrFSRemoteMountPoint | hrFSType | hrFSAccess | hrFSStorageIndex |
|-----------------------------------|--------------------------|-----------------|--------------|------------------|
| / | | hrFSBerkeleyFFS | readWrite(1) | 3 |
| /usr | | hrFSBerkeleyFFS | readWrite(1) | 4 |
| /at22 | | hrFSBerkeleyFFS | readWrite(1) | 5 |
| /net | | hrFSOther | readOnly(2) | 0 |
| /tmp_mnt/net/ht1 | morticia:/ht1 | hrFSNFS | readWrite(1) | 0 |
| /tmp_mnt/net/usr.local-sun4-4.1.3 | morticia:/usr/local | hrFSNFS | readWrite(1) | 0 |
| /tmp_mnt/net/at1 | morticia:/at1 | hrFSNFS | readWrite(1) | 0 |
| /tmp_mnt/net/mail | morticia:/usr/spool/mail | hrFSNFS | readWrite(1) | 0 |
| /tmp_mnt/net/other | trantor:/data1/other | hrFSNFS | readWrite(1) | 0 |
| /tmp_mnt/net/local5 | lennon:/u2/local5 | hrFSNFS | readWrite(1) | 0 |

OK Next-Page

Host Resources Running Software Group

The hrSWRunTable lists the software currently running on the host system. For each running software process, the table lists the following:

- Unique identification number
- Name
- Product ID
- Directory path where the software resides
- Run-time parameters with which the software was started
- Type of software (for example, operating system or application)
- Status of the software (for example, running, runnable, not runnable, or invalid)

The following illustration shows part of an hrSWRunTable for a Sun SPARC IPX system:

| Instance | hrPartitionIndex | hrPartitionLabel | hrPartitionID | hrPartitionSize | hrPartitionFSIndex |
|----------|------------------|------------------|---------------|-----------------|--------------------|
| 12.1 | 1 | Partition sd1a | /dev/sd1a | 28905 | 0 |
| 12.2 | 2 | Partition sd1b | /dev/sd1b | 58515 | 0 |
| 12.3 | 3 | Partition sd1c | /dev/sd1c | 1023660 | 4 |
| 12.7 | 7 | Partition sd1g | /dev/sd1g | 936240 | 0 |
| 13.1 | 1 | Partition sd0a | /dev/sd0a | 16560 | 1 |
| 13.2 | 2 | Partition sd0b | /dev/sd0b | 65520 | 0 |
| 13.3 | 3 | Partition sd0c | /dev/sd0c | 414360 | 0 |
| 13.7 | 7 | Partition sd0g | /dev/sd0g | 178200 | 2 |
| 13.8 | 8 | Partition sd0h | /dev/sd0h | 154080 | 3 |

Host Resources Installed Software Group

The hrSWInstalledTable lists the software currently installed on the host system. For each software package, the table lists the following:

- Unique identifier
- Software package's name
- Description
- Installation date

This table shows the operating system patches and versions of software installed. It is ideal for checking software-version consistency between systems. The following illustration shows part of a hrSWInstalledTable for a Solaris 2.x system:

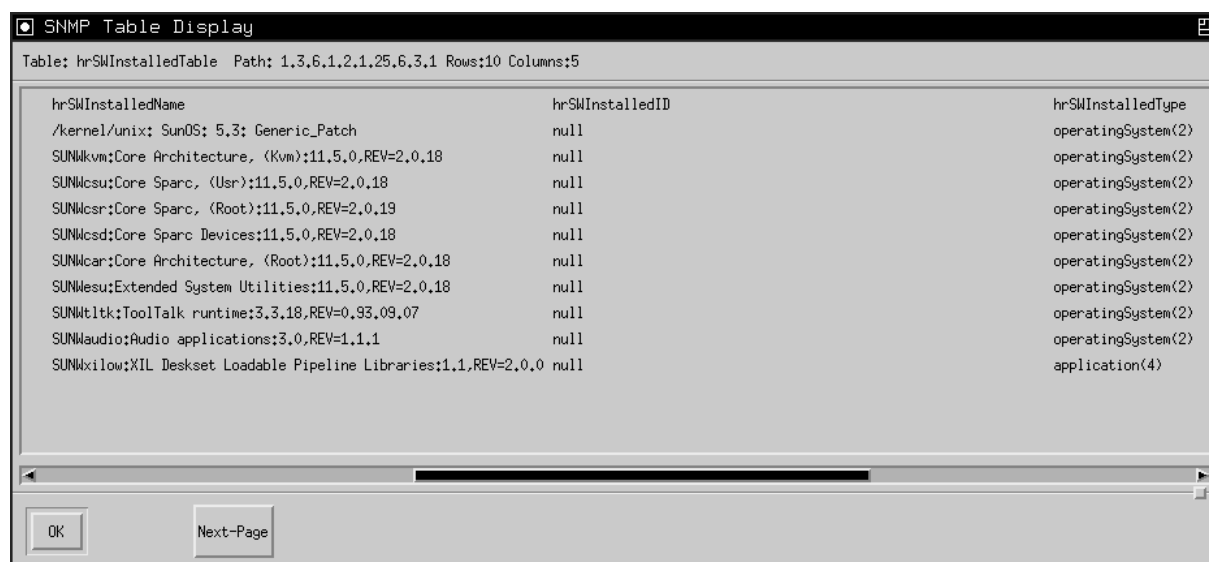


Table: hrSWInstalledTable Path: 1.3.6.1.2.1.25.6.3.1 Rows:10 Columns:5

| hrSWInstalledName | hrSWInstalledID | hrSWInstalledType |
|---|-----------------|--------------------|
| /kernel/unix; SunOS; 5.3; Generic_Patch | null | operatingSystem(2) |
| SUNWkvm:Core Architecture, (Kvm):11.5.0,REV=2.0.18 | null | operatingSystem(2) |
| SUNWcsu:Core Sparc, (User):11.5.0,REV=2.0.18 | null | operatingSystem(2) |
| SUNWcsr:Core Sparc, (Root):11.5.0,REV=2.0.19 | null | operatingSystem(2) |
| SUNWcsd:Core Sparc Devices:11.5.0,REV=2.0.18 | null | operatingSystem(2) |
| SUNWcar:Core Architecture, (Root):11.5.0,REV=2.0.18 | null | operatingSystem(2) |
| SUNWesu:Extended System Utilities:11.5.0,REV=2.0.18 | null | operatingSystem(2) |
| SUNWtlk:ToolTalk runtime:3.3.18,REV=0.93.09.07 | null | operatingSystem(2) |
| SUNWaudio:Audio applications:3.0,REV=1.1.1 | null | operatingSystem(2) |
| SUNWxilow:XIL Deskset Loadable Pipeline Libraries:1.1,REV=2.0.0 | null | application(4) |

OK Next-Page

Note: Not all systems can support this table. To determine whether it is supported for the version of the operating system you are using, see the *CA eHealth SystemEDGE Release Notes*.

Unsupported MIB Objects on Windows Systems

The CA eHealth SystemEDGE agent for Windows supports the Host Resources MIB (RFC 1514), but some of the MIB objects within this MIB module are not supported by the underlying Windows operating system. Those objects, therefore, cannot be implemented by the Windows version of the agent.

The following Host Resources MIB objects are not supported on Windows systems:

- hrSystemInitialLoadParameters
- hrStorageAllocationFailures
- hrDeviceID (hardware manufacturers have not assigned IDs)
- hrDeviceErrors
- hrProcessorFrwID (manufacturers have not assigned IDs)
- hrPrinterTable (not implemented in current release)
- hrFSRemoteMountPoint
- hrFSLastFullBackupDate
- hrFSLastPartialBackupDate
- hrFSAccess
- hrFSBootable (not implemented in current release)
- hrSWRunEntry.SWRunPath
- hrSWRunEntry.SWRunParameters
- hrSWInstalledID

Chapter 9: Configuring Threshold Monitoring

This chapter explains how to use the CA eHealth SystemEDGE agent to monitor MIB variables against user-specified thresholds.

This section contains the following topics:

- [Threshold Monitoring](#) (see page 171)
- [The Monitor Table](#) (see page 172)
- [Monitor Table Flags](#) (see page 178)
- [Monitor Table Actions](#) (see page 181)
- [Monitor Entry Correlation](#) (see page 182)
- [View the Monitor Table with CA eHealth AdvantEDGE View](#) (see page 184)
- [Assigning Entry Rows for the Monitor Table](#) (see page 184)
- [Configuring the Monitor Table](#) (see page 185)
- [monitor Directive--Add Entries to the Monitor Table](#) (see page 188)
- [Threshold Monitoring Examples](#) (see page 190)
- [edgemon Utility--Monitor Thresholds](#) (see page 202)
- [Removing Threshold Monitoring Entries](#) (see page 208)

Threshold Monitoring

The CA eHealth SystemEDGE agent includes a flexible Monitor table through which you can dynamically configure the agent to monitor any integer-based MIB variable under its control. You can specify the variable to monitor, the polling interval, a comparison operator (greater than, equal to, and so on), and a threshold value. The CA eHealth SystemEDGE agent automatically monitors that variable and sends a trap to the management system if the condition you specified is met. For more information about traps sent by the CA eHealth SystemEDGE agent, see the chapter “Private Enterprise Traps.” The agent can also execute commands on the managed system to perform management functions to correct the cause of the exception immediately. For example, you can configure the CA eHealth SystemEDGE agent to monitor the percent capacity for the root file system and to notify the manager if it becomes too full.

Note: The CA eHealth SystemEDGE agent can also monitor processes, process groups, log files, and Windows events.

The Monitor Table

The Monitor table, located in the Systems Management MIB (empire.asn1), provides information about each condition that the agent is currently monitoring. Each row represents a single condition that the agent is monitoring. For each entry, the table provides the following types of information:

- Variable that the agent is monitoring
- Interval at which the agent checks the variable
- Current value
- Conditions that will cause the agent to send a trap
- Number of traps that have been sent already

Sample Entry in the Monitor Table

This section provides an example of an entry in the Monitor table that instructs the CA eHealth SystemEDGE agent to monitor the 1-minute load average on the target system:

| Index | Description | Interval | Sample Type | Object ID | Current Value | Operator | Threshold Value | Last Call | #Traps | Last Trap | Row Status | Min Value | Max Value | Actions | Flags | Superseded By |
|-------|-----------------|----------|-------------|---------------|---------------|----------|-----------------|-------------|--------|-----------|------------|-----------|-----------|---------|-------|---------------|
| 12 | "1min load avg" | 60 | absolute | loadAvg1Min.0 | 1 | gt | 300 | 0d 22:40:00 | 0 | 0 | active | 1 | 3 | "" | 0x0 | 0 |

This entry provides the following information:

- This is the twelfth row in the table.
- Its purpose is to monitor the system's 1-minute load average.
- Every 60 seconds, the agent checks the absolute value of the loadAvg1Min MIB variable (whose current value is 1) to see if it is greater than 300 (the specified threshold value). If it is, the agent sends a trap to any configured NMS and updates the NumTraps and LastTrap columns appropriately.
- With each poll, the agent updates the CurrVal column with the value it has just retrieved, records the time in the LastCall column, and updates the MinValue and MaxValue columns if appropriate (that is, if the value just polled is the lowest or highest value observed thus far).

- The RowStatus column shows that this entry is active. In this example, the action is null, so no command will be run when the trap is sent.
- The Flags column is set to zero, which indicates the default Monitor table behavior. For more information about flags for the Monitor table, see Monitor Table Flags in this chapter.
- The SupersededBy Index is set to zero, which indicates default threshold event handling. For more information about this column, see Columns of the Monitor Table.

Columns of the Monitor Table

The following list defines the columns of the Monitor table. For a complete description of the Monitor table and its fields, see the `empire.asn1` file, located in the `doc` subdirectory of the CA eHealth SystemEDGE agent's installation.

monIndex

Specifies an integer (1 to MAXINT) that indicates the row index for this entry. Rows 1 through 10 are reserved for the agent's internal use; the index for additional rows must fall in the range of 11 to MAXINT.

Permissions: Read-only

monDesc

Specifies a quoted string, 0 to 128 characters in length, that typically contains a description of the object being monitored and a severity level for this event.

Permissions: Read-write

Default: :Default Entry:

monInterval

Specifies an integer value (30 to MAXINT) that indicates how often (in seconds) the agent should monitor the variable. For example, the value 30 instructs the agent to monitor this entry every 30 seconds. The value must be a multiple of 30 seconds.

Permissions: Read-write

Default: 60

monSampleType

Indicates whether this entry should sample the object's absolute value (`absoluteValue(1)`) or take the difference between successive samples (`deltaValue(2)`). For example, when monitoring counter variables, use `deltaValue` because it describes the rate of change. When monitoring gauges, use `absoluteValue` because it describes the object's exact value.

Permissions: Read-write

Default: `absoluteValue(1)`

monOID

Specifies the complete object-instance identifier that represents the value to be monitored. The instance portion of the object-identifier (for example, .0 for scalars) is required. The object-instance must exist and must be contained within the CA eHealth SystemEDGE agent's supported MIBs. That is, any supported (integer-based) object that exists in MIB-II, the Host Resources MIB, or the Systems Management MIB is valid. Objects should be of integer type, including counter, gauge, integer, or enumerated integer.

Permissions: Read-write

Default: 0.0

monCurrVal

Specifies the value that was last recorded for the MIB variable being monitored. Every `monInterval` seconds, the agent updates this field to reflect the latest value of the variable.

Permissions: Read-only

monOperator

Specifies the operator type, a Boolean operator, used to evaluate the expression *currval operator value*. The operator can be one of the following:

- `nop` (no operation; monitor the object's value, but do not evaluate the Boolean expression)
- `>` (greater than)
- `<` (less than)
- `>=` (greater than or equal to)
- `<=` (less than or equal to)
- `==` (equal)
- `!=` (not equal)

Permissions: Read-write

Default: `nop(1)`

monValue

Specifies the integer value to which the current value of the monitored MIB variable is compared during each monitoring cycle. If the comparison evaluates to True, the agent sends a trap. For example, if you want to be notified if the value of a gauge exceeds 100, set 100 as the monValue against which the current value of the gauge is compared.

Permissions: Read-write

Default: 0

monLastCall

Specifies the time (based on sysUpTime) at which the agent last sampled (called) the MIB variable it is monitoring. 0 indicates that the MIB variable has not yet been sampled.

Permissions: Read-only

Default: 0

monNumTraps

Specifies the number of traps that have been sent for this Monitor table entry. This value provides a useful metric for determining how often the exception condition has occurred. It also provides a means to detect a missed trap message.

Permissions: Read-only

Default: 0

monLastTrap

Specifies the time (based on sysUpTime) at which the agent last sent a trap for this Monitor table entry. 0 indicates that no traps have been sent.

Permissions: Read-only

Default: 0

monRowStatus

Specifies the status of the row, which can be one of the following:

- active
- notInService
- notReady
- createAndGo
- createAndWait

Typically, a row is either active or notInService. These values are identical in meaning to those defined by the SNMPv2 SMI RowStatus textual convention.

Permissions: Read-write

Default: createAndWait(5)

monMinValue

Specifies the lowest (minimum) value that the agent has observed since it began polling the MIB variable.

Permissions: Read-only

Default: 0

monMaxValue

Specifies the highest (maximum) value that the agent has observed since it began polling the MIB variable.

Permissions: Read-only

Default: 0

monAction

Specifies a quoted string, 0 to 256 characters in length, that specifies the full path of the command (with any parameters) to run when the expression evaluates to True and the agent sends a trap. If the string is empty, the agent performs no action for this entry.

Permissions: Read-write

Default: No action

monFlags

Specifies the unsigned integer flags value that indicates additional behavioral semantics that this row should follow during the course of its operation. For more information, see Monitor Table Flags.

Permissions: Read-write

Default: 0x0

monSupersededBy

Specifies the monitor index that acts as a parent or overriding partner to this entry. If the entry index specified in this object matches its comparison value, then this entry, being its child, is overridden or superseded, and will not trap, will not send events to syslog, and will not execute actions. For example, if you create three monitor entries to track CPU utilization at different thresholds (that is 40%, 60%, and 80%), you can have the 40% threshold entry indicate that it will be superseded by the 60% threshold entry, and the 60% threshold entry can indicate that it will be superseded by the 80% threshold entry. This means that a CPU utilization of 85% only triggers one trap or event instead of three.

Accepted values in this field are existing or future SystemEDGE monitor indexes (a positive integer value).

Note: For more information, see Monitor Entry Correlation in this chapter.

Permissions: Read-write

Default: 0

The following illustration shows a sample Monitor table:

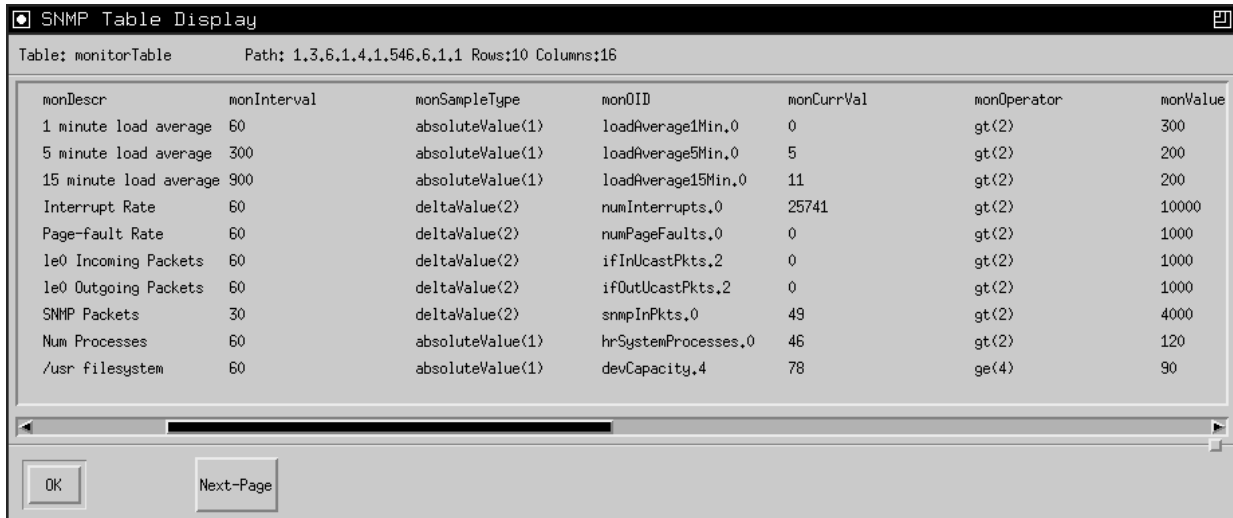


Table: monitorTable Path: 1.3.6.1.4.1.546.6.1.1 Rows:10 Columns:16

| monDescr | monInterval | monSampleType | monOID | monCurrVal | monOperator | monValue |
|------------------------|-------------|------------------|---------------------|------------|-------------|----------|
| 1 minute load average | 60 | absoluteValue(1) | loadAverage1Min,0 | 0 | gt(2) | 300 |
| 5 minute load average | 300 | absoluteValue(1) | loadAverage5Min,0 | 5 | gt(2) | 200 |
| 15 minute load average | 900 | absoluteValue(1) | loadAverage15Min,0 | 11 | gt(2) | 200 |
| Interrupt Rate | 60 | deltaValue(2) | numInterrupts,0 | 25741 | gt(2) | 10000 |
| Page-fault Rate | 60 | deltaValue(2) | numPageFaults,0 | 0 | gt(2) | 1000 |
| 1e0 Incoming Packets | 60 | deltaValue(2) | ifInUcastPkts,2 | 0 | gt(2) | 1000 |
| 1e0 Outgoing Packets | 60 | deltaValue(2) | ifOutUcastPkts,2 | 0 | gt(2) | 1000 |
| SNMP Packets | 30 | deltaValue(2) | snmpInPkts,0 | 49 | gt(2) | 4000 |
| Num Processes | 60 | absoluteValue(1) | hrSystemProcesses,0 | 46 | gt(2) | 120 |
| /usr filesystem | 60 | absoluteValue(1) | devCapacity,4 | 78 | ge(4) | 90 |

OK Next-Page

Optimizing Row Creation

You can use the following MIB objects with the Monitor table to optimize row creation.

monUnusedIndex

Returns an unused index number for the Monitor table when you perform an SNMP Get on the variable.

monMatchDescr

Determines the index number that corresponds to a particular entry description. Perform an SNMP Set of this MIB object to cause the agent to search through entries in the Monitor table and put the index value of the last entry whose description matches in the monMatchIndex MIB object.

monMatchIndex

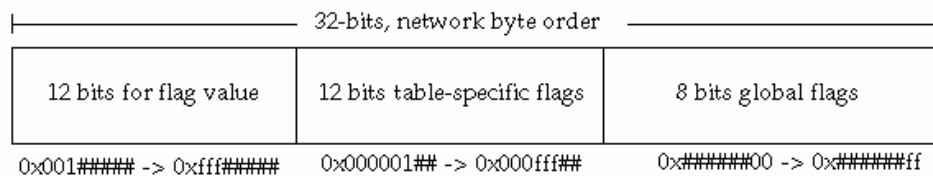
Matches a particular entry description with its index number when used with monMatchDescr.

Monitor Table Flags

The monFlags column in the Monitor table is a 32-bit unsigned integer that can specify additional behavioral semantics for the corresponding Monitor table row. By default, the Monitor table row does the following:

- Attempts to reinitialize itself
- Sends SNMP traps
- Logs events through the syslog facility
- Invokes actions (if they are configured)

You can set flag bits to alter these defaults. The CA eHealth SystemEDGE agent interprets all flags in hexadecimal (base 16) notation. The following illustration shows the composition of the Monitor Table flags field (monFlags).

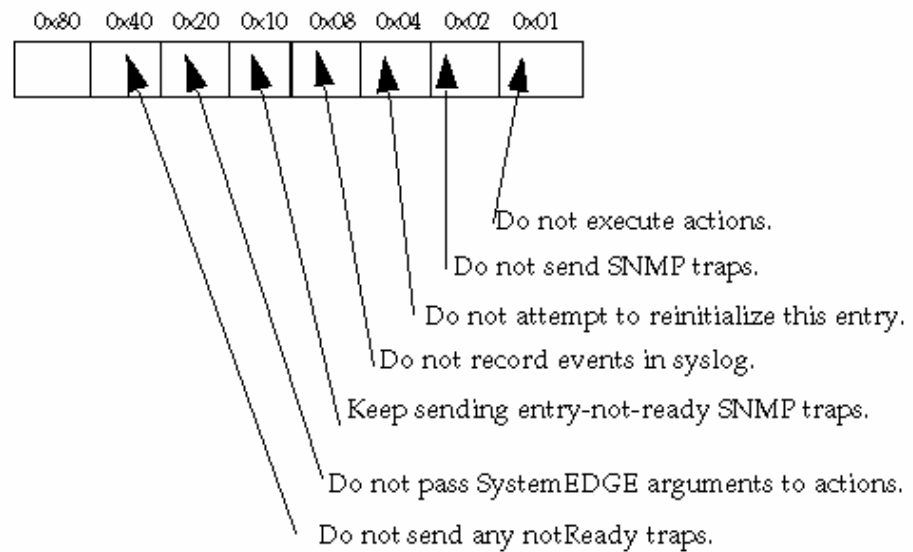


The flags value consists of three fields:

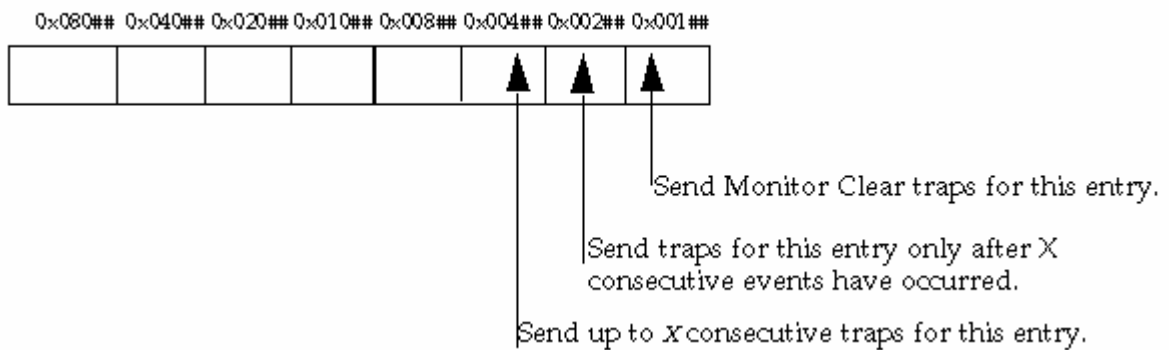
- **Field 1:** Common table flags defined for the self-monitoring tables of the Systems Management MIB. This portion is the low-order 8 bits of the flag.
- **Field 2:** Table-specific flags defined separately for each of the self-monitoring tables. This field defines the next 12 low-order bits after the common table flags.
- **Field 3:** Reserved 12 high-order bits for an integer value for use with table-specific flags. This field includes flags specific to the Monitor table.

The following sections explain each flag bit. You can combine these flag values through a logical OR operation.

The following illustration shows common flags for the monitoring tables:



The following illustration shows the flags specific to the Monitor table.



The following list describes the Monitor Table flags:

0x00000001

Disables running of actions for this entry.

0x00000002

Disables sending of SNMP traps for this entry. This flag bit overrides any other flag bit with respect to traps.

0x00000004

Disables attempts to reinitialize this entry. By default, the CA eHealth SystemEDGE agent periodically tries to reinitialize this entry by attempting to query the MIB object it is monitoring.

Note: Setting this bit disables automatic reinitialization.

0x00000008

Disables logging of events for this entry through the syslog facility. Setting this bit does not affect trap sending or threshold monitoring, but it does prevent the event from being logged through syslog. On Windows systems, the agent does not log the event in the agent's log file sysedge.log. Disabling event logging is useful when events occur frequently or when a particular entry is used as an agent heartbeat.

0x00000010

Sends continuous monitorEntryNotReady traps for this entry each time the agent attempts to reinitialize monitoring and fails to query the MIB object. The agent's default behavior is to send a single monitorEntryNotReady trap when a MIB object it is monitoring ceases to exist and then to attempt periodically to reinitialize the entry. Enabling this feature causes the agent to send an additional monitorEntryNotReady trap each time reinitialization fails.

0x00000020

Disables the passing of CA eHealth SystemEDGE arguments to action scripts or programs. CA eHealth SystemEDGE typically passes default action parameters that indicate the trap type, description field, and so on. This flag disables the passing of those arguments. For more information about action parameters, see Monitor Table Actions.

0x00000040

Disables sending of notReady traps for this entry.

0x00000100

Sends a monitorClear trap for this entry when the threshold monitor expression transitions from True to False.

0x00000200

Sends monitorEvent traps only on the *X*th consecutive event. After the *X*th event occurs, the agent sends monitor traps for each subsequent True expression evaluation. If the threshold expression transitions from True to False, the row resets itself, and the agent begins counting subsequent events at zero. This flag also applies to action execution. You can specify the value of *X* through the flag value field. Event logging is unaffected by this flag bit. For an example of this behavior, see Threshold Monitoring Examples in this chapter.

0x00000400

Sends up to *X* consecutive monitor traps, and then sends no more. Enabling this feature puts an upper boundary on the number of consecutive monitor traps and action executions that can occur when a threshold has been exceeded. After the threshold expression transitions from True to False, the row resets itself, and the agent begins counting subsequent events at zero. This flag also applies to action execution. You can specify the value of *X* through the flag value field. Event logging is unaffected by this flag bit. For an example of this behavior, see Threshold Monitoring Examples in this chapter.

0x###00000

Several flag bits use a value *X* for sending traps and executing actions. The value *X* is specified as the high-order 12 bits of the flag field. Flag bits utilizing this field are mutually exclusive. For an example of this behavior, see Monitor Table Flags.

Monitor Table Actions

The CA eHealth SystemEDGE agent provides several default parameters to the action commands when they are invoked. These parameters are in addition to any parameters you specify in the action string and they are passed on the command line *after* those that you specify. The default action parameters are the same as the parameters provided in the SNMP traps that the agent sends for the Monitor table.

The following list describes the default parameters for Monitor table actions:

trapType

Specifies the type of trap being sent. For example, monitorEvent or monitorEntryNotReadyEvent.

monDescr

Specifies the description from this table entry.

monOID

Specifies the Object Identifier from this table entry.

monCurrVal

Specifies the current value from this table entry.

monValue

Specifies the value being monitored from this table entry.

monRowStatus

Specifies the current row status for this table entry.

monOperator

Specifies the operator being used from this table entry. The value passed to the action script is the numeric representation of the actual operator. For example, if the operator is >, the value passed to the action script is 2.

monIndex

Specifies the index from this table entry.

monFlags

Specifies the flags associated with this table entry. The value passed to the action script is in base 16 (hexadecimal) notation with a leading 0x to indicate that it is a hexadecimal number (for example, 0x00000020).

Monitor Entry Correlation

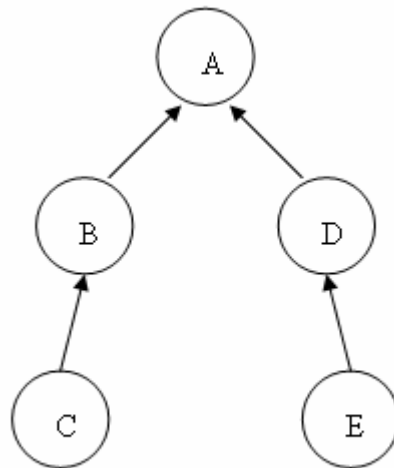
The CA eHealth SystemEDGE agent lets you correlate or associate several threshold monitoring entries into a single, multi-threshold monitor. This means that you can specify multiple independent thresholds and a parent-child tree structure for correlating them. You do this using the `monSupersededBy` column of the Monitor Entry Table. The column specifies which monitor entry actually overrides this entry's trap or event handling. Setting this value to zero (0) or to an invalid monitor entry index disables this functionality.

By combining this functionality with the monitor entry flags, you can minimize your trap throughput while still fully maintaining the state information you need.

As an example, assume you are tracking the CPU usage on a machine, and you want to set up LOW, WARNING, and CRITICAL level alarms based on the actual CPU usage. By correlating these two entries, you can suppress the LOW level alarms when the WARNING level alarm is active and suppress the WARNING level alarm when the CRITICAL level alarm is active. This has the effect of reducing your trap throughput while losing no fundamentally useful information.

You can expand this functionality, and use it across different metrics for the machine. For example, you might decide that when the machine Load reaches 10 or higher, all CPU and Load monitors should be disabled because this is useless information when the machine is in this state. You can configure the agent to disable all of these monitor entries by specifying the "Load > 10" entry as the parent to these other entries.

In the following diagram, entry **A** acts as the parent, and when its threshold is breached, it supersedes the threshold monitoring of entries **B** and **D**. Likewise, entry **B** supersedes entry **C**, and entry **D** supersedes entry **E**.



Note: For a specific example of this functionality, see Threshold Monitoring Examples in this chapter.

View the Monitor Table with CA eHealth AdvantEDGE View

Using CA eHealth AdvantEDGE View, you can query a system for Monitor table information. To do so, select the system you want to monitor from the System list, select Self Monitoring from the Configuration list, and then click the Configuration icon. For more information, see the CA eHealth AdvantEDGE View Web Help.

The following illustration shows a sample CA eHealth AdvantEDGE View Monitor table:

| Index | Description | Interval | Stype | OID | CurrVal | Operator | Compare | Last Call | Traps | Last Trap | Min | Max | Action | Flags | Superseded By | Row Status |
|-------------------|-----------------------------------|----------|-------------|-------------------|---------|----------|---------|-----------|-------|-----------|-----|-----|-------------|----------|---------------|------------|
| 11 | 1 minute load average | 60 | absolute(1) | loadAverage1Min | 11 | > | 300 | 0:11:41 | 0 | 0:00:00 | 11 | 58 | (no action) | 0x200508 | - | ● |
| 12 | 5 minute load average | 300 | absolute(1) | loadAverage5Min | 18 | > | 200 | 0:10:41 | 0 | 0:00:00 | 18 | 28 | (no action) | 0x200508 | - | ● |
| 13 | 15 minute load average | 900 | absolute(1) | loadAverage15Min | 20 | > | 200 | 0:00:38 | 0 | 0:00:00 | 20 | 20 | (no action) | 0x200508 | - | ● |
| 15 | Page-fault Rate | 60 | delta(2) | numPageFaults | 0 | > | 1000 | 0:00:00 | 0 | 0:00:00 | 0 | 0 | (no action) | 0x200508 | - | ● |
| 16 | le0 Incoming Packets | 60 | delta(2) | ifInUcastPkts.2 | 270 | > | 1000 | 0:11:41 | 0 | 0:00:00 | 183 | 401 | (no action) | 0x200508 | - | ● |
| 17 | le0 Outgoing Packets | 60 | delta(2) | ifOutUcastPkts.2 | 111 | > | 1000 | 0:11:41 | 0 | 0:00:00 | 40 | 143 | (no action) | 0x200508 | - | ● |
| 18 | SNMP Packets | 30 | delta(2) | snmpInPkts.0 | 0 | > | 4000 | 0:12:12 | 0 | 0:00:00 | 0 | 50 | (no action) | 0x200508 | - | ● |
| 21 | Num Processes | 60 | absolute(1) | hrSystemProcesses | 373 | > | 120 | 0:11:41 | 2 | 0:01:38 | 373 | 374 | (no action) | 0x200508 | - | ● |
| 22 | Warning: Swap utilization > 90% | 60 | absolute(1) | swapCapacity | 3 | >= | 90 | 0:11:41 | 0 | 0:00:00 | 3 | 3 | (no action) | 0x200508 | - | ● |
| 23 | Warning: Memory utilization > 85% | 60 | absolute(1) | memCapacity | 94 | >= | 85 | 0:11:41 | 2 | 0:01:38 | 94 | 94 | (no action) | 0x200508 | - | ● |
| 31 | CPU >20 | 60 | absolute(1) | cpu1Min | 14 | >= | 20 | 0:11:41 | 3 | 0:08:40 | 12 | 23 | (no action) | 0x21 | 32 | ● |
| 32 | CPU >40 | 60 | absolute(1) | cpu1Min | 14 | >= | 40 | 0:11:41 | 0 | 0:00:00 | 12 | 23 | (no action) | 0x21 | 33 | ● |
| 33 | CPU >60 | 60 | absolute(1) | cpu1Min | 14 | >= | 60 | 0:11:41 | 0 | 0:00:00 | 12 | 23 | (no action) | 0x21 | 34 | ● |
| 34 | CPU >80 | 60 | absolute(1) | cpu1Min | 14 | >= | 80 | 0:11:41 | 0 | 0:00:00 | 12 | 23 | (no action) | 0x21 | - | ● |
| 51 | / filesystem > 90 | 60 | absolute(1) | devCapacity.1 | 83 | >= | 90 | 0:11:41 | 0 | 0:00:00 | 83 | 83 | (no action) | 0x200508 | - | ● |
| Add Monitor Entry | | | | | | | | | | | | | | | | |

Assigning Entry Rows for the Monitor Table

The monIndex column of the Monitor table acts as a key field (or row index) to distinguish rows in the table. Rows 1 through 10 are reserved for internal use by the CA eHealth SystemEDGE agent. Users can configure rows in the range 11 to MAXINT. This section describes the benefits of reserving a block of rows for use by the system or application administrator.

Setting Local Policy

You may choose, as a matter of local policy, to reserve a block of rows for system administration. This policy lets you define entries within a reserved block of rows without being concerned that the row may already be taken by another user's entry. In compliance with the local policy, all other users should use row indices outside the reserved range when they define user-configured entries.

Reserve Blocks of Rows

By reserving a block of rows, you can define a consistent set of conditions (row entries) to be monitored across all computers such that the same condition is defined in the same row number on each computer. For example, you can use row 11 (`monIndex = 11`) to define an entry for monitoring the `swapCapacity` variable, and you can then distribute this configuration to every system so that row 11 is used to monitor the `swapCapacity` variable on every system.

To reserve a block of rows for threshold monitoring

1. Decide which block of rows you want to reserve for use.
2. Use that block of rows to define a set of row entries (conditions to be monitored) in the `sysedge.cf` initialization configuration file. For more information, see *Configuring the Monitor Table* in this chapter.

Note: You can also use this row number assignment policy with AdvantEDGE View for group-configuration operations.

3. Distribute the `sysedge.cf` file to all systems on which the CA eHealth SystemEDGE agent is installed.
4. Require users to avoid your block of rows when they define their own Monitor table entries.

Configuring the Monitor Table

You can control which MIB variables and conditions the CA eHealth SystemEDGE agent monitors (using its threshold monitoring capability) by adding, deleting, or modifying the entries in the Monitor table.

You can configure the Monitor table in the following ways:

- **Dynamically.** Use SNMP commands from a management system, such as AdvantEDGE View, to modify the table.

Note: For more information, see *Dynamic Configuration During Operation*.

- **At start-up initialization.** Specify the entries for the Monitor table in the `sysedge.cf` file that the agent reads on start-up.

Note: For more information, see *Initial Configuration During Setup*.

Initial Configuration During Startup

On startup, the agent reads the `sysedge.cf` file. You can use this file to specify which MIB variables you would like CA eHealth SystemEDGE to monitor by adding entries with the monitor configuration file keyword.

Note: For more information, see monitor Directive--Add Entries to the Monitor Table.

Select Objects for Monitoring

This section describes the process of specifying MIB objects for monitoring by adding them to the `sysedge.cf` file. As an alternative, you can specify MIB objects through SNMP Set operations.

Note: For more information, see Dynamic Configuration During Operation.

To specify MIB objects for monitoring

1. Select the SNMP object instance to be monitored. This object instance must be supported by the CA eHealth SystemEDGE agent and must be implemented on the platform on which the agent is running. For more information about object support, see the MIB specifications for your platform and the *CA eHealth SystemEDGE Release Notes*.

You can choose objects from the supported MIB modules: MIB-II, the Host Resources MIB, or the Systems Management MIB. (On Windows systems, you can select objects only from the Host Resources MIB and Systems Management MIB.)

Note: The object that you select must be of an integer-based type, such as integer, gauge, or counter. You can use textual conventions or enumerated types if they result in an integer ASN.1 value.

2. Assign the entry to a free row in the table by selecting the index number. The index number must be greater than 10, and must not yet be in use in the `sysedge.mon` file or by the agent.
3. Obtain the instance identifier for the object to be monitored.
4. Decide on the sample type. If the object you have selected is a counter, use `deltaValue`. For most other integer values (gauge, enumerated integer, integer, and so on), use `absoluteValue`.
5. Decide on the threshold and operator type against which the agent should compare the monitored variable's current value. The comparison expression that the agent uses is the following:

`current-value operator value`

Valid values for *operator* are described in the table in Columns of the Monitor Table. Choose an appropriate value for comparison. To help select an appropriate value, you can monitor the particular object for a period of time to see what a *normal* value is. *The choice of this value is critical and depends on the semantics of the object you are monitoring.* If you want to receive monitorClear traps, make sure you set the appropriate bit in the monFlags column.

6. Choose a monitor interval in seconds. The interval *must* be a multiple of 30 seconds. Choose the interval carefully. For example, you do not want the agent to sample so frequently that an operator cannot act on the condition being monitored if an exception occurs.
7. Choose the monitor options, and specify the appropriate flags. For more information, see Monitor Table Flags.
8. Add the entry to the sysedge.cf file, and then start the agent.

Dynamic Configuration During Operation

You can dynamically modify entries in the Monitor table by sending SNMP Set request messages from your NMS (including AdvantEDGE View) to the CA eHealth SystemEDGE agent. Each time an SNMP request successfully modifies the Monitor table, the agent updates the sysedge.mon file to record the changes. This file preserves changes made during the operation of the CA eHealth SystemEDGE agent across agent and system restarts, which means that if the agent is stopped, it can restart with the same Monitor table configuration.

Note: The CA eHealth SystemEDGE agent uses the SNMPv2 SMI textual convention for creating, deleting, and modifying rows in the self-monitoring tables.

Configuration file directives in sysedge.cf take precedence over entries in sysedge.mon. For example, if a Monitor table entry is contained in sysedge.mon at index 10, and a configuration file directive for index 10 of the Monitor table is added to sysedge.cf, the entry in sysedge.cf replaces the entry from sysedge.mon.

monitor Directive--Add Entries to the Monitor Table

Use the monitor keyword to add entries to the Monitor table as follows:

```
monitor attribute monIndex monFlags monInterval monSampleType monOperator  
monValue 'monDescr' 'monAction' monSupersededBy
```

attribute

Specifies the attribute to be monitored. This parameter can be one of the following:

oid variable name

Specifies the OID to be monitored. You can specify the OID using the complete dotted-decimal value (for example, 1.3.6.1.2.1.25.1.5.0) or the symbolic MIB name (for example, hrSystemNumUsers.0). Either way, *you must specify the object instance*, which is typically zero for non-tabular MIB variables.

filesystem 'filesystem-name' variable name

Specifies the name of a mounted file system that you want to monitor and the variable from the Systems Management MIB devTable that you want to monitor. For instance, you can specify filesystem /usr devCapacity.

monIndex

Specifies the row (index) of the Monitor table to use for this entry. Each row in the agent's Monitor table is uniquely identified by an index number. Rows 1 through 10 are reserved for internal use by the agent, so the monIndex value must be greater than 10.

monFlags

Contains hexadecimal flags (for example, 0x00000001) that specify any additional behavioral semantics for this entry.

monInterval

Specifies an integer value (30 to MAXINT) that indicates how often (in seconds) the monitoring should occur. For example, the value 30 instructs the agent to monitor this entry every 30 seconds.

Note: This value must be a multiple of 30 seconds.

monSampleType

Indicates whether the agent should sample the object's absolute value (absolute) or take the difference between successive samples (delta).

monOperator

Specifies the operator type, which is a Boolean used for evaluating an expression:

current-value operator value

The operator can be any of the following:

- nop (no operation)
- > (greater than)
- < (less than)
- >= (greater than or equal to)
- <= (less than or equal to)
- == (equal)
- != (not equal)

monValue

Specifies an integer value (threshold) to which the current value of the monitored MIB variable is compared.

monDescr

Specifies a quoted string, 0 to 128 characters in length, that typically contains a description of the object being monitored and a severity level.

monAction

Specifies a quoted string, 0 to 256 characters in length, that specifies the full path of the command (with any parameters) to be run when the expression evaluates to True and a trap is sent. If the string is empty, the agent performs no action for this entry.

monSupersededBy

Specifies the table index that takes precedence on this entry. The default is zero (0), indicating default behavior.

The CA eHealth SystemEDGE agent logs action-command invocations at syslog level LOG_DEBUG and action-command invocation errors at syslog level LOG_WARNING. For more information about configuring syslog, see the appendix "Using the syslog Facility." For more information about starting the agent with its debugging options turned on, see Configuring Support for Agent Debugging in the chapter "Configuring the CA eHealth SystemEDGE Agent".

Threshold Monitoring Examples

This section provides sample entries for the Monitor table to monitor thresholds on the target system. Each example shows how to define the entry and describes the condition being monitored. You can add these entries to sysedge.cf.

Example: Monitor the 1-Minute Load Average

The following examples configure the CA eHealth SystemEDGE agent to monitor the system's 1-minute load average:

```
(monitor oid 1.3.6.1.4.1.546.1.1.7.8.26.0 11 0x00 60 absolute > 300 'Monitor 1
minute load average' '' 0
monitor oid loadAverage1Min.0 11 0x00 60 absolute > 300 'Monitor 1 minute load
average' '' 0
```

1.3.6.1.4.1.546.1.1.7.8.26.0

Corresponds to the OID for the loadAverage1Minute variable contained within the Systems Management MIB.

11

Indicates that this entry will occupy row 11 (monIndex=11) in the Monitor table.

60

Specifies that the load average should be sampled once every 60 seconds.

absolute

Indicates that the agent should use the object's value, not the difference between successive samples.

300

Indicates the value against which the current load average is compared. If the currently sampled value is greater than (>) 300, an event occurs.

Note: The agent returns load averages as the underlying system's load average multiplied by 100. For example, a load average of 3 is returned as 300.

Example: Monitor the 5-Minute Load Average

The following example configures the CA eHealth SystemEDGE agent to monitor the system's 5-minute load average:

```
monitor oid loadAverage5Min.0 12 0x00500300 300 absolute > 200 'Monitor 5 minute  
load average' '' 0
```

loadAverage5Min.0

Corresponds to the OID for the loadAverage5Minute variable contained within the Systems Management MIB.

12

Indicates that this entry will occupy row 12 (monIndex=12) in the Monitor table.

0x00500300

- Configures the agent to send monitorClear traps when the threshold expression transitions from True to False.
- Configures the agent to begin sending traps (and run actions if they are configured) only at the Xth consecutive occurrence of the event.
- Specifies that X=5. That is, the agent begins to send traps at the fifth occurrence of the event.

300

Specifies that the load average should be sampled once every 300 seconds.

absolute

Indicates that the agent should use the object's value, not the difference between successive samples.

200

Indicates the value against which the current load average is compared. If the currently sampled value is greater than (>) 200, the agent sends a trap to all configured managers.

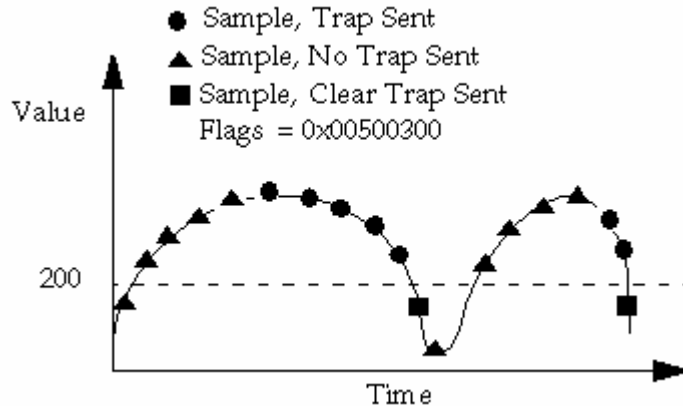
''

Indicates that no action is specified.

0

Indicates that default threshold event handling is specified.

The following illustration shows how the agent would send and clear traps based on this monitor directive.



Example: Monitor the 15-Minute Load Average

The following example configures the CA eHealth SystemEDGE agent to monitor the system's 15-minute load average:

```
monitor oid loadAverage15Min.0 13 0x0 900 absolute > 200 'Monitor 15 minute load average' '' 0
```

loadAverage15Min.0

Corresponds to the OID for the loadAverage15Min variable contained within the Systems Management MIB.

13

Indicates that this entry will occupy row 13 (monIndex=13) in the Monitor table.

900

Specifies that the load average should be sampled once every 900 seconds.

absolute

Indicates that the agent should use the object's value, not the difference between successive samples.

200

Indicates the value against which the current load average is compared. If the currently sampled value is greater than (>) 200, the agent sends a trap to all configured managers.

''

Indicates that no action is specified.

0

Indicates that default threshold event handling is specified.

Example: Monitor the System's Interrupt Rate

The following example configures the CA eHealth SystemEDGE agent to monitor the rate at which hardware interrupts are occurring on the local system:

```
monitor oid numInterrupts.0 14 0x00500400 60 delta > 1000 'Monitor Interrupt  
Rate' '' 0
```

numInterrupts.0

Corresponds to the OID for the numInterrupts counter object contained within the Systems Management MIB.

14

Indicates that the entry is index 14 in the Monitor table.

0x00500400

- Configures the agent to send a maximum of *X* consecutive traps when this monitor expression evaluates to True.
- Specifies that *X*=5. That is, the agent sends 5 consecutive monitorEvent traps, and then stops sending traps until the entry resets itself. The entry resets itself when the expression transitions from True to False.
- Does *not* specify that the agent should send monitorClear traps; consequently, a monitorClear trap is not sent when the expression transitions from True to False.

60

Indicates that the interrupt rate should be sampled every 60 seconds.

delta

Tells the agent to measure the rate at which the number of interrupts has changed. Because this object is a counter, delta is an appropriate sample type.

1000

Indicates the value against which the current number of interrupts is compared.

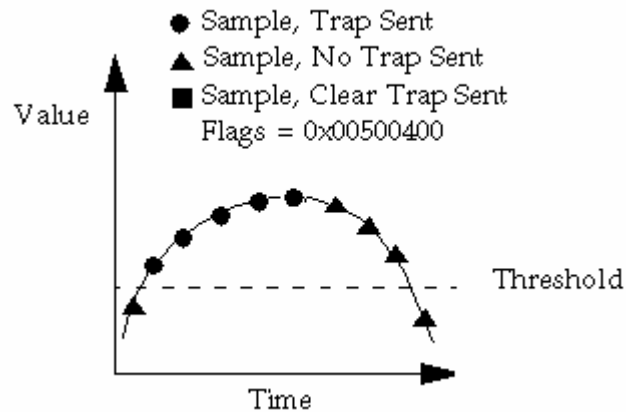
''

Indicates that no action is specified.

0

Indicates that default threshold event handling is specified.

The following illustration shows how the agent would send and clear traps based on this monitor directive.



Example: Monitor the System's Page-Fault Rate

The following example configures the CA eHealth SystemEDGE agent to monitor the rate at which hardware page-interrupts are occurring on the local system:

```
monitor oid numPageFaults.0 15 0x00500500 60 delta > 1000 'Monitor Page-fault Rate' '' 0
```

numPageFaults.0

Corresponds to the OID for the numPageFaults counter object contained within the Systems Management MIB.

15

Indicates that the entry is index 15 in the Monitor table.

0x00500500

- Configures the agent to send monitorClear Traps when the expression transitions from True to False.
- Specifies that the agent should send at most *X* consecutive monitorEvent traps, and then send no more until the entry resets itself.
- Specifies that *X*=5. That is, the agent sends 5 consecutive monitorEvent traps, and then stops sending traps until the entry resets itself.

60

Indicates that the interrupt rate should be sampled every 60 seconds.

delta

Tells the agent to measure the rate at which the number of interrupts has changed. Because this object is a counter, delta is an appropriate value for this entry's sample type.

1000

Indicates the value against which the current number of interrupts is compared.

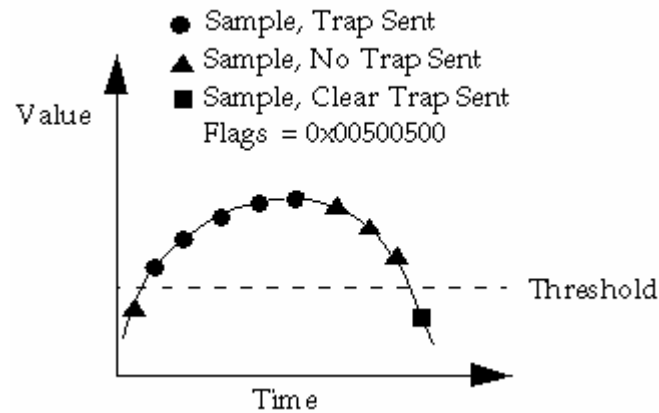
''

Indicates that no action is specified.

0

Indicates that default threshold event handling is specified.

The following illustration shows how the agent would send and clear traps based on this monitor directive:

**Example: Monitor Number of Incoming Packets on the Interface**

The following example configures the CA eHealth SystemEDGE agent to monitor the rate at which packets are received by the first ethernet interface (which is le0 for Sun systems):

```
monitor oid ifInUcastPkts.2 16 0x0 60 delta > 1000 'Monitor le0 Incoming Packets'
'' 0
```

ifInUcastPkts.2

Indicates the particular MIB object-instance to sample.

16

Indicates that the entry is index 16 in the Monitor Table.

60

Indicates that the agent should calculate the rate every 60 seconds.

delta

Tells the agent to measure the rate at which the number of incoming packets (ifInUcastPkts) is changing.

1000

Specifies the value to use in the comparison. If the change in rate is greater than (>) 1000, the agent sends a trap.

''

Indicates that no action is specified.

0

Indicates that default threshold event handling is specified.

Example: Monitor Number of Outgoing Packets on the Interface

The following example configures the CA eHealth SystemEDGE agent to monitor the rate at which packets are transmitted by the first ethernet interface (which is le0 for Sun systems):

```
monitor oid ifOutUcastPkts.2 17 0x0 60 delta > 1000 'Monitor le0 Outgoing  
Packets' '' 0
```

ifOutUcastPkts.2

Indicates the particular MIB object-instance to sample.

17

Indicates that the entry is index 17 in the Monitor table.

60

Indicates that the agent should calculate the rate every 60 seconds.

delta

Indicates the sample type because the object being monitored is a MIB-II ifEntry counter.

1000

Specifies the value to use in the comparison. If the change in rate is greater than (>) 1000, the agent sends a trap message to all configured managers.

''

Indicates that no action is specified.

0

Indicates that default threshold event handling is specified.

Example: Monitor Number of SNMP Packets Received

The following example configures the CA eHealth SystemEDGE agent to monitor the rate at which the agent receives SNMP requests:

```
monitor oid snmpInPkts.0 18 0x0 30 delta > 4000 'Monitor SNMP Packets' '' 0
```

snmpInPkts.0

Indicates the MIB II object-instance to sample.

18

Specifies that the entry is index 18 in the Monitor table.

30

Indicates that the agent should calculate the rate every 30 seconds.

delta

Indicates the sample type because the object is a counter.

4000

Specifies the value to use in the comparison. If the change in rate is greater than (>) 4000, the agent sends a trap message to all configured managers.

''

Indicates that no action is specified.

0

Indicates that default threshold event handling is specified.

Example: Monitor Space on the Root File System

The following example configures the CA eHealth SystemEDGE agent to monitor the root (/) file system and to send a trap message when it becomes more than 95% full:

```
monitor filesystem / devCapacity 19 0x0 120 absolute > 95 'Monitor / Filesystem' '' 0
```

devCapacity

Indicates the particular MIB object-instance to monitor; in this case, the object instance is devTableEntry.devCapacity from the Systems Management MIB devTable. The object instance is not specified because it is determined automatically based on the name of the file system.

19

Indicates that this entry is row 19 in the Monitor table.

120

Indicates that the agent should sample every 120 seconds.

absolute

Indicates the appropriate sample type because the agent is sampling an integer (not counter) value that represents how full the file system is.

95

Specifies the value to use in the comparison. If the file system becomes greater than (>) 95% full, the agent sends a trap message to all configured managers.

''

Indicates that no action is specified.

0

Indicates that default threshold event handling is specified.

Example: Monitor Space on the /usr File System

The following example configures the CA eHealth SystemEDGE agent to monitor the /usr file system and to send a trap message when it becomes more than 95% full:

```
monitor filesystem /usr devCapacity 20 0x00100500 120 absolute > 95 'Monitor /usr
Filesystem' '' 0
```

devCapacity

Indicates the particular MIB object-instance to monitor; in this case, the object instance is devTableEntry.devCapacity from the Systems Management MIB devTable. The object instance is not specified because it is determined automatically based on the name of the file system.

20

Indicates that this entry is row 20 of the Monitor table.

120

Indicates that the agent should sample every 120 seconds.

absolute

Indicates the appropriate sample type because the agent is sampling an integer value that represents how full the file system is.

95

Specifies the value to use in the comparison. If the file system becomes greater than (>) 95% full, the agent sends a trap message to all configured managers.

''

Indicates that no action is specified.

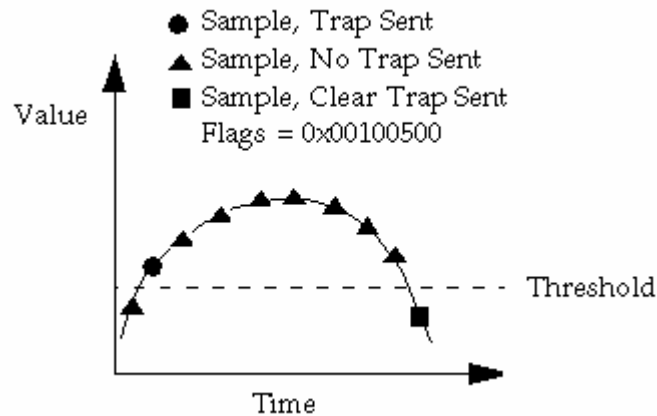
0

Indicates that default threshold event handling is specified.

0x00100500

- Configures the agent to send monitorClear traps when the expression transitions from True to False.
- Specifies that the agent should send at most X consecutive monitorEvent traps and then send no more until the entry resets itself.
- Specifies X=1, which indicates that the agent should send only one monitorEvent trap before waiting for the entry to reset itself.

The following illustration shows how the agent would send and clear traps based on this monitor directive:

**Example: Monitor the Number of Processes**

The following example configures the CA eHealth SystemEDGE agent to monitor the number of processes currently executing on the system and to send a trap when that number is greater than 120:

```
monitor oid hrSystemProcesses.0 21 0x0 60 absolute >= 120 'Monitor Number of Processes' '' 0
```

hrSystemProcesses

Indicates the variable to be monitored.

21

Indicates that this entry will be index 21 in the Monitor table.

60

Indicates that the agent should sample the number of processes every 60 seconds.

absolute

Indicates the sample type because the object is a gauge.

>=

Instructs the agent to send a trap whenever the number of processes is greater than or equal to 120.

..

Indicates that no action is specified.

0

Indicates that default threshold event handling is specified.

Example: Monitor the System's CPU Usage at Multiple Thresholds

The following example configures the CA eHealth SystemEDGE agent to monitor the rate of CPU usage of the system at different thresholds, with only a single event at any time. This example uses the `monSupersededBy` column.

Note: For more information, see Monitor Entry Correlation in this chapter.

```
monitor oid cpu1Min.0 31 0x0 60 absolute >= 20 'CPU >20' '' 32
```

```
monitor oid cpu1Min.0 32 0x0 60 absolute >= 50 'CPU >50' '' 33
```

```
monitor oid cpu1Min.0 33 0x0 60 absolute >= 80 'CPU >80' '' 0
```

cpu1Min.0

Corresponds to the OID for the one minute average CPU usage (in percent) in the Systems Management MIB.

31, 32, and 33

Indicates the entries in the Monitor table.

0x0

Configures the agent to run in default mode. This means that a trap is sent each scan when the monitor expression evaluates to True.

60

Indicates that the CPU usage should be sampled every 60 seconds.

absolute

Indicates the appropriate sample type because the agent is sampling an integer value that represents the percent CPU usage.

20, 50, and 80

Indicates the values against which the percent CPU usage is compared.

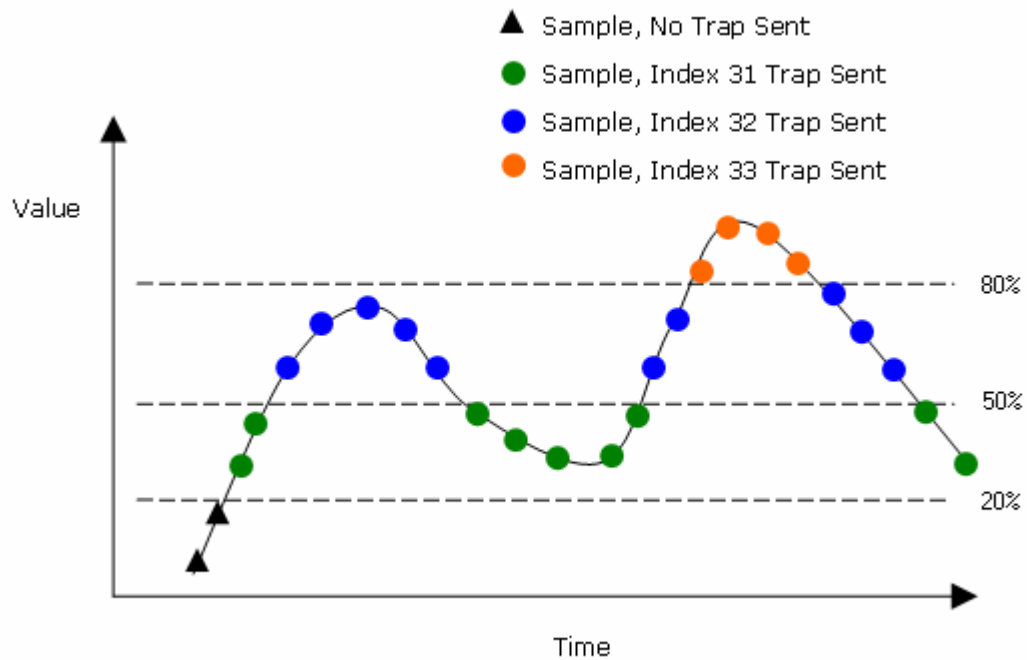
..

Indicates that no action is specified.

32, 33, and 0

Indicates that these three entries should act as a single, multi-threshold monitor, so that the lower CPU percent entries are superseded by the higher ones. For example, if the CPU usage was 56%, then entry index 31 ($\geq 20\%$) is superseded by entry index 32 ($\geq 50\%$). Entry index 34 ($\geq 80\%$) is not superseded by any other index.

The following illustration shows how the agent would send traps based on this monitor directive:



edgemon Utility--Monitor Thresholds

edgemon is a command-line utility that automatically configures the CA eHealth SystemEDGE agent to monitor a MIB variable that you specify. With this utility, you specify the following:

- MIB variable, either by name or by object-identifier
- Threshold value and comparison operator
- Flags
- Description
- (Optional) Action
- (Optional) Superseded By Index

The edgemon utility then issues an SNMP Set request to create the appropriate entry in the target agent's Monitor table.

Use the edgemon utility as follows:

```
edgemon [-h hostname | ip_addr] [-p port] [-c community]
        [-v 1 | 2c | 3] [-u secName] [-s secLevel] [-n contextName]
        [-a authPassword] [-A MD5 | SHA]
        [-x privPassword] [-X DES | AES | 3DES]
        [-m FIPS_mode]
        [-r retries]
        [-t timeout] [-d logLevel] [-f logFile]
        [-o] [command]
```

-h *hostname* | *ip_addr*

Specifies the hostname or IP address of the system on which the agent exists.

Default: localhost

-p *port*

Specifies the UDP port the agent runs on.

Default: 1691

-c *community*

Specifies the read-write community string that the agent runs with. This option is valid for SNMPv1 and SNMPv2 only.

Default: public

-v 1 | 2c | 3

Specifies the SNMP version that the agent is running.

-u *secName*

Specifies the USM (User-based Security Model) name for SNMP v3 security.

-s *secLevel*

Specifies one of the following security levels:

- 1 - noAuthNoPriv
- 2 - AuthNoPriv
- 3 - AuthPriv (SNMPv3 only)

-n *contextName*

Specifies the context name used by the agent if configured for SNMPv3.

Note: This option is not required for SNMPv3 communication.

-a *authPassword*

Specifies the authentication password if the agent is configured for SNMPv3 with secLevel 2 (AuthNoPriv) or secLevel 3 (AuthPriv).

-A MD5 | SHA

Indicates the authentication protocol if the agent is configured with Message Digest Algorithm (MD5) or SNMPv3 with secLevel 2 (AuthNoPriv) or secLevel 3 (AuthPriv).

-x *privPassword*

Specifies the privacy password if the agent is configured with SNMPv3 secLevel 3 (AuthPriv).

-X DES | AES | 3DES

Indicates the privacy protocol if the agent is configured for SNMPv3 with secLevel 3 (AuthPriv) and Data Encryption Standard (DES).

-m *FIPS_mode*

Controls the FIPS mode of operation. Accepted values are 0, 1, and 2.

0

Indicates Non-FIPS mode.

1

Indicates FIPS co-existence mode.

2

Indicates FIPS only mode.

-r *retries*

Specifies the number of retries.

Default: 3

-t *timeout*

Specifies the time duration before the SNMP receiver considers a request timed out.

Default: 10 seconds

-d *logLevel*

Indicates the debug level.

Default: 0

-f *logFile*

Specifies the log file that contains error and debug information.

-o *command*

Specifies the command and associated arguments. Supported commands include the following:

- oid (for monitoring an object)
- filesystem (for monitoring a file system)
- list (for listing the current Monitor table entries)
- setstatus (for setting the status of a Monitor table entry)
- delete (for deleting a Monitor table entry)

Note: For more information about supported commands, see edgemon Commands for Threshold Monitoring in this chapter.

Important! The following usage of the edgemon utility is deprecated:

edgemon *ipaddr[:port][,timeout] commstr [command]*

edgemon Commands for Threshold Monitoring

The edgemon command and associated arguments are as follows:

```
oid [object-instance] [index][flags] [interval] [sampleType] [operator] [value]
["descr"] ["action"] [supersededBy]
```

```
filesystem [filesystem-name] [variable-name] [index] [flags] [interval]
[sampleType] [operator] [value] ["descr"] ["action"] [supersededBy]
```

```
setstatus [index] [status]
```

```
delete [index]
```

```
list
```

Note: The arguments listed in brackets are updatable values. Arguments not listed in brackets are string literals and must be typed exactly as shown.

object-instance

Specifies the object-identifier or object-name to monitor. You can specify the OID using the complete dotted-decimal value (for example, 1.3.6.1.2.1.25.1.5.0) or the symbolic MIB name (for example, hrSystemNumUsers.0). Either way, you must specify the instance, which is typically 0 for non-tabular MIB variables. The object-instance or object-name must point to a MIB object that exists within the Systems Management MIB, the Host Resources MIB, or MIB-II. On Windows systems, this object instance or object name can only point to an object within the Systems Management MIB or Host Resources MIB.

index

Specifies the row (index) of the Monitor table to use for this monitoring entry. Each row in the agent's Monitor table is uniquely identified by an index number. Rows 1 through 10 are reserved for internal use by the agent, so the monIndex value must be greater than 10.

flags

Specifies the hexadecimal flags (for example, 0x00000001) that direct the additional behavioral semantics of this Monitor table entry. For a list of flags, see Process Monitor Table Flags in the chapter "Configuring Process and Service Monitoring".

interval

Specifies an integer value (30 to MAXINT) that indicates how often (in seconds) the monitoring should be performed. For example, the value 30 instructs the agent to monitor this entry every 30 seconds.

Note: This value must be a multiple of 30 seconds.

sampleType

Specifies sample type of either absolute or delta. This value indicates whether the agent should sample the object's absolute value or take the difference between successive samples.

oper

Specifies the boolean operator to use when comparing the sampled value to the threshold value. The operator can be one of the following:

- nop (no operation)
- > (greater than)
- < (less than)
- >= (greater than or equal to)
- <= (less than or equal to)
- == (equal)
- != (not equal)

value

Specifies the integer value (threshold) to which the current value of the monitored MIB variable is compared.

"descr"

Specifies a quoted string, 0 to 128 characters in length, that typically contains a description of the object being monitored and a severity level.

"action"

Specifies a quoted string, 0 to 256 characters in length, that specifies the full path of the command (with any parameters) to be run when the expression evaluates to True and a trap is sent. If the string is empty, no action is performed for this entry.

supersededBy

Specifies an index in this table that overrides or supersedes this entry. To disable this functionality, enter a value of zero (0).

Note: For more information about this option, see Monitor Entry Correlation in this chapter.

filesystem-name

Specifies the name of the mounted file system to monitor.

variable-name

Specifies the file system variable from the Systems Management MIB devTable to monitor. For example, you can monitor a file system's capacity by specifying devCapacity as the variable-name.

status

Specifies the entry's status, which can be one of the following:

- active (activate a row)
- notInService (deactivate but preserve a row)
- destroy (delete a Monitor table row)

edgemon Examples

This section provides examples of how to use the edgemon command.

Example: Monitor 1-Minute Load Average with edgemon

The following example creates an entry at index 11 in the agent's Monitor table that monitors the system's 1-minute load average for a threshold of 3. The examples below show IPv4 addresses, but you can specify IPv6 addresses as well.

```
edgemon -h fe80::901:dc19 -c private -v 1 -o oid loadAverage1Min.0 11 0x00 60  
absolute > 300 "Monitor 1 minute load average" "" 0
```

```
edgemon -h fe80::901:dc19 -c private -v 2c -o oid loadAverage1Min.0 11 0x00 60  
absolute > 300 "Monitor 1 minute load average" "" 0p
```

```
edgemon -h 143.45.0.12 -v 3 -u userName -s 3 -a authPassword -A MD5 -x  
encryptPassword -X DES -o oid loadAverage1Min.0 11 0x00 60 absolute > 300  
"Monitor 1 minute load average" "" 0
```

Deprecated:

```
edgemon 143.45.0.12 private oid loadAverage1Min.0 11 0x00 60 absolute > 300  
"Monitor 1 minute load average" "" 0
```

Example: Monitor Hardware Interrupts with edgemon

The following example creates a Monitor table entry to monitor the number of hardware interrupts on the underlying system against the threshold 1000. The examples below show IPv4 addresses, but you can specify IPv6 addresses as well.

```
edgemon -h fe80::901:dc19 -c private -v 1 -o oid numInterrupts.0 14 0x00500400 60  
delta > 1000 "Monitor Interrupt Rate" "" 0
```

```
edgemon -h fe80::901:dc19 -c private -v 2c -o oid numInterrupts.0 14 0x00500400  
60 delta > 1000 "Monitor Interrupt Rate" "" 0
```

```
edgemon -h 143.45.0.12 -v 3 -u userName -s 3 -a authPassword -A MD5 -x  
encryptPassword -X DES -o oid numInterrupts.0 14 0x00500400 60 delta > 1000  
"Monitor Interrupt Rate" "" 0
```

Deprecated:

```
edgemon 143.45.0.12 private oid numInterrupts.0 14 0x00500400 60 delta > 1000  
"Monitor Interrupt Rate" "" 0
```

The agent creates this entry at index 14 and samples the numInterrupts variable every 60 seconds. The flags field of 0x00500400 instructs the agent to modify the default Monitor table behavior as follows:

0x00000400

Instructs the agent to send up to X consecutive monitorEvent traps and then send no more.

0x00500000

Contains the flag value X=5 for use with the directive.

Example: Monitor the /usr File System with edgemon

The following example creates a Monitor table entry at index 20 to monitor the system's /usr file system for a capacity greater than or equal to 95%, checking the file system every two minutes (120 seconds). The examples below show IPv4 addresses, but you can specify IPv6 addresses as well.

```
edgemon -h 143.45.0.12 -c private -v 1 -o filesystem /usr devCapacity 20  
0x00100500 120 absolute >= 95 "Monitor /usr Filesystem" "" 0
```

```
edgemon -h 143.45.0.12 -c private -v 2c -o filesystem /usr devCapacity 20  
0x00100500 120 absolute >= 95 "Monitor /usr Filesystem" "" 0
```

```
edgemon -h fe80::901:dc19 -v 3 -u userName -s 3 -a authPassword -A MD5 -x  
encryptPassword -X DES -o filesystem /usr devCapacity 20 0x00100500 120 absolute  
>= 95 "Monitor /usr Filesystem" "" 0
```

Deprecated:

```
edgemon 143.45.0.12 private filesystem /usr devCapacity 20 0x00100500 120  
absolute >= 95 "Monitor /usr Filesystem" "" 0
```

Removing Threshold Monitoring Entries

To stop the threshold monitoring of a MIB variable, you must remove the appropriate entry from the Monitor table. This requires that you remove the entry from the both Monitor table and from the sysedge.cf file.

Monitor table entries are stored in the file sysedge.mon to make sure that they are not lost when the CA eHealth SystemEDGE agent is restarted. The monitor directives in the sysedge.cf file create a monitor entry in sysedge.mon whenever the CA eHealth SystemEDGE agent is started. For more information, see *Dynamic Configuration During Operation*.

Removing Entries from the sysedge.cf File

If you configured a Monitor table entry by adding a monitor directive to the sysedge.cf file, you must delete it manually from sysedge.cf. If you do not remove the sysedge.cf directive, the entry will be recreated in sysedge.mon the next time the CA eHealth SystemEDGE agent is restarted.

Removing Entries with the edgemon Utility

To remove a threshold-monitoring entry from the Monitor table, use the edgemon utility to delete the entry. The following example deletes row 14 from the Monitor table on host 143.45.0.12. After deletion, the row is removed from memory and from the sysedge.mon file.

```
edgemon -h fe80::901:dc19 -c private -v 1 -o delete 14
```

```
edgemon -h 143.45.0.12 -c private -v 2c -o delete 14
```

```
edgemon -h fe80::901:dc19 -c private -v 2c -o setstatus 14 destroy
```

```
edgemon -h fe80::901:dc19 -v 3 -u userName -s 3 -a authPassword -A MD5 -x  
encryptPassword -X DES -o setstatus 14 6
```

Note: The last argument, 6, indicates the set status 'destroy'.

Deprecated:

```
edgemon 143.45.0.12 private delete 14
```

Remove Entries Manually

In some cases, you may be unable to use the edgemon utility to delete Monitor table entries. For example, if you configured the CA eHealth SystemEDGE agent to disallow SNMP Set operations, the edgemon utility does not work. In this case, you must remove the threshold-monitoring entry from the Monitor table by editing the sysedge.mon. Because this is an active file, you must stop the CA eHealth SystemEDGE agent before you edit the file. For more information about the format of the sysedge.monfile, see the appendix "Adding Self-Monitoring Entries to sysedge.mon File".

To delete row 14 from the sysedge.mon file

1. Stop the CA eHealth SystemEDGE agent.
2. Open sysedge.mon for editing, delete the entry for monentry row 14, and save the file.

3. Open `sysedge.cf` for editing, delete the entry for monentry row 14 if it exists, and save the file.
4. Restart the CA eHealth SystemEDGE agent.

Chapter 10: Configuring Process and Service Monitoring

This chapter explains how to use the CA eHealth SystemEDGE agent to monitor processes and services. The CA eHealth SystemEDGE agent can monitor many attributes of a process, including whether it is running, its size, CPU and memory and usage, and the number of disk and network I/O operations.

This section contains the following topics:

[Monitoring Processes and Windows Services](#) (see page 211)

[The Process Monitor Table](#) (see page 213)

[Process Monitor Table Flags](#) (see page 220)

[Process Monitor Table Actions](#) (see page 226)

[View the Process Monitor Table with CA eHealth AdvantEDGE View](#) (see page 227)

[Assigning Entry Rows for the Process Monitor Table](#) (see page 228)

[Configuring the Process Monitor Table](#) (see page 228)

[edgewatch Utility--Monitor Processes](#) (see page 237)

[Removing Process Monitoring Entries](#) (see page 245)

[Recommendations for Process and Service Monitoring](#) (see page 247)

Monitoring Processes and Windows Services

The flexible Process Monitor table of the Systems Management MIB enables you to configure the agent dynamically to monitor specific attributes of important processes, services, and applications that are running on the underlying system. You specify the process, attribute, threshold value, and interval that you want to monitor. If a process attribute crosses the threshold you specified, the agent sends an SNMP trap to the management systems you have configured. The agent can also invoke an action command on the managed system to immediately correct the problem.

For information about monitoring process groups, see the chapter "Configuring Process Group Monitoring."

Monitoring Windows Services

On UNIX systems, services or daemons are processes, which can be monitored just like any other process. On Windows systems, however, services are special kinds of processes that are started and stopped using a graphical interface (for example, through the Services Control Panel). Windows services run within processes, but the mapping between them is not always one to one. For example, multiple Windows services may run *within* a single process. Consequently, you can monitor Windows services in two ways:

- Monitor Windows services by monitoring their underlying processes. It is not always easy to figure out the underlying process within which a service is running. In this case, you can have the CA eHealth SystemEDGE agent monitor the service itself rather than the underlying process.
- Instruct the CA eHealth SystemEDGE agent to monitor the Windows service, rather than the underlying process, by setting a flag or by using the configuration keyword watch ntsservice. For more information about Process Monitor table flags, see Process Monitor Table Flags.

Note: Because Windows does not track process attributes for Windows services, the CA eHealth SystemEDGE agent can monitor *only* the procAlive attribute for Windows services. To monitor other attributes (for example, procRSS) for a particular Windows service, you must monitor the underlying process that implements that Windows service.

Sample Process Monitor Table Entry

This section provides a sample Process Monitor table entry that instructs the CA eHealth SystemEDGE agent to monitor the httpd process to make sure it is up and running.

| Index | Description | Interval | Sample Type | Attribute | Current Value | Operator | Value | Last Call | #Traps | Last Trap | Flags | Row Status | Action | Reg Expr |
|-------|-----------------|----------|-------------|-----------|---------------|----------|-------|-------------|--------|-----------|------------|------------|--------|----------|
| 12 | "Monitor httpd" | 60 | absolute | procAlive | 3 | gt | 4 | 0d 22:40:00 | 0 | 0 | 0x00000100 | active | " | httpd |

This sample entry provides the following information:

- This entry is the 12th row in the Process Monitor table.
- It instructs the CA eHealth SystemEDGE agent to monitor the Web server daemon (httpd) every 60 seconds.
- The attribute being monitored is the status of the process (procAlive).
- The current value of 3 indicates the process is running normally (as far as the operating system is concerned). If the process status goes to 4, or if the process stops running, the agent will send a processStop trap to the management system.
- The RowStatus column shows that this entry is active.
- The Action column is null (""), which indicates that the agent invokes no action when the process stops running.
- The Flags value (0x00000100) instructs the agent to monitor the parent httpd daemon, rather than child processes. For more information about Process Monitor table flags, see Process Monitor Table Flags in this chapter.

The Process Monitor Table

The Process Monitor table provides information about each of the process/attribute pairs (or Windows service-status pairs) that the agent is currently monitoring. Each row in the table represents the combination of a process or Windows service and a particular attribute of that process or service that the agent is monitoring.

For each entry, the table provides the following types of information:

- Attribute being monitored
- Interval at which the agent checks the attribute
- Process that the agent is watching
- Number of traps that have been sent
- Current attribute value

Columns of the Process Monitor Table

The following list describes the columns of the Process Monitor table. For a complete description of the Process Monitor Table, see the Systems Management MIB specification (empire.asn1 in the /doc subdirectory of the CA eHealth SystemEDGE agent distribution).

pmonIndex

Specifies an integer (1 to MAXINT) that indicates the row index for this entry.

Permissions: Read-only

pmonDescr

Specifies a quoted string, 0 to 128 characters in length, that typically contains a description of the process and attribute that the agent is monitoring and a severity level.

Permissions: Read-write

pmonInterval

Specifies an integer value (30 to MAXINT) that indicates how often (in seconds) the agent should perform this monitoring. For example, the value 30 instructs the agent to monitor this entry every 30 seconds.

Note: This value must be a multiple of 30 seconds.

Permissions: Read-write

pmonSampleType

Indicates whether the agent should sample the attribute's absolute value (absoluteValue(1)) or take the difference between successive samples (deltaValue(2)). For example, use deltaValue to monitor counter attributes because it provides the rate of change. Use absoluteValue to monitor gauges, because it provides the object's exact value.

Permissions: Read-write

pmonAttribute

Specifies the process attribute being monitored. For a complete list of supported attributes, see Process Attributes in this chapter. For example, to monitor a process to verify that it is alive, specify the procAlive attribute. To track the number of packets received by the particular application or process, specify procMsgsSent.

Permissions: Read-write

pmonCurrVal

Specifies the attribute value that was last recorded for the process being monitored. Every *pmonInterval* seconds, the agent updates this field to reflect the latest reading for the attribute.

When monitoring `procAlive`, this value is mapped from the `hrSWRunStatus` variable of the Host Resources MIB. Possible values follow:

0

Indicates that the row is not ready. (If this is the case, the PID value will be -1)

1

Indicates that the process is running.

2

Indicates that the process is waiting for a resource (CPU, memory, or I/O).

3

Indicates that the process cannot be run. It is waiting for an event.

4

Indicates that the process is not loaded and is invalid.

Permissions: Read-only

pmonOperator

Specifies the operator type, which is a Boolean operator used for evaluating the following expression:

current-value operator value

The operator can be one of the following:

- `nop` (no operation; monitor the object's value, but do not evaluate the Boolean expression)
- `>` (greater than)
- `<` (less than)
- `>=` (greater than or equal to)
- `<=` (less than or equal to)
- `==` (equal)
- `!=` (not equal)

Permissions: Read-write

pmonValue

Specifies an integer value to which the current value of the monitored process attribute is compared during each monitoring cycle. If the comparison evaluates to True, the agent sends a trap. For example, if you want to be notified if the value of a gauge exceeds 100, set 100 as the `pmonValue` to which the agent compares the current value of the gauge.

Permissions: Read-write

pmonLastCall

Specifies the time (based on sysUpTime) at which the agent last sampled (called) the process attribute it is monitoring. 0 indicates that the MIB variable has not yet been sampled.

Permissions: Read-only

pmonNumTraps

Specifies the number of traps that have been sent for this entry. This column provides a useful metric for determining the frequency at which the exception condition is occurring and a means for detecting missed trap messages.

Permissions: Read-only

pmonLastTrap

Specifies the time (based on sysUpTime) at which the agent last sent a trap for this entry. 0 indicates that no traps have been sent.

Permissions: Read-only

pmonFlags

Specifies an integer flags value that indicates additional behavioral semantics this row should follow during the course of its operation. The default is 0x00. For more information about this field, see Process Monitor Table Flags in this chapter.

Permissions: Read-write

pmonAction

Specifies a quoted string, 0 to 256 characters in length, that specifies the full path of the command (with any parameters) to run when the expression evaluates to True and the agent sends a trap. If the string is empty, no action will be performed for this entry. By default, no action is performed.

Permissions: Read-write

pmonRegExpr

Specifies the regular expression to apply when the agent is attempting to acquire the process ID of an application or a process to monitor. For Windows service monitoring, this regular expression is used to match the name of the Windows service to monitor. Rather than requiring users to specify process IDs (PIDs) or service indexes, which may change, users specify a regular expression for process name or service name. The agent uses this user-specified name to find the process to monitor. By default, the Process Monitor table keeps attempting to apply the regular expression until a new PID or service is found if the process or service stops running.

Permissions: Read-write

pmonMinValue

Specifies the lowest (minimum) value that the agent has observed since it began polling the process attribute.

Permissions: Read-only

Default: 0

pmonMaxValue

Specifies the highest (maximum) value that the agent has observed since it began polling the process attribute.

Permissions: Read-only

Default: 0

pmonCurrentPID

Specifies the PID of the process/attribute pair currently being monitored.

Permissions: Read-write

pmonRowStatus

Specifies the row status, which can be one of the following:

- active
- notInService
- notReady
- createAndGo
- createAndWait

Typically, a row is either active or notInService. These values are defined in the SNMPv2 SMI RowStatus textual convention.

Permissions: Read-write

Process Attributes

The list that follows describes the attributes that the CA eHealth SystemEDGE agent can monitor for a particular process or service. The table also specifies the SNMP type for each attribute. You can use the SNMP type to select the sample type and operator. For example, absoluteValue sampling is usually most appropriate for attributes of type integer or gauge, while deltaValue sampling is usually most appropriate for attributes of type counter.

Note: The agent's ability to monitor a particular process attribute is dependent on the underlying operating system's ability to track the associated parameter or metric.

procAlive

Specifies whether the process or service is running.

Type: Boolean

procMEM

Specifies the percentage (0 to 100) of real memory used by this process.

Type: Gauge

procSize

Specifies the size of text, data, and stack segments (KB).

Type: Gauge

procRSS

Specifies the real memory (resident set) size of the process (KB).

Type: Gauge

procTime

Specifies the accumulated CPU time in seconds for this process.

Type: Integer

procInBlks

Specifies the number of blocks of data input by the process.

Type: Counter

procOutBlks

Specifies the number of blocks of data output by this process.

Type: Counter

procMsgsSent

Specifies the number of messages received by this process.

Type: Counter

procMsgsRecv

Specifies the number of messages sent by this process.

Type: Counter

procNice

Specifies the priority (nice value) of this process.

Type: Integer

procNumThreads

Specifies the number of threads that are running within this process.

Type: Integer

procNumSwaps

Specifies the number of times this process has been swapped.

Type: Counter

procSysCalls

Specifies the number of system calls invoked by this process.

Type: Counter

procMinorPgFlts

Specifies the number of minor page faults incurred by this process.

Type: Counter

procMajorPgFlts

Specifies the number of major page faults incurred by this process.

Type: Counter

procVolCtx

Specifies the number of voluntary context switches incurred by this process.

Type: Counter

procInvolCtx

Specifies the number of involuntary context switches incurred by this process.

Type: Counter

Optimizing Row Creation

You can use the following MIB objects with the Process Monitor table to optimize row creation.

pmonUnusedIndex

Returns an unused index number for the Process Monitor table when you perform an SNMP Get on the variable.

pmonMatchDescr

Determines the index number that corresponds to a particular entry description. Perform an SNMP Set of this MIB object to cause the agent to search through entries in the Process Monitor table and put the index value of the last entry whose description matches in the pmonMatchIndex MIB object.

pmonMatchIndex

Matches a particular entry description with its index number when used with pmonMatchDescr.

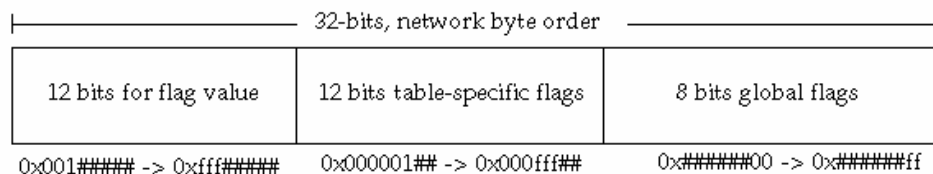
Process Monitor Table Flags

The pmonFlags column in the Process Monitor table is a 32-bit unsigned integer field that can specify additional behavioral semantics for the corresponding row.

By default, the Process Monitor table row does the following:

- Attempts to reinitialize itself
- Sends SNMP traps
- Logs syslog events
- Invokes actions (if they are configured)

You can set different flag bits to alter these defaults. The agent interprets all flags in hexadecimal (base 16) notation. The following illustration shows the composition of the Process Monitor table flags (pmonFlags) field.



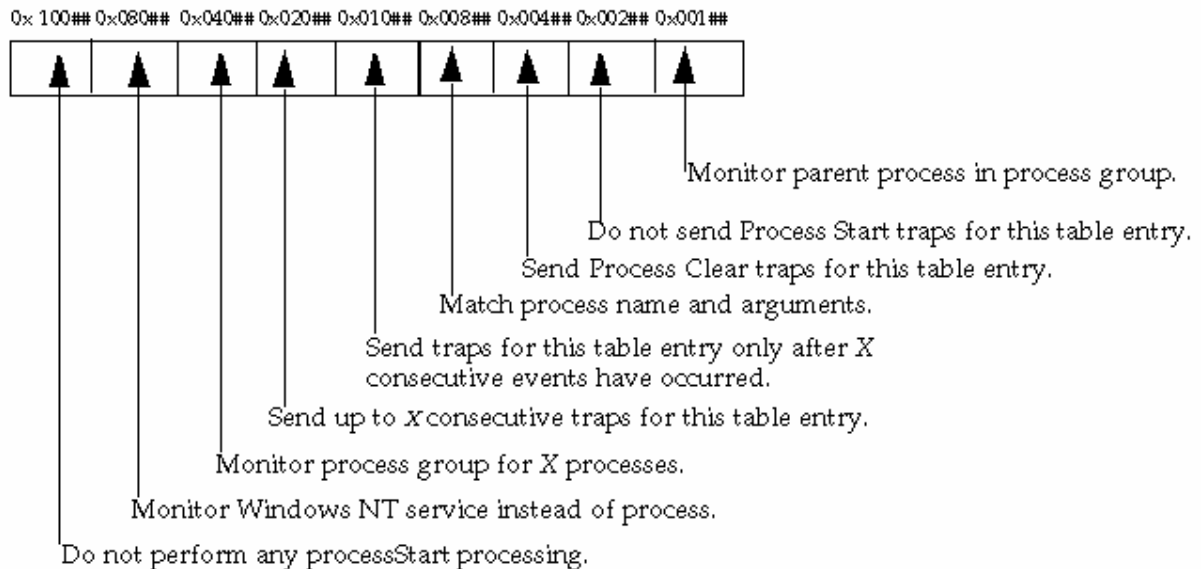
The flags value consists of three fields:

- Field 1: Common table flags for the self-monitoring tables of the Systems Management MIB. This portion is the low-order 8 bits of the flags field.
- Field 2: Table-specific flags that are defined separately for each of the self-monitoring tables. This field defines the next 12 low-order bits after the common table flags.

For more information about how these 12 bits are defined, see Process Monitor Table Flags in this chapter.

- Field 3: Reserved 12 high-order bits for an integer value for use with table-specific flags. This field defines the flags that are specific to the Process Monitor table.

The following sections explain each flag bit. You can combine flag values through a logical OR operation. The following illustration shows the flags that are specific to the Process Monitor table.



The following list describes the Process Monitor table flags:

0x00000001

Disables running of actions for this entry.

0x00000002

Disables sending of SNMP traps for this entry. This flag bit overrides any other flag bit with respect to traps.

0x00000004

Disables attempts to reinitialize this entry. By default, the agent periodically tries to reinitialize this entry by scanning the process table to determine the new process ID if the target process has been restarted. Setting this bit disables automatic reinitialization.

0x00000008

Disables logging of events for this entry through the syslog facility. Setting this bit does not affect trap sending or threshold monitoring, but it does prevent the event from being logged through syslog. On Windows systems, the agent does not log the event the agent's log file sysedge.log. Disabling event logging is useful when events occur frequently or when a particular entry is used as an agent heartbeat.

0x00000010

Sends continuous processStop traps for this entry each time the agent attempts to reinitialize process monitoring and fails to match a process. The agent's default behavior is to send a single processStop trap when a process dies and then attempt to periodically reinitialize the entry. Enabling this feature causes the agent to send an additional processStop trap each time reinitialization fails. In all cases, the agent does not send processStart and processStop traps unless the corresponding entry is monitoring the procAlive process attribute.

Note: This flag is valid only when the agent is monitoring the procAlive attribute. When you are monitoring Windows services, an entry does not enter the notReady state when the service is not running. Setting this flag causes CA eHealth SystemEDGE to generate processStop traps-and run any associated actions- for the entry when the service is not running, even though it remains in the ready state.

0x00000020

Disables the passing of CA eHealth SystemEDGE arguments to action scripts or programs. CA eHealth SystemEDGE typically passes default action parameters that indicate the trap type, description field, and so on. This flag disables the passing of those arguments. For more information about action parameters, see Process Monitor Table Actions.

0x00000040

Disables sending of notReady traps for this entry.

0x00000100

Monitors the parent process in the process group. Many applications and services (for example, Web server httpd daemons) are designed such that an initial daemon spawns child processes to handle actual service requests. These child processes often service several requests and then exit. In these cases, it is preferable to monitor the main parent daemon rather than the child processes. Enabling this feature causes the agent to search for and monitor the parent daemon rather than the first process it finds that matches the process regular expression.

The agent performs this search as follows: It scans the Process Monitor table for processes that match the name of the process regular expression (pmonRegExpr). If the matching parent process of the process also matches, the agent returns the parent process. This searching algorithm only accommodates parent/child relationships and cannot handle daemons forking daemons. This feature is not available on Windows systems because the notion of parent/child processes does not exist on Windows.

Note: You *cannot* set this flag if you are using the 0x00000800 flag.

0x00000200

Disables sending of processStart traps for this entry. If this feature is enabled, the agent sends processStop traps and logs events (unless those features were disabled through their corresponding flags bits).

0x00000400

Sends processClear traps for this entry when a process monitor expression transitions from True to False. This feature is only applicable when the attribute being monitored is not procAlive.

For more information, see Process Monitor Table Flags in this chapter.

0x00000800

Matches process name and arguments when targeting a process for monitoring. By default, the agent matches only against a process name. Enabling this option causes the agent to apply the pmonRegExpr to both the process name and process arguments, which is sometimes necessary to distinguish between similar processes or multiple invocations of the same application or binary.

Note: This flag is valid for UNIX systems only. You *cannot* set this flag if you are using the 0x00000100 flag.

0x00001000

Sends processThreshold traps only after the Xth consecutive event. Enabling this feature instructs the agent to wait until the Xth consecutive occurrence of an event before sending processThreshold traps. After the Xth event has occurred, the agent will send processThreshold traps for each subsequent, consecutive True expression evaluation. If the threshold expression transitions from True back to False, the row resets itself, and the agent begins counting events from zero. This flag also applies to action execution. You can specify the value of X through the flag value field. Event logging is unaffected by this flag bit. For an example, see Process Monitor Table Flags in this chapter.

0x10000

Disables any processStart processing. If this flag is enabled, CA eHealth SystemEDGE does not invoke actions, log events, or send traps when processStart events occur.

0x00002000

Sends up to X consecutive processThreshold traps, and then sends no more. Enabling this feature puts an upper boundary on the number of consecutive processThreshold traps and action executions that can occur when a process has exceeded a threshold. After the threshold expression transitions from True to False, the row resets itself, and the agent begins counting events from zero. This flag also applies to action execution. You can specify the value of X through the flag value field. Event logging is unaffected by this flag bit. For an example, see Process Monitor Table Flags in this chapter.

0x00004000

Monitors a process group for X processes.

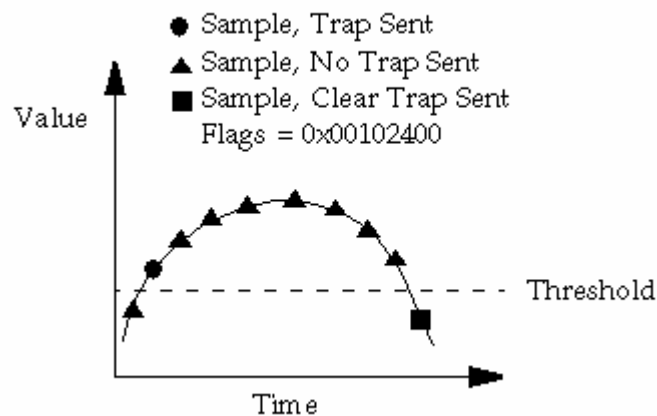
0x00008000

Monitors the Windows service that matches the corresponding regular expression. Setting this flag instructs CA eHealth SystemEDGE to monitor the procAlive attribute of the matching Windows service within the Windows service table. It is not necessary to set this flag bit if you use the watch ntsservice configuration file directive.

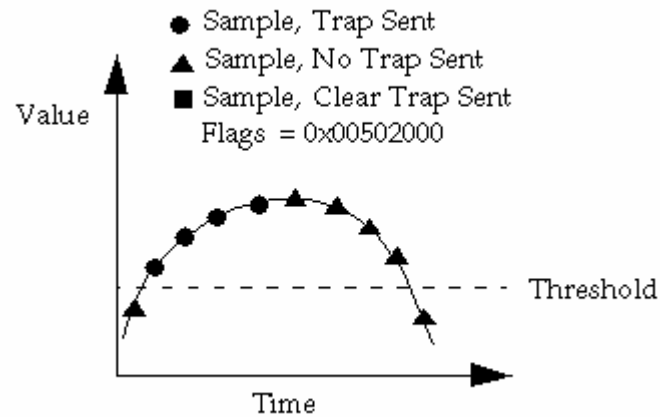
0x###00000

Several flag bits use a value X for sending traps and executing actions. The value X is specified as the high-order 12 bits of the flag field. Flag bits utilizing this field are mutually exclusive. For more information, see Process Monitor Table Flags in this chapter.

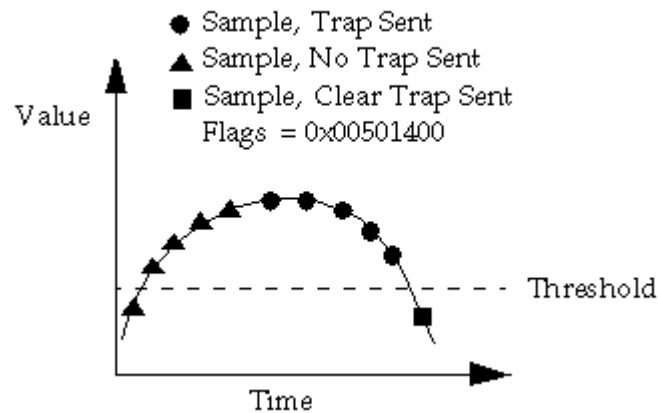
The following illustration shows the agent sending one trap to indicate that the monitored object crossed the threshold, and then sending a processClear trap when value drops below the threshold.



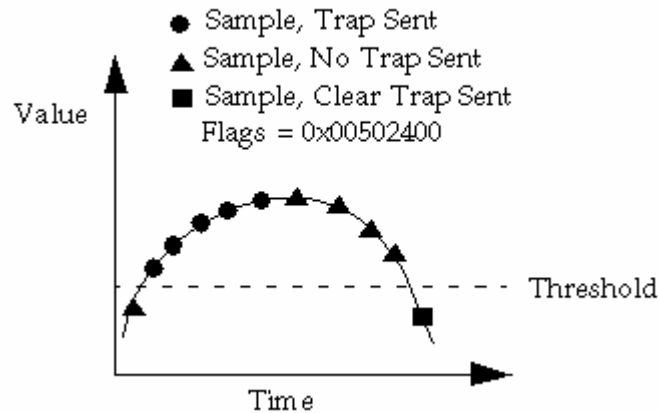
The following illustration shows the CA eHealth SystemEDGE agent sending four traps to indicate that the value of the monitored object is above the threshold. It does not send a processClear trap when the value falls below the threshold.



The following illustration shows the agent waiting until an event has occurred several times before it begins sending traps. It then sends traps until the value of the monitored object falls below the threshold, at which time it sends a processClear trap.



The following illustration shows the agent sending traps only a specified number of times. When the value of the monitored object falls below the threshold, the agent sends a processClear trap.



Process Monitor Table Actions

The CA eHealth SystemEDGE agent provides several default parameters to the action commands when they are invoked. These parameters are in addition to any parameters that you specify in the action string and are passed on the command line after those that you specify. The default parameters are the same as the parameters provided in the SNMP traps that are sent for the Monitor table. The following list describes the default parameters for Process Monitor table actions.

trapType

Specifies the type of trap being sent, which will be one of the following:

- processStop
- processStart
- processThreshold
- processClear

pmonIndex

Specifies the index assigned to this table entry.

pmonDescr

Specifies the table entry's description.

pmonAttribute

Specifies the process attribute being monitored by this entry.

pmonCurrVal

Specifies the current value obtained for this table entry.

pmonOperator

Specifies the operator being used by this table entry.

pmonValue

Specifies the comparison value or threshold applied by this table entry.

pmonFlags

Specifies flags that are associated with this table entry.

pmonRegExpr

Specifies the regular expression that the CA eHealth SystemEDGE agent uses to find the process ID of the process to monitor.




pmonCurrentPID

Specifies the process ID (PID) of the current process being monitored. If you are monitoring a Windows service (through the watch ntsservice directive or by setting flag 0x8000), this column represents the index in the NT Service MIB table that this process or service monitoring entry has acquired.

For more information about traps sent by the CA eHealth SystemEDGE agent, see the chapter "Private Enterprise Traps."

View the Process Monitor Table with CA eHealth AdvantEDGE View

If you are using CA eHealth AdvantEDGE View, you can query a system for Process Monitor table information by selecting the system you want to monitor from the System list, selecting Process Monitoring from the Configuration list, and clicking the Configuration icon. For more information, see the CA eHealth AdvantEDGE View Web Help. The following illustration shows a sample CA eHealth AdvantEDGE View Process Monitor table.

| Index | Description | Interval | Type | Attribute | Curr Val | Oper | Compare | Last Call | Traps | Last Trap | Flags | Action | RegExpr | Min | Max | PID | Row Status |
|---------------------------|--------------------------------|----------|----------|---------------|-----------------|------|------------|-------------------|-------|-----------|-------|-------------|---------|-----|-----|-----|---|
| 503 | Monitor ypbind (a group entry) | 60 | absolute | procAlive (1) | 0 | > | invalid(4) | 49 days, 16:53:20 | 0 | 0:00:00 | 0x1 | (no action) | ypbind | 0 | 0 | -1 |  |
| 1000 | Monitor inetd | 60 | absolute | procAlive (1) | notRunnable (3) | > | invalid(4) | 49 days, 16:53:20 | 0 | 0:00:00 | 0x100 | (no action) | inetd | 3 | 3 | 150 |  |
| 5000 | Group monitor inetd | 30 | absolute | procAlive (1) | notRunnable (3) | == | invalid(4) | 49 days, 16:53:50 | 0 | 0:00:00 | 0x100 | (no action) | inetd | 3 | 3 | 150 |  |
| Add Process Monitor Entry | | | | | | | | | | | | | | | | | |

Assigning Entry Rows for the Process Monitor Table

The pmonIndex column of the Process Monitor table acts as a row index to distinguish rows in the table. Rows 1 through 10 are reserved for internal use by the CA eHealth SystemEDGE agent. Users can configure rows in the range of 11 to MAXINT. For more information about reserving blocks of rows, see Reserve Blocks of Rows in the chapter “Configuring Threshold Monitoring.”

Configuring the Process Monitor Table

You can control the processes and process attributes that the CA eHealth SystemEDGE agent monitors by adding, deleting, or modifying the entries in the Process Monitor table.

You can configure the Process Monitor table in the following ways:

- Dynamically. Use SNMP commands from a management system, such as CA eHealth AdvantEDGE View, to modify the table. For more information, see the section Dynamic Configuration During Operation in this chapter.
- At start-up initialization. Specify the process attributes to monitor through the agent's configuration file sysedge.cf. For more information, see Initial Configuration During Startup in this chapter.

You can also dynamically add, delete, or modify Process Monitor Table entries through the edgewatch utility. For more information, see edgewatch Utility-- Monitor Processes in this chapter.

Dynamic Configuration During Operation

The CA eHealth SystemEDGE agent uses the SNMPv2 SMI Row Status textual convention for creating, deleting, and modifying rows in the table.

You can modify the entries in the Process Monitor table by issuing SNMP Set request messages from your NMS to the CA eHealth SystemEDGE agent. Each time the Process Monitor table is successfully modified, the agent updates the /etc/sysedge.mon file to record the changes. This enables the agent to start up with the same Process Monitor table configuration that it had when it was stopped. The sysedge.mon file preserves changes that are made during the operation of the agent across agent and system restarts. The agent *overwrites* the /etc/sysedge.mon or %SystemRoot%\system32\sysedge.mon configuration files every time the Process Monitor table is modified.

Note: Configuration file directives in `sysedge.cf` take precedence over entries in `sysedge.mon`. For example, if a Process Monitor table entry is in `sysedge.mon` at index 10, and a configuration file directive for index 10 of the Process Monitor table is added to `sysedge.cf`, the entry defined in `sysedge.cf` replaces the entry from `sysedge.mon`.

The CA eHealth SystemEDGE agent software distribution includes a command-line utility named `edgwatch` that takes a process name or PID as a command-line argument and dynamically configures an entry in the Process Monitor table to monitor the process. For instructions on how to use the `edgwatch` utility, see `edgwatch Utility--Monitor Processes` in this chapter.

Initial Configuration During Startup

On startup, the agent reads the `sysedge.cf` file. You can use this file to specify which MIB variables you want CA eHealth SystemEDGE to monitor. You can do so by adding `watch process` configuration file directives to the file. This directive automatically configures the agent to monitor an attribute of a process that you specify. You identify the process to be monitored through a regular expression that matches the process name and (optionally, for UNIX systems) its arguments. The agent automatically determines the PID for the specified process and then creates the appropriate entry in the agent's Process Monitor table. You need not know the PID ahead of time or to use `SNMP Set` requests to add an entry to the Process Monitor table to use the `watch process` directive. In addition, if the process is not yet running, the agent continues to try to match the regular expression to a process until it succeeds. Then, it initiates process monitoring.

Select Processes and Attributes for Monitoring

This section describes how to specify MIB objects for process monitoring by adding them to the `sysedge.cf` file. As an alternative, you can specify MIB objects through `SNMP Set` operations. For more information about specifying objects through `SNMP Set` operations, see `Dynamic Configuration During Operation` in this chapter.

To specify process attributes to monitor

1. Select the process and attribute to monitor. The process and attribute you select must be supported by the CA eHealth SystemEDGE agent and implemented on the platform on which the agent is running. For a list of supported variables, see the MIB specifications (in the `/doc` subdirectory of the CA eHealth SystemEDGE agent distribution).
2. Assign the entry to a free row in the table by selecting the index number.
3. Decide on the sample type: if the attribute you want to monitor is a counter, use `deltaValue`. For other integer values (gauge, enumerated integer, integer, and so on), use `absoluteValue`.

4. Decide on the threshold and operator type against which the agent should compare the monitored variable's current value. The agent uses the *current-value operator value* comparison expression for the agent.

Valid values for *operator* are in the section Columns of the Process Monitor in this chapter.

5. Select an appropriate value for comparison. To help determine an appropriate value, monitor the object for a period of time to find a normal value. *The choice of this value is critical and depends on the semantics of the object you are monitoring.* If you want to receive processClear traps, enable that feature through the pmonFlags field. For more information, see Columns of the Process Monitor Table in this chapter.
6. Select a monitor interval in seconds. The interval *must* be a multiple of 30 seconds. Select the interval carefully. For example, do not set the agent to sample so frequently that an operator does not have time to act on the monitored condition if an exception occurs.

Note: Alternatively, you could use a management station supporting the RowStatus operation to add the row via SNMP. For more information, see Dynamic Configuration During Operation in this chapter.

7. Write a configuration directive to define the process you have selected, and add it to the sysedge.cf file.
8. Start the CA eHealth SystemEDGE agent.

Note: If you are monitoring the procAlive process attribute, the CA eHealth SystemEDGE agent will automatically construct the appropriate Boolean expression.

Monitoring a Process to Make Sure It Is Running

You can use the procAlive process attribute to verify that a process is up and running. When it is not running, the CA eHealth SystemEDGE agent sends a processStop trap and invokes an action. When the process is restarted, the agent sends a processStart trap.

watch process procAlive Directive--Add Entries to Process Monitor Table

You can use the watch process procAlive directive to add entries to the Process Monitor table as follows:

```
watch process procAlive 'procname' index flags interv 'description' 'action'
```

'procname'

Specifies a quoted string that indicates the regular expression to apply when attempting to match a process name and optional arguments.

Note: Because the Windows kernel does not track the arguments used in a process, the CA eHealth SystemEDGE agent does not match process arguments for Windows systems.

index

Specifies the row (index) to use for this entry.

flags

Specifies any additional, non-default behavior to apply to this entry. Specify all flags as hexadecimal numbers (for example, 0x0000).

interv

Indicates how often (in seconds) the agent monitors the process.

'description'

Specifies a quoted string, 0 to 128 characters in length, that typically contains a description of the process and attribute that the agent is monitoring and a severity level for this event.

'action'

Specifies a quoted string, 0 to 256 characters in length, that specifies the full path of the command (with any parameters) to run when the process starts or stops. If the string is empty, the agent performs no action for this entry.

The watch process procAlive configuration file directive automatically creates the following Boolean expression for the entry:

```
hrSWRunStatus = 4
```

If the status of the process (as determined by the Host Resources Running Software table) equals invalid(4), or if the process stops running, the agent sends a processStop SNMP trap. If an action is configured, the agent also invokes the action. If the process restarts, the agent automatically detects the new PID, reinitializes the corresponding entry, and sends a processStart SNMP trap.

When CA eHealth SystemEDGE is monitoring Windows services directly (as enabled through the Process Monitor table flag 0x08000), the watch process procAlive configuration file directive automatically creates the following Boolean expression for the Process Monitor table entry:

```
ntServiceState ≠ 1
```

When the service's status (as determined by the NT Service table) becomes notRunning (its status does not equal 1), this expression evaluates to True, and the CA eHealth SystemEDGE agent sends a processStop SNMP trap. If an action is configured, the agent also invokes it. If the service restarts, the agent automatically detects that the service has restarted, and then reinitializes the corresponding Process Monitor Table entry and sends an processStart SNMP trap.

watch process Directive--Monitor Process Attributes

You can use the watch process directive to configure the CA eHealth SystemEDGE agent to monitor any attribute of a process other than its liveness (procAlive). Use the watch process directive as follows:

```
watch process attribute 'procname' index flags interval stype operator threshold  
'description' 'action'
```

attribute

Specifies the process attribute that the CA eHealth SystemEDGE agent monitors for the specified threshold. You may select any process attribute (except procAlive) from the section Process Attributes in this chapter.

'procname'

Specifies a quoted string that indicates the regular expression to apply when attempting to match a process name and optional arguments.

Note: Because the Windows kernel does not track the arguments used in a process, the CA eHealth SystemEDGE agent does not match process arguments for Windows systems.

index

Specifies the row (index) to use for this entry.

flags

Specifies any additional, non-default behavior to apply to this entry. Specify all flags as hexadecimal numbers (for example, 0x0000).

interv

Indicates how often (in seconds) the agent monitors the process.

stype

Indicates whether the agent should sample the process attribute's absolute value (absolute), or take the difference between successive samples (delta).

operator

Specifies the operator type, which is a Boolean operator used for evaluating the following expression:

current-value operator value

The operator can be one of the following:

- nop (no operation; monitor the object's value, but do not evaluate the Boolean expression)
- > (greater than)
- < (less than)
- >= (greater than or equal to)
- <= (less than or equal to)
- == (equal)
- != (not equal)

threshold

Specifies the integer value (threshold) to which the agent compares the current value (either absolute or delta).

'description'

Specifies a quoted string, 0 to 128 characters in length, that typically contains a description of the process and attribute that the agent is monitoring and a severity level.

'action'

Specifies a quoted string, 0 to 256 characters in length, that specifies the full path of the command (with any parameters) to run when the process starts or stops. If the string is empty, the agent performs no action for this entry.

The watch process configuration file directive automatically creates the following Boolean expression for the Process Monitor table entry:

attribute operator threshold

When this expression evaluates to True, the CA eHealth SystemEDGE agent sends a processThreshold SNMP trap. If you have configured an action, the agent invokes it. If the process has died, the Process Monitor table entry becomes notReady, and the agent sends a notReady SNMP trap. However, if the process restarts, the agent automatically reinitializes the corresponding Process Monitor table entry, reacquires the PID, and continues monitoring that process attribute for the specified threshold.

Process Monitoring Examples

This section contains sample configuration file directives for process monitoring. Each example shows how to define an instance of process monitoring and explains the attribute or threshold being monitored.

Example: Monitor Sendmail to Make Sure It Is Running

The following example configures the CA eHealth SystemEDGE agent to monitor the sendmail daemon on the underlying system:

```
watch process procAlive 'sendmail' 11 0x00000100 60 'Monitor sendmail' ''
```

11

Indicates that this entry will occupy row 11 (pmonIndex=11) in the Process Monitor table.

0x00000100

Indicates that the agent should monitor the parent sendmail process if more than one sendmail daemon is present and running.

60

Indicates that the agent should check the sendmail process every 60 seconds.

No action is specified, so the agent will not invoke a command when it sends a trap.

Example: Monitor the Simple TCP/IP Services Process To Make Sure It Is Running

The following example configures the CA eHealth SystemEDGE agent to monitor the TCPSVCS process that makes up the Simple TCP/IP Services service:

```
watch process procAlive 'TCPSVCS' 15 0x00000000 30 'Monitor NT TCP services' ''
```

15

Indicates that this entry will occupy row 15 (pmonIndex=15) in the Process Monitor table.

0x00000000

Indicates that the agent should provide the default process-monitoring behavior.

30

Indicates that the agent should check the TCPSVCS process every 30 seconds.

No action is specified, so the agent will not invoke a command when it sends a trap.

Note: This example illustrates how to monitor the underlying process that provides the Windows Simple TCP/IP Services service. The following example illustrates how to monitor the Windows service itself rather than its underlying process.

Example: Monitor the Simple TCP/IP Services Service

Both of the following examples configure the CA eHealth SystemEDGE agent to monitor the TCPSVCS service itself rather than the underlying process:

```
watch process procAlive 'Simple TCP/IP Services' 15 0x08000 30 'Monitor NT TCP/IP Services' ''
```

-or-

```
watch ntservice 'Simple TCP/IP Services' 15 0x0 30 'Monitor NT TCP/IP Services' ''
```

15

Indicates that this entry will occupy row 15 (pmonIndex=15) in the Process Monitor table.

0x08000

Indicates that the agent should monitor the Windows service, rather than the underlying process.

30

Indicates that the agent should check the Simple TCP/IP Services service every 30 seconds.

No action is specified, so the agent will not invoke a command when it sends a trap.

Example: Monitor ypbind To Make Sure It Is Running

The following example configures the CA eHealth SystemEDGE agent to monitor the UNIX ypbind daemon on the underlying system:

```
watch process procAlive 'ypbind' 16 0x00000000 60 'Monitor ypbind'
'/example/pager.sh'
```

16

Indicates that this entry will occupy row 16 (pmonIndex=16) in the Process Monitor table.

60

Indicates that the agent should check the ypbind process every 60 seconds.

The agent invokes the specified action script /example/pager.sh each time it sends a trap. In this case, it invokes the script each time a processStop or a processStart trap is sent. The script should examine its arguments to determine which trap is being sent and then send the appropriate message to the target pager.

Example: Monitor the Size of a Process

The following example configures the CA eHealth SystemEDGE agent to monitor the overall size of a particular process:

```
watch process procSize 'netscape' 20 0x00a02400 60 absolute '>' 35000 'Monitor
netscape size' ''
```

procSize

Indicates the attribute that the agent is monitoring. It returns the size of text, data, and stack segments of the corresponding process. Monitoring this attribute for a given threshold lets you to determine if it is leaking memory or growing unbounded.

netscape

Indicates the name of the process that the agent will monitor.

20

Indicates that this entry will occupy row 20 (pmonIndex=20) of the Process Monitor table.

0x00a02400

Instructs the agent to modify the default Process Monitor table behavior as follows:

0x00000400

Instructs the agent to send processClear traps.

0x00002000

Instructs the agent to send up to 10 consecutive traps and then send no more.

0x00a00000

Contains the flag value 10 for use with the directive in this example.

absolute

Instructs the agent to compare each sampled value to the threshold rather than to measure the difference (delta) between successive samples.

>

Instructs the agent to compare the sampled procSize attribute against the value 35000 (35,000 KB or 35 MB), and to send a processThreshold trap when that threshold is exceeded.

edgwatch Utility--Monitor Processes

edgwatch is a command-line utility that automatically configures the CA eHealth SystemEDGE agent to monitor processes, log files, and Windows event logs. After you specify the particular process, log file or Windows event log, and the associated arguments, the edgwatch utility issues an SNMP Set request to create the appropriate entry in the target agent's self-monitoring table.

You can use the edgwatch utility for process monitoring as follows:

```
edgwatch [-h hostname | ip_addr] [-p port] [-c community]

[-v 1 | 2c | 3] [-u secName] [-s secLevel] [-n contextName]
[-a authPassword] [-A MD5 | SHA]
[-x privPassword] [-X DES]
[-m FIPS_mode]
[-r retries]
[-t timeout] [-d logLevel] [-f logFile]
[-o] [facility] [command]
```

- h *hostname* | *ipaddr*

Specifies the hostname or IP address (in dotted notation) of the system on which the agent exists.

Default: localhost

-p *port*

Specifies the UDP port that the agent is running on (for example, 1691).

Default: 161

-c community

Specifies the community string that edgework uses in its SNMP requests to the agent. Valid on SNMPv1 and SNMPv2c.

Default: public

[-v 1 | 2c | 3]

Specifies the version of SNMP the agent is running. Specify 1 for SNMPv1, 2c for SNMPv2c, or 3 for SNMPv3.

Default: none

-u secName

Specifies the User-based Security Model (USM) user name used for SNMPv3 security.

Default: none

-s secLevel

Specifies the security level. Specify 1 for noAuthPriv, 2 for AuthNoPriv, or 3 AuthPriv.

Default: none

-n contextName

Specifies the context name the agent uses if configured as SNMPv3.

Note: This option is not required for SNMPv3 communication.

-a authPassword

Specifies the authentication password if the agent is configured with SNMPv3 with secLevel 2 (AuthNoPriv) or 3 (AuthPriv).

Note: This option is not required for SNMPv3 communication.

-A MD5 | SHA

Specifies the authentication protocol if the agent with SNMPv3 with secLevel 2 (AuthNoPriv) or 3 (AuthPriv). Specify MD5 for Message Digest Algorithm or SHA for Secure Hash Algorithm.

-x privPassword

Specifies the privacy password if the agent is configured for SNMPv3 with secLevel 3 (AuthPriv).

Default: none

-X DES

Specifies the privacy protocol if the agent is configured for SNMPv3 with secLevel 3 (AuthPriv) and Data Encryption Standard (DES).

Default: none

-m FIPS_mode

Controls the FIPS mode of operation. Accepted values are 0, 1, and 2.

0

Indicates Non-FIPS mode.

1

Indicates FIPS co-existence mode.

2

Indicates FIPS only mode.

-r *retries*

Specifies the number of retries.

Default: 3

-t *timeout*

Specifies the duration before the SNMP receiver considers a request timed out.

Default: 10 seconds

-d *logLevel*

Specifies the debug level.

Default: 0

-f *logfile*

Specifies the name of the log file that contains error or debug information.

Default: 0

-o *facility command*

Specifies the command and associated arguments. Supported commands include the following:

- add
- setstatus
- delete
- list

For more information about these commands, see the section *edgework Commands for Process Monitoring*.

The following usage of the *edgework* utility is deprecated:

```
edgework hostname[:port][,timeout] community process command
```

edgwatch Commands for Process Monitoring

The edgwatch process-monitoring commands and associated arguments are as follows:

```
add procAlive [processname] [index] [flags] [interval] ['description'] ['action']
```

```
add [attribute] [processname] [index] [flags] [interval] [sampleType] [operator]  
[value] ['description'] ['action']
```

```
setStatus [index] [status]
```

```
delete [index]
```

```
list
```

```
dump
```

Note: The arguments listed in brackets are updatable values. Arguments not listed in brackets are string literals and must be enter exactly as is, such as add, procAlive, setStatus, delete, list, and dump.

processname

Specifies the regular expression used to find the PID of the process to monitor. Enclose this value in quotation marks if it contains spaces or other special characters.

index

Specifies the row (index) to use for this monitoring entry.

flags

Specifies the hexadecimal flags (for example, 0x00000001) that indicate the additional behavioral semantics of this entry.

interval

Specifies an integer value (30 to MAXINT) that indicates how often (in seconds) the agent should monitor the process. For example, the value 30 instructs the agent to monitor this entry every 30 seconds.

Note: This value must be a multiple of 30 seconds.

sampleType

Specifies the type of sampling. Valid values are absolute and delta.

absolute

Indicates absolute value sampling. The agent uses the sampled value when evaluating the monitor table's boolean expression.

delta

Indicates delta value sampling. The agent samples two values and uses their difference when evaluating the monitor table's boolean expression. Delta value sampling is most often used when monitoring counter based objects, because it is the rate of change rather than the absolute value.

operator

Specifies the boolean operator to use when comparing the sampled value to the threshold value. The operator can be one of the following:

- nop (no operation)
- > (greater than)
- < (less than)
- >= (greater than or equal to)
- <= (less than or equal to)
- == (equal)
- != (not equal)

value

Specifies a threshold value for the attribute.

'description'

Specifies a quoted string, 0 to 128 characters in length, that typically contains a description of the process and attribute that are being monitored and a severity level for this event.

'action'

Specifies a quoted string, 0 to 256 characters in length, that specifies the full path of the command (with any parameters) to run when the expression evaluates to True and a trap is sent. If the string is empty, the agent invokes no action for this entry.

status

Specifies the status of the entry, which can be one of the following:

- active (activate a row)
- notInService (deactivate but preserve a row)
- destroy (delete a row)

attribute

Specifies the process attribute that the agent monitors for the given threshold. You may select any process attribute from the table in the section The Process Monitor Table, Process Attributes of the chapter "Configuring Process and Service Monitoring." The arguments for procAlive differ from those for the other attributes. For more information, see watch process Directive--Add Entries to Process Monitor Table in this chapter.

edgework Examples

This section includes examples for using edgework.

Example: Monitor the ypbind Process

The following example creates a Process Monitor table entry at index 16 to monitor the ypbind process running on the target system for SNMPv1:

```
edgework -h 143.45.0.12 -c private -v 1 -o process add procAlive "ypbind" 16 0x00 60 "Monitor ypbind" "/example/pager.sh"
```

The following example creates a Process Monitor table entry at index 16 to monitor the ypbind process running on the target system for SNMPv2c:

```
edgework -h 143.45.0.12 -c private -v 2c -o process add procAlive "ypbind" 16 0x00 60 "Monitor ypbind" "/example/pager.sh"
```

The following example creates a Process Monitor table entry at index 16 to monitor the ypbind process running on the target system for SNMPv3:

```
edgework -h 143.45.0.12 -v 3 -u userName -s 3 -a authPassword -A MD5 -x encryptPassword -X DES -o process add procAlive "ypbind" 16 0x00 60 "Monitor ypbind" "/example/pager.sh"
```

The following *deprecated* example creates a Process Monitor table entry at index 16 to monitor the ypbind process running on the target system:

```
edgework 143.45.0.12 private process add procAlive "ypbind" 16 0x00 60 "Monitor ypbind" "/example/pager.sh"
```

procAlive

Indicates the process attribute being monitored. It instructs the agent to monitor the process to make sure it is running. ypbind is the process that the agent is monitoring. It is responsible for client directory lookups and is necessary for computers running Network Information Services (NIS).

0x00

Instructs the agent to provide the default behavior for this table entry.

If the process dies, the agent sends a processStop trap and runs the action script /example/pager.sh.

Example: Monitor the netscape Process

The following example creates a Process Monitor table entry at index 20 to monitor the netscape process running on the target computer for SNMPv1:

```
edgework -h 143.45.0.12 -c private -v 1 -o process add procSize "netscape" 20 0x00a02400 60 absolute ">" 35000 "Monitor netscape size" ""
```

The following example creates a Process Monitor table entry at index 20 to monitor the netscape process running on the target computer for SNMPv2c:

```
edgework -h 143.45.0.12 -c private -v 2c -o process add procSize "netscape" 20 0x00a02400 60 absolute ">" 35000 "Monitor netscape size" ""
```

The following example creates a Process Monitor table entry at index 20 to monitor the netscape process running on the target computer for SNMPv3:

```
edgework -h 143.45.0.12 -v 3 -u userName -s 3 -a authPassword -A MD5 -x encryptPassword -X DES -o process add procSize "netscape" 20 0x00a02400 60 absolute ">" 35000 "Monitor netscape size" ""
```

The following *deprecated* example creates a Process Monitor table entry at index 20 to monitor the netscape process running on the target computer for SNMPv1:

```
edgework 143.45.0.12 private process add procSize "netscape" 20 0x00a02400 60 absolute ">" 35000 "Monitor netscape size" ""
```

procSize

Indicates the process attribute being monitored. It instructs the agent to monitor the overall size of the program's text, data, and stack segments.

netscape

Indicates the application that the agent is monitoring.

0x00a02400

Instructs the agent to modify the default Process Monitor table behavior as follows:

0x00000400

Instructs the agent to send processClear traps.

0x00002000

Instructs the agent to send up to 10 consecutive traps and then send no more.

0x00a00000

Contains the flag value 10 for use with this directive.

>

Indicates that an event should occur when the process size of netscape exceeds the threshold (35 MB).

35,000 KB or 35 MB

Indicates the threshold.

Example: Monitor the Windows TCPSVCS Process

The following example creates a Process Monitor table entry at index 15 to monitor the Windows *TCPSVCS* process (or service) running on the target system for SNMPv1:

```
edgework -h 143.45.0.12 -c private -v 1 -o process add procAlive "TCPSVCS" 15
0x00 30 "Monitor NT TCP services" ""
```

The following example creates a Process Monitor table entry at index 15 to monitor the Windows *TCPSVCS* process (or service) running on the target system for SNMPv2c:

```
edgework -h 143.45.0.12 -c private -v 2c -o process add procAlive "TCPSVCS" 15
0x00 30 "Monitor NT TCP services" ""
```

The following example creates a Process Monitor table entry at index 15 to monitor the Windows *TCPSVCS* process (or service) running on the target system for SNMPv3:

```
edgework -h 143.45.0.12 -v 3 -u userName -s 3 -a authPassword -A MD5 -x
encryptPassword -X DES -o process add procAlive "TCPSVCS" 15 0x00 30 "Monitor NT
TCP services" ""
```

The following *deprecated* example creates a Process Monitor table entry at index 15 to monitor the Windows *TCPSVCS* process (or service) running on the target system:

```
edgework 143.45.0.12 private process add procAlive "TCPSVCS" 15 0x00 30 "Monitor
NT TCP services" ""
```

procAlive

Indicates the process attribute being monitored. It instructs the agent to scan the Process Monitor table periodically (every 30 seconds) to verify that this process is running.

TCPSVCS

Indicates the Windows service responsible for TCP-related services on Windows systems.

0x00

Instructs the agent to provide the default behavior for this table entry.

Example: Display all of the processes on a system

The following example displays (dumps) all processes on the local host that is running sysedge on port 1691:

```
edgewatch -c private -p 1691 -v 1 -o process dump
```

Removing Process Monitoring Entries

To stop the self-monitoring of a particular process attribute, you must remove the appropriate entry from the Process Monitor table. The watch process directives in the sysedge.cf file creates a Process Monitor table entry whenever the CA eHealth SystemEDGE agent starts. This row creation results in a new Process Monitor table entry stored in sysedge.mon. Permanent removal of a Process Monitor table entry requires two steps:

1. Remove the entry from the sysedge.cf file.
2. Remove the entry from the Process Monitor table.

Removing Entries from the sysedge.cf File

If you configured a Process Monitor table entry by adding a watch process directive to the sysedge.cf file, you must remove it from that file as part of removing the entry from the table. If you do not remove the sysedge.cf directive, the entry will be recreated the next time the CA eHealth SystemEDGE agent starts.

Removing Entries with the edgemon Utility

To remove a process-monitoring entry from the Process Monitor table, use the edgemon utility to delete the entry. The examples below show IPv4 addresses, but you can specify IPv6 addresses as well.

The following example deletes row 14 from the Process Monitor table on host 143.45.0.12. After deletion, the row will be removed both from memory and from the sysedge.mon file for SNMPv1:

```
edgemon -h 143.45.0.12 -c private -v 1 -o process delete 14
```

The following example deletes row 14 from the Process Monitor table on host 143.45.0.12. After deletion, the row will be removed both from memory and from the sysedge.mon file for SNMPv2c:

```
edgemon -h 143.45.0.12 -c private -v 2c -o process setstatus 14 6
```

The following example deletes row 14 from the Process Monitor table on host 143.45.0.12. After deletion, the row will be removed both from memory and from the sysedge.mon file for SNMPv3:

```
edgemon -h 143.45.0.12 -v 3 -u userName -s 3 -a authPassword -A MD5 -x  
encryptPassword -X DES -o process setstatus 14 destroy
```

The following *deprecated* example deletes row 14 from the Process Monitor table on host 143.45.0.12. After deletion, the row will be removed both from memory and from the sysedge.mon file:

```
edgemon 143.45.0.12 private process delete 14
```

Remove Entries Manually

In some cases it may not be possible to use the edgemon utility to delete Process Monitor table entries. For example, if you have configured the CA eHealth SystemEDGE agent to disallow SNMP Set operations, the edgemon utility does not work. In this case, you must remove the entry from the Process Monitor table by editing the sysedge.mon file to remove the entry. Because this is an active file, you must stop the CA eHealth SystemEDGE agent before editing the file. For more information about the format of this file, see the appendix "Adding Self-Monitoring Entries to the sysedge.mon File."

To delete row 14 manually

1. Stop the CA eHealth SystemEDGE agent.
2. Open sysedge.mon for editing, delete the entry for processmon row 14, and save the file.
3. Open sysedge.cf for editing, delete the entry for processmon row 14 if it exists, and save the file.
4. Restart the CA eHealth SystemEDGE agent.

Recommendations for Process and Service Monitoring

When you are configuring process and service monitoring, consider the following:

- Monitor the following:
 - Status of the processes:
 - Use the processState (operating-system dependent) or processStateStr (operating-system independent) OIDs to return the process state. For more information, see `empire.asn1`.
 - For processes that run through an interpreter such as Perl, set the 0x00000800 flag to match the process name and arguments (UNIX only) when creating process-monitoring entries. For more information, see Process Monitor Table Flags in this chapter.
 - CPU time over interval (if this value is not incrementing, the process might be in the zombie state.)
 - Total CPU time (a high value can indicate problems)
 - Leaking memory (RSS over time; use alarm thresholds to notify you of problems)
- Configure CA eHealth SystemEDGE to automatically restart failed processes.
- Set your UNIX application shutdown files to disable process and service monitoring to prevent race conditions.
- Use the Process Group Monitor table to monitor multiple processes with the same name. For more information, see the chapter “Configuring Process and Service Monitoring.”

Chapter 11: Configuring Process Group Monitoring

This chapter explains how to use the CA eHealth SystemEDGE agent to monitor groups of processes. The agent uses process groups to aggregate the per-process information into a single, easy-to-poll, easy-to-monitor value.

This section contains the following topics:

[Monitoring Process Groups](#) (see page 249)

[The Process Group Monitor Table](#) (see page 250)

[Process Group Monitor Table Flags](#) (see page 255)

[Process Group Monitor Table Actions](#) (see page 257)

[View the Process Group Monitor Table with CA eHealth AdvantEDGE View](#) (see page 257)

[Assigning Entry Rows for the Process Group Monitor Table](#) (see page 258)

[Configuring the Process Group Monitor Table](#) (see page 258)

[watch procgroup Directive--Monitor a Process Group](#) (see page 259)

[Removing Process Group Monitoring Entries](#) (see page 261)

Monitoring Process Groups

The flexible Process Group Monitor table of the Systems Management MIB lets you configure the CA eHealth SystemEDGE agent dynamically to monitor groups of processes running on the underlying system. You select the process group, regular expression, and interval, and the agent uses that information to monitor those process groups. You can configure the CA eHealth SystemEDGE agent to monitor process groups to determine what processes exist in each group and whether the group membership changes. If components of an application start or fail, or if members leave a group or are added to a group, the CA eHealth SystemEDGE agent can automatically notify the NMS.

Note: For information about monitoring individual processes and Windows services, see the chapter “Configuring Process and Service Monitoring.”

The Process Group Monitor Table

The Process Group Monitor table provides information about a group of processes that the agent is currently monitoring. You can add entries to this table, modify existing entries, or remove entries from the table. The CA eHealth SystemEDGE agent sends a processGroupChange trap to indicate when a process group changes, and you can configure the agent to perform actions whenever a group changes.

Columns of the Process Group Monitor Table

The list that follows describes the columns that make up the Process Group Monitor table. For a complete description of the Process Group Monitor table and its fields, see the Systems Management MIB specification (empire.asn1 in the doc subdirectory of the CA eHealth SystemEDGE agent's installation).

Note: CA eHealth SystemEDGE maintains a history of group membership and tracks process statistics even after an individual process has left the group.

pgmonIndex

Specifies an integer (1 to MAXINT) that indicates the row index for this entry.

Permissions: Read-only

pgmonDescr

Specifies a quoted string, 0 to 128 characters in length, that typically contains a description of the entry and who created it.

Permissions: Read-write

pgmonInterval

Specifies an integer value (1 to MAXINT) that indicates how often (in seconds) the agent should sample the variable. For example, the value 30 instructs the agent to monitor this entry every 30 seconds.

Note: This value must be a multiple of 30 seconds.

Permissions: Read-write

pgmonProcRegExpr

Specifies the regular expression to apply when the agent is attempting to match processes by name.

Permissions: Read-write

pgmonFlags

Specifies the integer flags value that dictates the behavior of each entry. The default is 0x00. For more information about this field, see Process Group Monitor Table Flags in this chapter.

Permissions: Read-write

pgmonNumProcs

Specifies the current number of processes in the process group that this entry is tracking. A process belongs to the process group if its name (and possibly, its arguments) matches the regular expression configured for this entry.

Permissions: Read-only

pgmonPIDList

Lists the numeric PIDs in this process group. Each PID is separated from the next by a space character.

Permissions: Read-only

pgmonStatusList

Lists the process status in this process group. Each process state is separated from the next with a space character. Entries in the status list have a one-to-one correspondence with entries in the PID list. For more information about states of a process, see the processStateStr MIB variable.

Permissions: Read-only

pgmonAction

Specifies a quoted string, 0 to 256 characters in length, that specifies the full path of the command (with any parameters) to run when the expression evaluates to True. If the string is empty, the agent invokes no action for this entry. By default, it invokes no action.

Permissions: Read-write

pgmonNumEvents

Specifies the number of events for this entry. Events do not necessarily imply traps; traps can be turned off for the row through a flags setting.

Permissions: Read-only

pgmonNumTraps

Specifies the number of traps that the agent has sent for this entry.

Permissions: Read-only

pgmonLastTrap

Specifies the time (based on sysUpTime) at which the agent last sent a trap for this entry. A value of 0 indicates that no traps have been sent.

Permissions: Read-only

pgmonRowStatus

Specifies the row status, which can be one of the following:

- active(1)
- notInService(2)
- notReady(3)
- createAndGo(4)
- createAndWait(5)
- destroy(6)

Typically, a row is either active or notInService. These values are identical in meaning to the SNMPv2 SMI RowStatus textual convention.

Permissions: Read-write

pgmonRSS

Specifies the combined resident set size (RSS) of the group of processes. The RSS for each process in the group is summed at each interval and stored in this variable. Because RSS also includes shared memory, the total RSS for a group could exceed total possible physical memory for the underlying system. For more information, see the processRSS variable of the Systems Management MIB (in empire.asn1).

Permissions: Read-only

pgmonSize

Specifies the combined size of the text, data, and stack segments of the group of processes. The size of each process in the group is summed at each interval and stored in this variable. Size includes shared memory, so the total size for a group could exceed the total virtual memory for the underlying system. For more information, see the processSize variable of the Systems Management MIB (in empire.asn1).

Permissions: Read-only

pgmonThreadCount

Specifies the total number of threads for the group of processes. The number of threads running in each process in the group is summed at each interval and stored in this MIB variable. For more information, see the processNumThreads variable of the Systems Management MIB (in empire.asn1).

Permissions: Read-only

pgmonMEM

Specifies the total percentage of real memory being used by the processes in this group. The percentage of memory being used is summed at each interval and stored in this MIB variable. Memory usage includes shared memory (shared libraries and DLLs), so the total percentage may exceed 100.

Permissions: Read-only

pgmonInBlks

Specifies the number of blocks of data input by processes in this group.

Permissions: Read-only

pgmonOutBlks

Specifies the number of blocks of data output by processes in this group.

Permissions: Read-only

pgmonMsgsSent

Specifies the number of messages sent by processes in this group.

Permissions: Read-only

pgmonMsgsRecv

Specifies the number of messages received by processes in this group.

Permissions: Read-only

pgmonSysCalls

Specifies the number of system calls invoked by processes in this group.

Permissions: Read-only

pgmonMinorPgFlts

Specifies the number of minor page faults incurred by processes in this group.

Permissions: Read-only

pgmonMajorPgFlts

Specifies the number of major page faults incurred by processes in this group.

Permissions: Read-only

pgmonNumSwaps

Specifies the number of times processes in this group have been swapped.

Permissions: Read-only

pgmonVolCtx

Specifies the number of voluntary context switches incurred by processes in this group.

Permissions: Read-only

pgmonInvolCtx

Specifies the number of involuntary context switches incurred by processes in this group.

Permissions: Read-only

pgmonCPUSecs

Specifies the number of seconds of CPU time used by processes in this group.

Permissions: Read-only

pgmonMatchUser

Matches running processes by user name and any process name regular expression when set to a valid user name. This variable is valid only on UNIX systems.

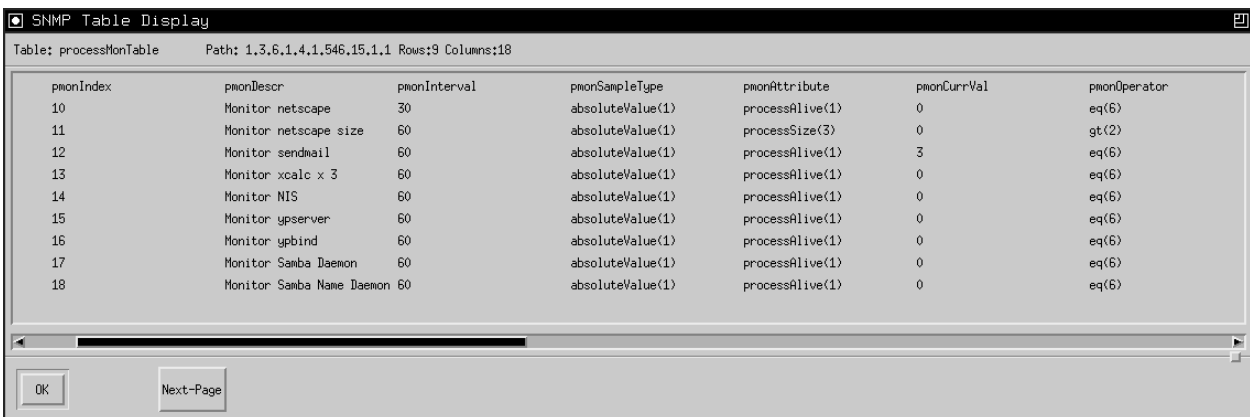
Permissions: Read-write

pgmonMatchGroup

Matches running processes by group name, process name regular expression, and user name when set to a valid group name. This variable is valid only on UNIX systems.

Permissions: Read-write

The following illustration shows a sample Process Group Monitor table:



The image shows a screenshot of a software window titled "SNMP Table Display". Inside the window, there is a table with 7 columns: pmonIndex, pmonDescr, pmonInterval, pmonSampleType, pmonAttribute, pmonCurrVal, and pmonOperator. The table contains 9 rows of data, representing various system monitors. Below the table, there are two buttons: "OK" and "Next-Page".

| pmonIndex | pmonDescr | pmonInterval | pmonSampleType | pmonAttribute | pmonCurrVal | pmonOperator |
|-----------|---------------------------|--------------|------------------|-----------------|-------------|--------------|
| 10 | Monitor netscape | 30 | absoluteValue(1) | processAlive(1) | 0 | eq(6) |
| 11 | Monitor netscape size | 60 | absoluteValue(1) | processSize(3) | 0 | gt(2) |
| 12 | Monitor sendmail | 60 | absoluteValue(1) | processAlive(1) | 3 | eq(6) |
| 13 | Monitor xcalc x 3 | 60 | absoluteValue(1) | processAlive(1) | 0 | eq(6) |
| 14 | Monitor NIS | 60 | absoluteValue(1) | processAlive(1) | 0 | eq(6) |
| 15 | Monitor ypserver | 60 | absoluteValue(1) | processAlive(1) | 0 | eq(6) |
| 16 | Monitor ypbind | 60 | absoluteValue(1) | processAlive(1) | 0 | eq(6) |
| 17 | Monitor Samba Daemon | 60 | absoluteValue(1) | processAlive(1) | 0 | eq(6) |
| 18 | Monitor Samba Name Daemon | 60 | absoluteValue(1) | processAlive(1) | 0 | eq(6) |

Optimizing Row Creation

You can use the following MIB objects with the Process Group Monitor table to optimize row creation:

pgmonUnusedIndex

Returns an unused index number for the Process Group Monitor table when you perform an SNMP Get on the variable.

pgmonMatchDescr

Determines the index number that corresponds to a particular entry description. Perform an SNMP Set of this MIB object to cause the agent to search through entries in the Process Group Monitor table and put the index value of the last entry whose description matches in the pgmonMatchIndex MIB object.

pgmonMatchIndex

Matches a particular entry description with its index number when used with pgmonMatchDescr.

Process Group Monitor Table Flags

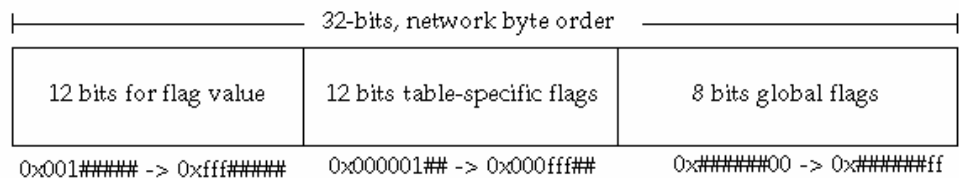
The pgmonFlags column in the Process Group Monitor table is a 32-bit unsigned integer field that can specify additional behavioral semantics for the corresponding row.

By default, the Process Group Monitor table row does the following:

- Attempts to reinitialize itself
- Sends SNMP traps
- Logs syslog events
- Invokes actions (if they are configured)

You can set different flag bits to alter these defaults. The agent interprets all flags in hexadecimal (base 16) notation.

The following illustration shows the composition of the Process Group Monitor Table flags (pgmonFlags) field:



The flags value consists of three fields:

- **Field 1:** Common table flags that are defined for the self-monitoring tables of the Systems Management MIB. This portion is the low-order 8 bits of the flags field.
- **Field 2:** Table-specific flags that are defined separately for each of the self-monitoring tables. This field defines the next 12 low-order bits after the common table flags.
- **Field 3:** Reserved 12 high-order bits for an integer value for use with table-specific flags. Flags in the Process Group Monitor table-specific flags field define the contents of this field.

The following sections explain each flag bit. You can combine flag values through a logical OR operation. One flag is specific to the Process Group Monitor table: 0x00100. This flag instructs the CA eHealth SystemEDGE agent to match the process name and arguments for this entry. The following list describes the Process Monitor table flags:

0x00000001

Disables running of actions for this entry.

0x00000002

Disables sending of SNMP traps for this entry. This flag bit overrides any other flag bit with respect to traps.

0x00000004

Disables attempts to reinitialize this entry. By default, the agent periodically tries to reinitialize this entry by scanning the process table to determine the new process ID if the target process has been restarted. Setting this bit disables automatic reinitialization.

0x00000008

Disables logging of events for this entry through the syslog facility. Setting this bit does not affect trap sending or threshold monitoring, but it does prevent the event from being logged through syslog. On Windows systems, the agent does not log the event in the agent's log file sysedge.log. Disabling event logging is useful when events occur frequently or when a particular entry is used as an agent heartbeat.

0x00000020

Disables the passing of CA eHealth SystemEDGE arguments to action scripts or programs. CA eHealth SystemEDGE typically passes default action parameters that indicate the trap type, description field, and so on. This flag disables the passing of those arguments. For more information about action parameters, see Process Group Monitor Table Actions.

0x00000040

Disables sending of notReady traps for this entry.

0x00000100

Matches the process name and arguments for this entry.

Process Group Monitor Table Actions

The CA eHealth SystemEDGE agent provides several default parameters to the action commands when they are invoked. These parameters are in addition to any parameters that you specify in the action string and are passed on the command line *after* those that you specify. The default parameters are the same as the parameters provided in the SNMP traps that are sent for the Monitor table. For more information about traps sent by the CA eHealth SystemEDGE agent, see the chapter "Private Enterprise Traps."

View the Process Group Monitor Table with CA eHealth AdvantEDGE View

If you are using CA eHealth AdvantEDGE View, you can query a system for Process Group Monitor table information by selecting the system you want to monitor from the System list, selecting Process Group Monitoring from the Configuration list, and clicking the Configuration icon. For more information, see the CA eHealth AdvantEDGE View Web Help.

The following illustration shows a sample CA eHealth AdvantEDGE View Process Group Monitor table:

| Index | Description | Interval | RegExpr | User | Group | Flags | Action | Row Status | Num Procs |
|-----------|---------------------|----------|---------|-------|-------|-------|-------------|------------|-----------|
| <u>9</u> | Monitor NFS Server | 60 | nfs | admin | (any) | 0x10b | (no action) | | <u>2</u> |
| <u>10</u> | Monitor Web Server | 60 | http | (any) | (any) | 0xa | (no action) | | <u>0</u> |
| <u>11</u> | Monitor SSH Server | 30 | sshd | (any) | (any) | 0x0 | (no action) | | <u>0</u> |
| <u>12</u> | Monitor FTP Server | 30 | ftpd | admin | (any) | 0x0 | (no action) | | <u>0</u> |
| <u>13</u> | Monitor all | 60 | .* | (any) | (any) | 0xb | (no action) | | <u>35</u> |
| <u>14</u> | Monitor RPC daemons | 60 | rpc | (any) | (any) | 0xa | (no action) | | <u>1</u> |

Add Entry

Assigning Entry Rows for the Process Group Monitor Table

The pgmonIndex column of the Process Group Monitor table acts as a key field (or row index) to distinguish rows in the table. Rows 1 through 10 are reserved for internal use by the CA eHealth SystemEDGE agent. Users can configure rows in the range of 11 to MAXINT. For more information about reserving blocks of rows, see Reserve Blocks of Rows in the chapter "Configuring Threshold Monitoring."

Configuring the Process Group Monitor Table

You can control which processes and process attributes the CA eHealth SystemEDGE agent monitors by adding, deleting, or modifying the entries in the Process Group Monitor table. You can configure the Process Group Monitor table in one of these ways:

- Dynamically. Use SNMP commands from a management system, such as CA eHealth AdvantEDGE View, to modify the table. For more information, see Dynamic Configuration During Operation in this chapter.
- At start-up initialization. Specify the process attributes to monitor through the agent's configuration file sysedge.cf. For more information, see Initial Configuration During Startup.

Dynamic Configuration During Operation

You can use your NMS platform to issue SNMP Set request messages to the CA eHealth SystemEDGE agent to modify the entries in the Process Group Monitor table. Each time a Set request successfully modifies the table, the agent updates the sysedge.mon file to record the changes so that the agent starts up with the same Process Group Monitor table configuration it had when it was stopped. That is, the agent *overwrites* the /etc/sysedge.mon or %SystemRoot%\system32\sysedge.mon configuration files every time the Process Group Monitor table is modified. Any changes made during the operation of the agent are preserved in this file across agent and system restarts.

Note: The CA eHealth SystemEDGE agent uses the SNMPv2 SMI Row Status textual convention for creating, deleting, and modifying rows in the table.

Configuration file directives in sysedge.cf take precedence over entries in sysedge.mon. If, for example, a Process Group Monitor table entry is in sysedge.mon at index 10, and a configuration file directive is added to sysedge.cf at index 10 for the Process Group Monitor table, the entry defined in sysedge.cf replaces the entry in sysedge.mon.

Initial Configuration During Startup

On startup, the agent reads the sysedge.cf file. You can use this file to specify which MIB variables you would like CA eHealth SystemEDGE to monitor. You can do so through the watch procgroup configuration file directive. You can identify the process group to be monitored through a regular expression that matches the process name and (optionally) its arguments.

watch procgroup Directive--Monitor a Process Group

You can use the watch procgroup directive to monitor a process group as follows:

```
watch procgroup 'regexpr' index flags interval 'description' 'action'
```

'regexpr'

Specifies a quoted string that specifies the regular expression to apply when attempting to match a process name and optional arguments.

index

Specifies the row (index) to use for this entry.

flags

Specifies any additional, non-default behavior to apply to this entry. Specify all flags as hexadecimal numbers (for example, 0x0000).

interval

Specifies the interval that indicates how often (in seconds) the agent monitors the process group.

'description'

Specifies a quoted string, 0 to 128 characters in length, that typically contains a description of the process group being monitored.

'action'

Specifies a quoted string, 0 to 256 characters in length, that specifies the full path of the command (with any parameters) to run when a match is found for the process group. If the string is empty, the agent invokes no action for this entry.

Process Group Monitoring Examples

This section contains sample configuration file directives for process group monitoring.

Example: Monitor the xterm Process Group

The following example configures the CA eHealth SystemEDGE agent to monitor the xterm process group:

```
watch procgroup 'xterm.*' 10 0x00 60 'Watch xterms' ''
```

10

Indicates that this entry will occupy row 10 (pgmonIndex=10) in the Process Group Monitor table.

0x00

Indicates the default behavior.

60

Indicates that the agent should check the xterm process group every 60 seconds.

No action is specified, so the agent invokes no command when it sends a trap.

Example: Monitor the emacs Process Group

The following example configures the CA eHealth SystemEDGE agent to monitor the emacs process group:

```
watch procgroup 'emacs.*|xmibmgr' 11 0x00 60 'Watch emacs' ''
```

11

Indicates that this entry will occupy row 11 (pgmonIndex=11) in the Process Group Monitor table.

0x00

Indicates the default behavior.

60

Indicates that the agent should check the emacs process group every 60 seconds.

No action is specified, so the agent invokes no command when it sends a trap.

Example: Monitor the DT Process Group

The following example configures the CA eHealth SystemEDGE agent to monitor the DT process group:

```
watch procgroup 'dt/bin' 12 0x00 60 'Watch DT stuff' ''
```

12

Indicates that this entry will occupy row 12 (pgmonIndex=12) in the Process Group Monitor Table.

0x00

Indicates the default behavior.

60

Indicates that the agent should check the DT process group every 60 seconds.

No action is specified, so the agent invokes no command when it sends a trap.

Removing Process Group Monitoring Entries

To stop the self-monitoring of a particular process group, you must remove the appropriate entry from the Process Group Monitor table. The watch procgroup directives in the sysedge.cf file create a Process Group Monitor table entry whenever the CA eHealth SystemEDGE agent is started. This row creation results in a new Process Group Monitor table entry that will be stored in sysedge.mon. Permanent removal of a Process Group Monitor table entry requires two steps:

1. Remove the entry from the sysedge.cf file.
2. Remove the entry from the Process Group Monitor table.

Removing Entries from the sysedge.cf File

If you configured a Process Group Monitor table entry by adding a watch procgroup directive to the sysedge.cf file, you must remove it from that file as part of removing the entry from the table. If you do not remove the sysedge.cf directive, the entry will be recreated the next time the CA eHealth SystemEDGE agent is restarted.

Removing Entries with the edgemon Utility

To remove a process group monitoring entry from the Process Group Monitor table, use the edgemon utility to delete the entry. The following examples delete row 14 from the Process Group Monitor table on host 143.45.0.12. After deletion the row will be removed both from memory and from the sysedge.mon file:

```
edgemon -h 143.45.0.12 -c private -v 1 -o procgroup delete 14
```

```
edgemon -h 143.45.0.12 -c private -v 2c -o procgroup setstatus 14 destroy
```

```
edgemon -h 143.45.0.12 -v 3 -u username -s 3 -A MD5 -a authPassword -X DES -x  
encryptPassword -o procgroup delete 14
```

Remove Entries Manually

In some cases it may not be possible to use the edgemon utility to delete Process Group Monitor table entries. For example, if you have configured the CA eHealth SystemEDGE agent to disallow SNMP Set operations, the edgemon utility will not work. In this case, you must remove the entry from the Process Group Monitor table by editing the sysedge.mon file to remove the entry. Because this is an active file, you must stop the CA eHealth SystemEDGE agent before editing the file. For more information about the format of this file, see the appendix "Adding Self-Monitoring Entries to the sysedge.mon File."

To delete row 14 manually

1. Stop the CA eHealth SystemEDGE agent.
2. Open sysedge.mon for editing, delete the entry for procgroupmon row 14, and save the file.
3. Open sysedge.cf for editing, delete the entry for procgroupmon row 14 if it exists, and save the file.
4. Restart the CA eHealth SystemEDGE agent.

Chapter 12: Configuring Log File Monitoring

This chapter explains how to use the CA eHealth SystemEDGE agent to monitor log files for regular expressions.

This section contains the following topics:

[Monitoring Log Files](#) (see page 263)

[Log Monitor Table](#) (see page 264)

[Log Monitor Table Flags](#) (see page 267)

[Log Monitor Table Actions](#) (see page 270)

[View the Log Monitor Table with CA eHealth AdvantEDGE View](#) (see page 272)

[Configuring the Log Monitor Table](#) (see page 272)

[edgewatch Utility--Monitor Log Files](#) (see page 275)

[edgewatch Commands for Log File Monitoring](#) (see page 279)

[Removing Log Monitoring Entries](#) (see page 282)

[Recommendations for Log File Monitoring](#) (see page 283)

Monitoring Log Files

The CA eHealth SystemEDGE agent can monitor ASCII-based text files continuously for the appearance of user-specified regular expressions. Whenever a match for the regular-expression is written to the log file the agent is monitoring, the agent notifies the management system with a trap message.

The flexible Log Monitor table of the Systems Management MIB lets you dynamically configure the agent to monitor a log file for the regular expressions that you specify. You can also specify a simple wildcard expression for the log file you want to monitor, which is evaluated to the single, most recently updated log file matching this expression. Each entry in the table represents the monitoring of a specified log file for a particular regular expression.

This log file monitoring provides a very flexible solution for monitoring applications by monitoring the messages that the applications log. This feature is also useful for security management; for example, you can configure the agent to monitor system log files for su messages to notify you of possible security violations.

When the agent starts (or after rows have been added to the Log Monitor table), it evaluates the log file expression, identifying the most recently updated log file for its current length and last access time. Thereafter, the agent periodically stats each log file that matches the log file expression for additions or modifications since the last status check. This allows a single monitor entry to follow log files that change names (with perhaps date or revision information) without a need to manually modify the given entry.

If the monitored log file has changed, the agent scans *only* the changes--not the entire log file--to see if there is a match for the specified regular expression. If the agent finds a match, you can configure it to send the enterprise-specific logMonMatch SNMP trap to the configured NMS and run the specified action for the row, so long as the action field is not null.

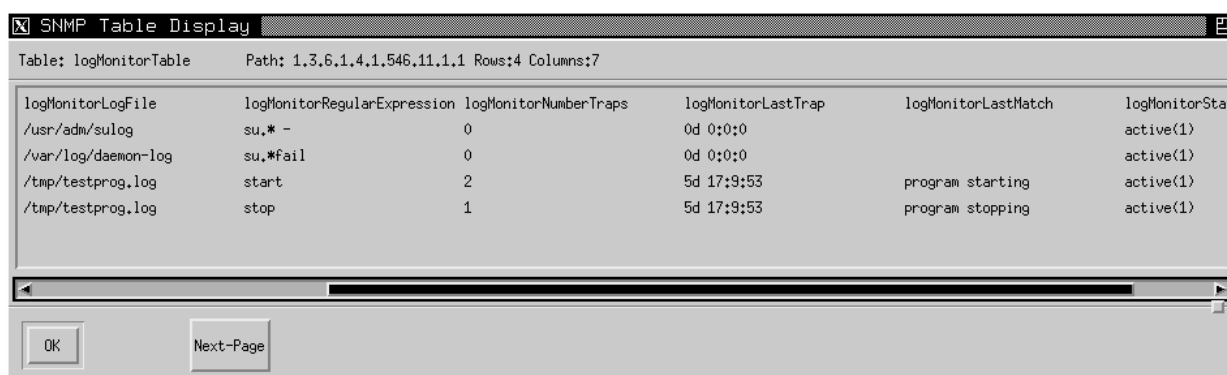
Note: For more information about the logMonMatch trap, see logMonMatch Trap in the chapter "Private Enterprise Traps."

Log Monitor Table

For each entry, the Log Monitor table provides the following types of information:

- Name of the log file that the agent is monitoring for a particular regular expression.
- Number of times a trap has been sent because a match was found.
- Time at which the last trap was sent.
- Log entry that caused the last match.

The following illustration shows the sample entries for the Log Monitor table:



The image shows a window titled "SNMP Table Display" with a table of data. The table has 7 columns and 4 rows. The columns are: logMonitorLogFile, logMonitorRegularExpression, logMonitorNumberTraps, logMonitorLastTrap, logMonitorLastMatch, and logMonitorSta. The rows represent different log files being monitored.

| logMonitorLogFile | logMonitorRegularExpression | logMonitorNumberTraps | logMonitorLastTrap | logMonitorLastMatch | logMonitorSta |
|---------------------|-----------------------------|-----------------------|--------------------|---------------------|---------------|
| /usr/adm/sulog | su,* - | 0 | 0d 0:0:0 | | active(1) |
| /var/log/daemon-log | su,*fail | 0 | 0d 0:0:0 | | active(1) |
| /tmp/testprog.log | start | 2 | 5d 17:9:53 | program starting | active(1) |
| /tmp/testprog.log | stop | 1 | 5d 17:9:53 | program stopping | active(1) |

Columns of the Log Monitor Table

Note: For more information about the Log Monitor Table and its fields, see the specification `empire.asn1` in the `doc` subdirectory of the CA eHealth SystemEDGE agent's installation and the chapter "Systems Management MIB."

Following are the columns of the Log Monitor Table:

LogMonitorIndex

Specifies the row of the table.

Permissions: Read-only

LogMonitorLogFile

Specifies the complete path and file name of the log file to be monitored. This can be a simple wildcard expression, and it will be evaluated by the agent on each scan to identify the most recently updated log file.

Note: The file you monitor must be an ASCII-based text file. CA eHealth SystemEDGE does not support monitoring of other character sets, such as Unicode. You can determine a file's encoding by opening it in a text editor and selecting Save As. The encoding is listed in the Save as type field.

Permissions: Read-write

LogMonitorRegularExpression

Specifies the regular expression to search for when scanning the log files for matches. For information about the rules for specifying regular expressions, refer to the UNIX man page on `egrep(1)`.

Permissions: Read-write

LogMonitorNumberTraps

Specifies the number of times that a trap was sent because a string matching the regular expression was logged to the file.

Permissions: Read-only

LogMonitorLastTrap

Specifies the time, based on `sysUpTime`, at which the agent last sent a trap for this entry.

Permissions: Read-only

LogMonitorLastMatch

Specifies the last log file entry that matched the regular expression; this variable is updated each time a match occurs.

Permissions: Read-write

LogMonitorStatus

Specifies the SNMPv2 RowStatus, which can be one of the following:

- active(1)
- notInService(2)
- notReady(3)

Permissions: Read-write

LogMonitorLogFileSize

Specifies the current size in bytes of the file being monitored.

Permissions: Read-only

LogMonitorLogFileLastUpdate

Specifies the time that the file was last updated.

Permissions: Read-only

LogMonitorDescr

Specifies a quoted string, 0 to 128 characters in length, that typically contains a description of the file being monitored and a severity level for this event.

Permissions: Read-write

LogMonitorAction

Specifies a quoted string, 0 to 256 characters in length, that specifies the full path of the command, with any parameters, to run when the regular expression is matched and a trap is sent. If the string is empty, the agent invokes no action for this entry.

Permissions: Read-write

LogMonitorFlags

Specifies the unsigned integer flags value indicating additional behavior that this row should follow during the course of its operation. By default, this field is assigned the hexadecimal value 0x00.

Note: For more information about this field, see Log Monitor Table Flags.

Permissions: Read-write

LogMonitorMatches

Specifies the number of logfile entries matching the regular expression; this value increments regardless of whether traps are sent for this entry or not. Polling this variable lets you determine the rate or number of matches over time.

Permissions: Read-only

LogMonitorInterval

Specifies the best-effort interval, in minutes, between successive scans of the log file.

Permissions: Read-write

Optimizing Row Creation

You can use the MIB objects with the Log Monitor table to optimize row creation.

The following list describes the scalar objects for optimizing row creation:

logmonUnusedIndex

Returns an unused index number for the Log Monitor table when you perform an SNMP Get on the variable.

logmonMatchDescr

Determines the index number that corresponds to a particular entry description. Perform an SNMP Set of this MIB object to cause the agent to search through entries in the Log Monitor table and put the index value of the last entry whose description matches in the logmonMatchIndex MIB object.

logmonMatchIndex

Matches a particular entry description with its index number when used with logmonMatchDescr.

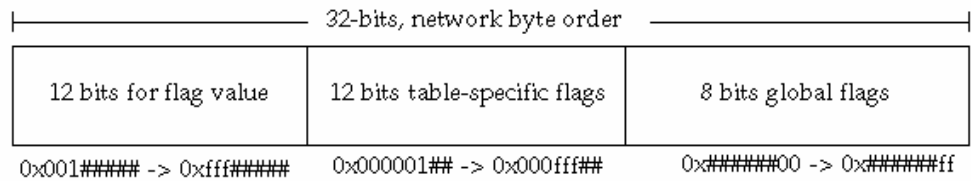
Log Monitor Table Flags

The logMonitorFlags column in the Log Monitor table is a 32-bit unsigned integer that can specify additional behavior for the corresponding Log Monitor table row. By default, the Log Monitor table row does the following:

- Attempts to reinitialize itself
- Sends SNMP traps
- Logs matches to syslog
- Invokes actions (if they are configured)

You can specify different flag bits to alter these defaults. The CA eHealth SystemEDGE agent interprets all flags in hexadecimal (base 16) notation.

The following illustration shows the flags field (logMonitorFlags):



The flags consist of three fields:

- Field1 defines the Common table flags for self-monitoring tables of the System management MIB. This portion is the low-order 8 bits of the flags.
- Field 2 defines the next 12 low-order bits after the common table flags. Table-specific flags are defined separately for each of the self-monitoring tables.

For more information about how the 12 bits are defined for the Log Monitor table, see the illustration in Log Monitor Table.

- Field3 reserves 12 high-order bits for an integer value for use with table-specific flags. This field includes flags specific to the Log Monitor table.

Note: The following sections define each flag bit. You can combine flag values through a logical OR operation.

Following are the flags of the Log Monitor table:

0x00000001

Disables running of actions for this entry.

0x00000002

Disables sending of SNMP traps for this entry. This flag bit overrides any other flag bit with respect to traps.

0x00000004

Disables attempts to reinitialize this entry. By default, if the monitored log file is ever unavailable, the agent will periodically try to reinitialize this table entry. Setting this bit disables automatic reinitialization.

0x00000008

Disables logging of events for this entry through the syslog facility. Setting this bit does not affect trap sending nor action execution. On Windows systems, the agent does not log the event in the agent's log file sysedge.log. Disabling event logging is useful when events occur frequently or when a particular entry is used as an agent heartbeat.

0x00000010

Sends continuous logMonEntryNotReady traps for this entry every time the agent attempts to reinitialize logfile monitoring and fails.

The agent's default behavior sends a single logMonEntryNotReady trap when the log file being monitored ceases to exist, or when an error accessing that log file occurs.

The agent periodically attempts to reinitialize the entry. Enabling this feature causes the agent to send an additional logMonEntryNotReady trap each time reinitialization fails.

0x00000020

Disables the passing of CA eHealth SystemEDGE arguments to action scripts or programs. CA eHealth SystemEDGE typically passes default action parameters that indicate the trap type, description field, and so on. This flag disables the passing of those arguments. For more information about action parameters, see Process Group Monitor Table Actions.

0x00000040

Disables sending of notReady traps for this entry.

0x00000100

Applies the logical NOT operator to the regular-expression evaluation. If the regular expression evaluation equals False, this flag bit will be set to True and cause an event to occur.

If the regular expression evaluation equals True, this flag bit will be set to False. Use caution when utilizing this capability, because False evaluations are converted to True, and vice versa.

0x00000200

Tracks the log file's size, but does not parse through the file. This flag is useful for tracking log file existence and file size in conjunction with Threshold Monitoring.

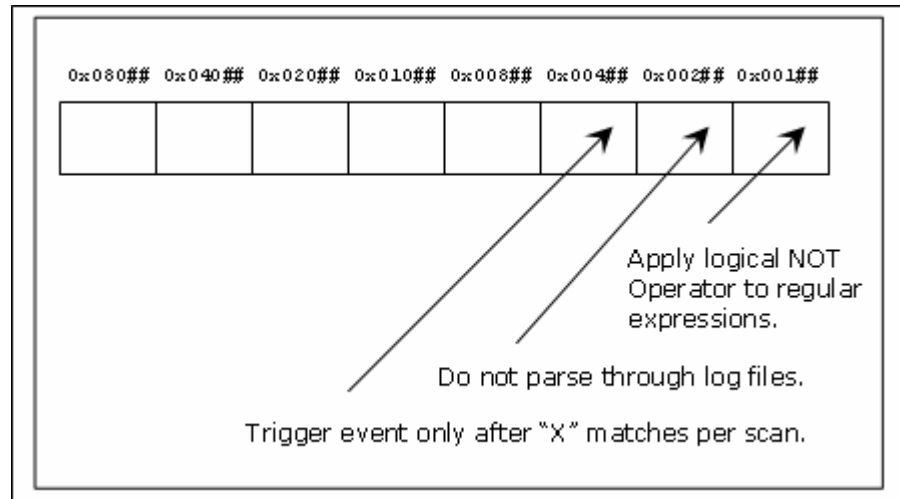
0x00000400

Specifies that the monitor does not trigger an event until after X matches have occurred. This value is calculated on a per-scan basis. If this flag is set, traps, event logging, and actions are suppressed until the number of matches exceeds the X value stored in 0x###00000; at that point, a single event is triggered. This helps to minimize trap traffic with events that occur often.

0x###00000

This X value is used in conjunction with 0x00000400 to specify the number of events required before triggering an event.

The following illustration shows the flag bits specific to the Log Monitor Table:



Log Monitor Table Actions

The CA eHealth SystemEDGE agent provides several default parameters to the action commands when they are invoked. These parameters are in addition to any parameters you specify in the action string and are passed on the command line after those that you specify. The default parameters are the same as the parameters provided in the SNMP traps sent for the Log Monitor table.

Following are the default parameters for Log Monitor table actions:

trapType

Specifies the type of trap being sent, such as logMonMatchEvent or logMonNotReadyEvent.

logMonitorLogFile

Specifies the name of the file that the agent is monitoring. The file you monitor must be an ASCII-based text file. CA eHealth SystemEDGE does not support monitoring of other character sets, such as Unicode.

You can determine a file's encoding by opening it in a text editor and selecting Save As. The encoding is listed in the Save as type field.

logMonitorRegularExpression

Specifies the regular expression that the agent is attempting to match for this entry.

logMonitorLastTrap

Specifies the time that this trap was sent.

logMonitorLastMatch

Specifies the line from the log file that triggered this trap.

logMonitorDescr

Specifies the description of this entry.

logMonitorIndex

Specifies the index of this entry.

logMonitorFlags

Specifies the flags field, in hexadecimal notation (for example, 0x0000), for this entry.

logMonitorInterval

Specifies the poll interval of this entry, in minutes.

LogFileMonitored

Specifies the current log file that is being monitored. This parameter is very useful when the log file entry is specified based on a wildcard expression.

Note: The agent logs action-command invocations at the syslog level LOG_DEBUG. It logs action-command invocation errors at syslog level LOG_WARNING. For information about configuring syslog, see the appendix "Using the syslog Facility." For information about starting the agent with its debugging options turned on, see the chapter "Starting the CA eHealth SystemEDGE Agent."

View the Log Monitor Table with CA eHealth AdvantEDGE View

If you are using CA eHealth AdvantEDGE View, you can query a system for Log Monitor table information by selecting the system you want to monitor from the System list, selecting Logfile Monitoring from the Configuration list, and clicking the Configuration icon.

Note: For more information, see the CA eHealth AdvantEDGE View Web Help.

The following illustration shows the sample CA eHealth AdvantEDGE View Log Monitor table:

| Index | Description | Logfile | Regexpr | Traps | Last Trap | Last Match | Logfile Size | Logfile Last Update | Action | Flags | Interval | Row Status |
|-------|---|---------------------|----------------------------------|-------|-----------|------------|--------------|---------------------|-------------|-------|----------|------------|
| 1 | su attempt - WARNING | /var/log/messages | su.*fail | 0 | 0:00:00 | (no match) | 1,911,901 | 0:14:24 | (no action) | 0x8 | 1 | ● |
| 2 | sysedge fail-NOTICE | /var/log/messages | sysedge.*fail | 0 | 0:00:00 | (no match) | 1,911,901 | 0:14:24 | (no action) | 0x8 | 1 | ● |
| 3 | su attempt - WARNING | /var/log/messages | su.: * - | 0 | 0:00:00 | (no match) | 1,911,901 | 0:14:24 | (no action) | 0x8 | 1 | ● |
| 4 | su attempt - WARNING | /var/log/daemon-log | su.*fail | 0 | 0:00:00 | (no match) | 0 | 0:00:00 | (no action) | 0x8 | 1 | ● |
| 5 | Su attempt | /var/log/messages | su.*session | 0 | 0:00:00 | (no match) | 1,911,901 | 0:14:24 | (no action) | 0x8 | 1 | ● |
| 300 | LPRng exploited | /var/log/messages | Dispatch_input: bad request line | 0 | 0:00:00 | (no match) | 1,911,901 | 0:14:24 | (no action) | 0x8 | 2 | ● |
| 301 | wuoftpd exploited | /var/log/messages | SITE EXEC (lines: 0): | 0 | 0:00:00 | (no match) | 1,911,901 | 0:14:24 | (no action) | 0x8 | 2 | ● |
| 302 | rpc.statd exploited | /var/log/messages | SM_MON request for | 0 | 0:00:00 | (no match) | 1,911,901 | 0:14:24 | (no action) | 0x8 | 2 | ● |
| 303 | adore worm has compromised system! | /var/log/maillog | divsowned | 0 | 0:00:00 | (no match) | 1,223 | 0:00:00 | (no action) | 0x8 | 1 | ● |
| 304 | adore worm has added username dead - system compromised | /etc/shadow | dead: | 0 | 0:00:00 | (no match) | 1,335 | 0:00:00 | (no action) | 0x8 | 1 | ● |
| 305 | ramen worm has compromised system! | /var/log/maillog | chicha | 0 | 0:00:00 | (no match) | 1,223 | 0:00:00 | (no action) | 0x8 | 1 | ● |
| 306 | ramen worm has compromised system! | /var/log/maillog | gb31337 | 0 | 0:00:00 | (no match) | 1,223 | 0:00:00 | (no action) | 0x8 | 1 | ● |
| 307 | Monitor telnets | /var/log/secure | in.telnetd* | 0 | 0:00:00 | (no match) | 493 | 0:00:00 | (no action) | 0x8 | 1 | ● |
| 308 | Monitor login | /var/log/secure | login: | 0 | 0:00:00 | (no match) | 493 | 0:00:00 | (no action) | 0x8 | 1 | ● |

Configuring the Log Monitor Table

You can control which log files the CA eHealth SystemEDGE agent monitors by adding, deleting, or modifying entries in the Log Monitor table. You can configure the Log Monitor table in the following ways:

- Dynamically

Use SNMP commands from a management system, such as CA Health AdvantEDGE View, to modify the table.

Note: For more information, see Dynamic Configuration During Operation.

- At start-up initialization

Specify the entries for the Log Monitor table in the CA eHealth SystemEDGE agent configuration file, sysedge.cf.

Note: For more information, see Initial Configuration During Start-Up.

Initial Configuration During Start-Up

On start-up, the CA eHealth SystemEDGE agent reads the sysedge.cf configuration file and uses the watch logfile directive to specify initial entries to the Log Monitor table. You can add entries to the sysedge.cf file to specify the text files that you want the agent to monitor.

watch logfile Directive--Add Entries to Log Monitor Table

You can use the watch logfile directive to add entries in the Log Monitor table as follows:

```
watch logfile index flags logFilename 'logMonRegExpr' 'logMonDescr'
'logMonAction' logMonInterval
```

index

Specifies the row number of the entry to be created.

flags

Specifies the hexadecimal flags (for example, 0x00001) that direct the additional behavior of this entry.

logFileName

Specifies the complete path (starting from root [/]) and file name of the log file to be monitored. You can specify this as a simple wildcard expression.

The file you monitor must be an ASCII-based text file. CA eHealth SystemEDGE does not support monitoring of other character sets, such as Unicode.

You can determine a file's encoding by opening it in a text editor and selecting Save As. The encoding is listed in the Save as type field.

'logMonRegExpr'

Specifies the regular expressions to apply when scanning the log file for matches. You must enclose this value in single quotation marks ('.').

For information about the rules for specifying regular expressions, see the UNIX man page on egrep(1).

'logMonDescr'

Specifies the description for this entry. It is a quoted string 0 to 128 characters in length and typically contains a descriptions of the file being monitored and the severity level for this event.

'logMonAction'

Specifies a quoted string, 0 to 256 characters in length, that specifies the full path of the command (with any parameters) to run when the regular expression is matched and a trap is sent.

If the string is empty or not specified, the agent invokes no action for this entry.

logMonInterval

Specifies the interval, in minutes, for this entry.

watch logfile Examples

This section provides examples for using the watch logfile directive.

Example: Search for pop Connection Attempts

The following example instructs the CA eHealth SystemEDGE agent to add an entry to the Log Monitor table at table index 15 to search for pop connection attempts on a system:

```
watch logfile 15 0x00 /var/log/syslog 'popper' 'NOTICE - pop connection' '' 1
```

Example: Search for su Attempts

The following example instructs the CA eHealth SystemEDGE agent to add an entry to the Log Monitor table at table index 16 to search for su attempts on a system:

```
watch logfile 16 0x02 /var/adm/messages 'su.*fail' 'WARNING - su attempt'
'/local/bin/mail2admin' 5
```

0x02

Specifies that the agent should not send traps. Instead, the agent invokes the specified action command.

Dynamic Configuration During Operation

You can use your management system to issue SNMP Set request messages to the CA eHealth SystemEDGE agent to modify the entries in the Log Monitor table. Log Monitor table entries are saved to the CA eHealth SystemEDGE agent's sysedge.mon configuration file. This makes sure that any changes made during the operation of the agent are preserved across agent and system restarts.

The agent uses the SNMPv2 SMI Row Status textual convention for creating, deleting, and modifying rows in the table.

Configuration file directives in sysedge.cf take precedence over entries in sysedge.mon. For example, if a Log Monitor table entry is in sysedge.mon at index 10, and a configuration file directive is added to sysedge.cf for index 10 of the Log Monitor table, the entry defined in sysedge.cf replaces the entry taken from sysedge.mon.

edgewidth Utility--Monitor Log Files

To facilitate log file monitoring, the CA eHealth SystemEDGE agent distribution includes the edgewidth command-line utility. edgewidth acts in a manager role to configure entries in the Log Monitor table and list entries that currently exist in the table. The edgewidth utility is located in the bin subdirectory of the CA eHealth SystemEDGE agent's installation.

You can use edgewidth to add, delete, and set the status of, or list entries in the Log Monitor table as follows:

```
edgewidth [-h hostname | ip_addr] [-p port] [-c community]
          [-v 1 | 2c | 3] [-u secName] [-s secLevel] [-n contextName]
          [-a authPassword] [-A MD5 | SHA]
          [-x privPassword] [-X DES | AES | 3DES]
          [-m FIPS mode]
          [-r retries]
          [-t timeout] [-d logLevel] [-f logFile]
          [-o] [facility] [command]
```

-h *hostname* | *ipaddr*

Specifies the hostname or IP address of the system on which the agent is running. Accepts IPv4 and IPv6 addresses.

Default: localhost

-p *port*

Specifies the UDP port that the agent is running on (for example, 1691).

Default: 161

-c *community*

Specifies a community string that the agent uses. Valid for SNMPv1 and SNMPv2c only.

Note: A read-write community string has to be specified for snmpset.

Default: public

-v 1 | 2c | 3

Indicates the version of SNMP that the agent is running. Specify 1 for SNMPv1, 2c for SNMPv2c, or 3 for SNMPv3.

Default: 1

-u *secName*

Specifies the name of the SNMPv3 secure user.

Default: none

-s *secLevel*

Specifies one of the following security levels for SNMPv3 communication:

- 1 - noAuthNoPriv
- 2 - AuthNoPriv
- 3 - AuthPriv (SNMPv3 only)

Default: none

-n *contextName*

Specifies the context name used by the agent if it is configured as SNMPv3.

Note: This option is not required for SNMPv3 communication.

Default: none

-a *authPassword*

Specifies the authentication password if the agent is configured for SNMPv3 with secLevel 2 (AuthNoPriv) or 3 (AuthPriv).

Default: none

-A MD5 | SHA

Specifies the authentication protocol to be used by SNMPv3. This is required if the SNMPv3 user is configured with secLevel 2 (AuthNoPriv) or 3 (AuthPriv). Currently only MD5 (Message Digest Algorithm) and SHA (Secure Hash Algorithm, if the agent is configured for SNMPv3 with secLevel 2 (AuthNoPriv) or 3 (AuthPriv)) are used.

Default: MD5

-x *privPassword*

Specifies the privacy (encryption) password if the agent is configured for SNMPv3 with secLevel 3 (AuthPriv).

Default: none

-X *DES | AES | 3DES*

Specifies the privacy protocol if the SNMPv3 user is configured with secLevel 3 (AuthPriv). Specify DES for Data Encryption Standard, AES for Advanced Encryption Standard using cryptographic keys of 128 bits (AES128), and 3DES for Triple Data Encryption Standard.

Default: none

-m *FIPS_mode*

Controls the FIPS mode of operation. Accepted values are 0, 1, and 2.

0

Indicates Non-FIPS mode.

1

Indicates FIPS co-existence mode.

2

Indicates FIPS only mode.

-r *retries*

Specifies the number of retries.

Default: 3

-t *timeout*

Specifies the duration before the SNMP receiver considers the request as timed out.

Default: 10 seconds

-d logLevel

Specifies the log level of the SNMP messages. Accepted values are 0 to 5.

0

Logs fatal messages.

1

Logs critical messages.

2

Logs warning messages.

3

Logs informational messages.

4

Logs all of the messages.

5

Logs all of the messages including debugging messages.

Default: 0

-f logfile

Specifies the name of the log file that contains error and debug information. The default log file name for most of the utilities is `sysedge_utility.log`

Default: none

facility

Specifies a facility to use. Supported values are the following:

- process
- logfile
- ntevent
- procgroup

Note: The value for [facility] for log file monitoring is logfile.

command

Specifies the command and associated arguments. Supported commands are the following:

- add
- setstatus
- delete
- list

Note: For more information about the commands, see edgework Commands for Log File Monitoring.

edgework Commands for Log File Monitoring

You can use the edgework command for log file monitoring as follows:

```
add logMonIndex logMonFlags "logFilename" "logMonRegExpr" "logMonDescr"  
"logMonAction" logMonInterval  
setstatus logMonIndex status  
delete logMonIndex  
list
```

logMonIndex

Specifies the row in the table. Rows are indexed starting at 1. An index of 0 is not permitted.

Because SNMP does not include a Create PDU type, new table entries are created as a side effect of setting the columnar values for a non-existent row. Therefore, you must include this value for add operations to specify the table index (of an unused row) to use for row creation.

logMonFlags

Specifies the hexadecimal flags (for example, 0x00001) that direct the additional behavior of this entry.

"logFilename"

Specifies the complete path (starting from root [/]) and file name of the log file to be monitored. On Windows systems, the log file name must start with a drive letter and absolute path. You can specify this log file as a simple wild card expression if you want to monitor rotating log files.

Notes:

- For more information, see Rotating Log Files in this chapter.
- The file you monitor must be an ASCII-based text file. CA eHealth SystemEDGE does not support monitoring of other character sets, such as Unicode. You can determine a file's encoding by opening it in a text editor and selecting Save As. The encoding is listed in the Save as type field.

"logMonRegExpr"

Specifies the regular expression to apply when scanning the log file for matches. You must enclose values for logMonRegExpr in quotation marks ("..").

Note: For information about the rules for specifying regular expressions, see *UNIX man* page on *egrep(1)*.

"logMonDescr"

Specifies the description for this entry. The quoted string, 0 to 28 characters in length, typically contains a description of the file being monitored and a severity level for this event.

"logMonAction"

Specifies a quoted string, 0 to 256 characters in length, that specifies the full path of the command (with any parameters) to run when the agents finds a match for the regular expression and sends a trap. If the string is empty or not specified, the agent invokes no action for this entry.

logMonInterval

Specifies the polling interval, in minutes, for this monitor entry.

Status

Specifies the RowStatus textual convention value to use in setting the status of a row in the Log Monitor table. Used with the setstatus operation.

The Row Status parameter is an integer that can take on one of the following values. The values can be either assigned integer values or the actual spelled out status text:

- active(1)
- notInService(2)
- notReady(3)
- destroy(6)

Sample Uses of the edgwatch Utility

This section provides examples for using the edgwatch utility.

Example: List Entries in Log Monitor Table

The contents of the CA eHealth SystemEDGE agent's Log Monitor table will be displayed as the following:

```
edgwatch -v 1 -h 127.0.0.1 -c public -o logfile list
```

```
edgwatch -v 2c -h fe80::2367:1 -c public -o logfile list
```

```
edgwatch -v 3 -s 3 -u userName -x privPassword -X encryptProtocol -A  
authProtocol -a authPassword -o logfile list
```


Add a Log Monitor Entry

The following example instructs the CA eHealth SystemEDGE agent to add an entry to the Log Monitor table at table index 5 to search for su failures on an HP-UX system. The agent runs the script /local/bin/mail2admin when it finds a match.

```
edgework -v 1 -h 127.0.0.1 -c private -o logfile add 5 0x00 /usr/adm/sulog "SU.*  
-" "su attempt - WARNING" "/local/bin/mail2admin" 1
```

```
edgework -v 2c -h fe80::2367:1 -c private -o logfile add 5 0x00 /usr/adm/sulog  
"SU.* -" "su attempt - WARNING" "/local/bin/mail2admin" 1
```

```
edgework -v 3 -h fe80::2367:1 -s 3 -u userName -A authProtocol -a authPassword -  
X encryptProtocol -x privPassword -o logfile add 5 0x00 /usr/adm/sulog "SU.* -"  
"su attempt - WARNING" "/local/bin/mail2admin" 1
```

Example: Delete a Log Monitor Entry

The following example deletes an entry from an agent's Log Monitor table at table index 5:

```
edgework -v 1 -h 127.0.0.1 -c private -o logfile delete 5
```

```
edgework -v 2c -h fe80::2367:1 -c private -o logfile delete 5
```

```
edgework -v 3 -h 127.0.0.1 -s 3 -u userName -A authProtocol -a authPassword -X  
encryptProtocol -x privPassword -o logfile delete 5
```

Example: Disable a Log Monitor Entry

The following example disables the Log Monitor table entry at table index 5 by setting that entry's status to notInService(2). The entry will remain in the table, but the agent will not scan the log file for the regular expression unless the entry's status returns to active (1).

```
edgework -v 1 -h 127.0.0.1 -c private -o logfile setstatus 5 notInService
```

```
edgework -v 2c -h fe80::2367:1 -c private -o logfile setstatus 5 notInService
```

```
edgework -v 3 -h 127.0.0.1 -s 3 -u userName -A authProtocol -a authPassword -X  
encryptProtocol -x privPassword -o logfile setstatus 5 notInService
```

Removing Log Monitoring Entries

To stop the self-monitoring of a log file, you must remove the appropriate entry from the Log Monitor table. Log Monitor table entries are stored in the `sysedge.mon` file to make sure that they will not be lost when the CA eHealth SystemEDGE agent restarts.

The watch logfile directives in the `sysedge.cf` file create a Log Monitor entry whenever the CA eHealth SystemEDGE agent starts. This row creation results in a new Log Monitor table entry that will be stored in `sysedge.mon`. Thus, permanent removal of a Log Monitor table entry requires two steps:

1. Remove the entry from the `sysedge.cf` file.
2. Remove the entry from the Log Monitor table.

Removing Entries from the `sysedge.cf` File

If you configured a Log Monitor table entry by adding a watch logfile directive to the `sysedge.cf` file, you must manually delete it from that file as part of removing the entry from the table. If you do not remove the `sysedge.cf` directive, the entry will be recreated the next time the CA eHealth SystemEDGE agent starts.

Removing Entries with the `edgewatch` Utility

To remove a log file monitoring entry from the Log Monitor table, use the `edgewatch` utility to delete the entry. The following examples delete row 14 from the Log Monitor table on system *ipaddress*. After deletion, the row is removed both from memory and from the `sysedge.mon` file.

```
edgewatch -v 1 -h fe80::2367:1 -c private -o logfile delete 14
```

```
edgewatch -v 2c -h 127.0.0.1 -c private -o logfile delete 14
```

```
edgewatch -v 2c -h 127.0.0.1 -c private -o logfile setstatus 14 6
```

```
edgewatch -v 3 -h fe80::2367:1 -s 3 -u userName -A authProtocol -a authPassword -  
X encryptProtocol -x privPassword -o logfile setstatus 14 destroy
```

Remove Entries Manually

In some cases, it may not be possible to use the edgemon utility to delete Log Monitor table entries. For example, if you have configured the CA eHealth SystemEDGE agent to disallow SNMP Set operations, the edgemon utility will not work. In this case, you must remove the entry from the Log Monitor table by editing the sysedge.mon file and removing the entry from the file. Because this is an active file, you must stop the CA eHealth SystemEDGE agent before you edit it.

Note: For more information about the format of this file, see the appendix “Adding Self-Monitoring Entries to the sysedge.mon File.”

To delete row 14 manually

1. Stop the CA eHealth SystemEDGE agent.
2. Open sysedge.mon for editing, delete the entry for logmon row 14, and save the file.
3. Open sysedge.cf for editing; delete the entry for logmon row 14 if it exists, and save the file.
4. Restart the CA eHealth SystemEDGE agent.

Recommendations for Log File Monitoring

You can monitor system and application logs to obtain in-depth information about user, system, and application behavior.

The following tables describes recommendations for which log files you can monitor and the regular expressions for which you can search for monitoring security, device failures, system capacity, Windows security, and applications and systems:

| Description | Log File to Monitor | Regular Expression |
|-----------------------|---------------------|--------------------|
| WARNING - daemon core | /var/log/daemon-log | core dumped |
| WARNING - daemon core | /var/log/syslog | core dumped |
| Monitor SU attempts | /var/adm/messages | su.*fail |
| Monitor rlogins | /var/log/syslog | in.rlogin |
| Monitor telnets | /var/log/syslog | in.telnet |
| Monitor rsh | /var/log/syslog | in.rsh |

| Description | Log File to Monitor | Regular Expression |
|--|---------------------|-----------------------|
| WARNING - Illegal Instruction, Daemon | /var/log/daemon-log | .*Illegal.*nstruction |
| Spam Relay Attempt | /var/log/syslog | Relaying denied |
| Monitor DENY packets from a Linux firewall | /var/log/messages | DENY |

Note: The log files described in this section are not the same for all operating systems. These examples are provided for reference. You should alter them for your operating system.

The following table describes the recommendations for regular expressions for which you can search in the syslog (/var/adm/messages) to monitor for device failures:

| Description | Regular Expression |
|-------------------------------------|--------------------------------------|
| Critical: Badtrap Error | .*BAD TRAP.* |
| Error: SCSI error | .*SCSI.*[E,e]rror.* |
| Error: SCSI error | .*SCSI.*failed.* |
| Error: SCSI error | .*SCSI.*hung.* |
| Critical: badsimms error | .*SIMM.* |
| Critical: badsimms error | .*BAD.*SIMM.* |
| Critical: memory error | .*[M,m]emory [E,e]rror.* |
| Error: disk error | .*disk not responding.* |
| Error: disk error | .*[D,d]isk.*[E,e]rror.* |
| 'Warning: disk fragmentation error' | .*optimization changed.* |
| Error: disk error | .*corrupt label.* |
| Error: I/O error | .*I/O.*[E,e]rror.* |
| Error: disk read/write errors | .*Error for Command:.*[read,write].* |
| Error: media error | .*Media Error.* |
| Info: serialport error | .*zs[0,1,2]: silo overflow.* |
| Warning: carrier error | .*no carrier.* |
| Warning: link is down | .*Link Down - cable problem?.* |
| Error: SDS error | .*[NOTICE,WARNING,PANIC]: md:.* |

The following table describes recommendations for regular expressions for which you can search in /var/adm/messages to monitor system capacity:

| Description | Regular Expression |
|------------------------|-----------------------------|
| Critical: memory error | *[O,o]ut of [M,m]emory.* |
| Critical: memory error | .*[F,f]ile system full.* |
| Error: diskspace error | .*No space left on device.* |

The following table describes recommendations for which logs and expressions you can monitor in the Windows event logs to monitor Windows security:

| Description | Event Log | Security Type | Regular Expression |
|---|-------------|---------------|--|
| Random Password Hack | Security | All | .*bad |
| Misuse of Privileges | Security | All | .*[user rights,group management,security change, restart,shutdown] |
| Improper File Access | Security | Failure | .*[read,write] |
| Improper Printer Access | Security | Failure | .*print |
| Virus Outbreak Warning: program files updated | Security | All | .*write.*[exe,dll,com] |
| Security Policies Change | Application | Information | .*[S,s]ecurity policy |

The following table describes recommendations for which logs and expressions you can monitor in the Windows event logs to monitor applications and systems:

| Description | Event Log | Security Type | Regular Expression |
|------------------------------|-------------|---------------|-------------------------|
| Application Error or Failure | Application | All | .*[F,f]ail.*[E,e]rror |
| Application Load Problems | Application | All | .*[L,l]oad.*[P,p]roblem |

| Description | Event Log | Security Type | Regular Expression |
|---|-------------|---------------|-----------------------------|
| New Software Installed | Application | All | .*[INSTALL,Install,install] |
| Server Process failed during Initialization | Application | All | .*4131 |
| Disk Failures and errors | All | All | .*[D,d]isk |
| Network Adapter Errors | All | Error | .*[N,n]etwork [A,a]dapter |

Monitor Log File Size

To monitor the size of a log file, add a log file monitoring entry to sysedge.cf with the “Do not parse file” flag turned on, and then set up threshold monitoring for the logFileSize x variable, where x is the log monitor entry.

To monitor the log file size

1. Create a log file monitoring entry using the edgewatch utility as follows. You do not need to restart the agent after issuing this command.

```
edgewatch -v 1 -h 127.0.0.1 -c public -o logfile add 10 0x20a "Monitor Log File Size" "/var/log/messages" "dontfindanything" ""
```

```
edgewatch -v 2c -h fe80::2367:1 -c public -o logfile add 10 0x20a "Monitor Log File Size" "/var/log/messages" "dontfindanything" ""
```

```
edgewatch -v 3 -s 3 -u userName -x privPassword -X encryptProtocol -A authProtocol -a authPassword -o logfile add 10 0x20a "Monitor Log File Size" "/var/log/messages" "dontfindanything" ""
```

Note: You can also create a log file monitoring entry manually in `sysedge.mon`. This is not a preferred method of creating a monitor, and it requires you to stop the agent before you add the monitor and restart it afterwards. Following is an example log monitoring entry in the `sysedge.mon` file:

```
logmon {
10
"Monitor Log File Size"
"/var/log/messages"
"dontfindanything"
""
0x20a
active
1
}
```

Note: These example are for a Linux system. For Solaris, specify `/var/adm/messages`, and for HP-UX, specify `/var/adm/syslog/syslog.log`.

2. Create a threshold-monitoring entry using the `edgemon` utility as follows. You do not need to restart the agent after issuing this command.

```
edgemon -h 143.45.0.12 -c private -v 1 -o oid 1.3.6.1.4.1.546.11.1.8.10 20
0x8 60 1 2 1048576 "Send Trap When file is larger than 1048576" ""
```

```
edgemon -h fe80::2367:1 -c private -v 2c -o oid 1.3.6.1.4.1.546.11.1.8.10 20
0x8 60 1 2 1048576 "Send Trap When file is larger than 1048576" ""
```

```
edgemon -h fe80::2367:1 -v 3 -u userName -s 3 -a authPassword -A MD5 -x
encryptPassword -X DES -o oid 1.3.6.1.4.1.546.11.1.8.10 20 0x8 60 1 2 1048576
"Send Trap When file is larger than 1048576" ""
```

Note: You can also create a threshold monitoring entry in the `sysedge.mon` file that monitors the tenth entry in the Log Monitor table (`logMonitorLogFileSize.10`, or `1.3.6.1.4.1.546.11.1.8.10`) and sends a trap when the log file is larger than 1048576. This is not a preferred method of creating a monitor, and it requires you to stop the agent before you add the monitor and start it afterwards. Following is an example log monitoring entry in the `sysedge.mon` entry:

```
monentry {
20
"Send Trap When file is larger than 1048576"
60
absoluteValue
1.3.6.1.4.1.546.11.1.8.10
gt
1048576
Active
""
0x8
0
}
```

```
}
```

Note: This example is for a Linux system. For Solaris, specify `/var/adm/messages`, and for HP-UX, specify `/var/adm/syslog/syslog.log`.

Rotating Log Files

With the latest CA eHealth SystemEDGE Agent, rotating log files are no longer a significant concern. Previously, you were required to use symbolic links. You now simply use an appropriate wild card entry to specify the log file entries. The wildcard syntax is platform dependent and allows you to use the asterisk (*) for any number of characters and the question mark (?) to indicate any single character. The agent monitors the most recent matching log file and automatically switches to the next rolling log file as it is created and updated.

Chapter 13: Configuring Windows Event Monitoring

This chapter explains how to use the CA eHealth SystemEDGE agent to monitor Windows event logs for regular expressions.

Important! This chapter is relevant only for Windows versions of the CA eHealth SystemEDGE agent.

This section contains the following topics:

[Monitoring Windows Events](#) (see page 289)

[Monitoring Windows Event Logs](#) (see page 290)

[NT Event Monitor Table](#) (see page 291)

[NT Event Monitor Table Flags](#) (see page 295)

[NT Event Monitor Table Actions](#) (see page 297)

[View the NT Event Monitor Table with CA eHealth AdvantEDGE View](#) (see page 298)

[Configuring the NT Event Monitor Table](#) (see page 299)

[Removing NT Event Monitoring Entries](#) (see page 309)

Monitoring Windows Events

The CA eHealth SystemEDGE agent enables you to instruct the agent to continuously monitor Windows event logs. This Windows event-log monitoring is similar to the standard log file monitoring described in the chapter “Configuring Log File Monitoring.”

Whenever a matching event is generated on a system that the agent is monitoring, the CA eHealth SystemEDGE agent notifies the management system with a trap message. It can also run action commands to handle the event immediately. Because Windows events include several identifying characteristics in addition to the text message, this monitoring capability is somewhat more sophisticated than the standard log file monitoring in the types of matches that you can specify.

Monitoring Windows Event Logs

The flexible NT Event Monitor Table of the Systems Management MIB is located under the nt system branch in the MIB. It enables you to configure the CA eHealth SystemEDGE agent dynamically to monitor event logs for the regular expression that you specify. Each entry in the NT Event Monitor table represents the monitoring of one event log for a particular regular expression.

The NT Event Monitor table monitors the messages that applications log to the Windows event mechanism. Windows event monitoring is also useful in security management; for example, you can configure the agent to monitor the Security Event Log for invalid logins.

Checking Log File Status

When the CA eHealth SystemEDGE agent starts (or after the addition of rows to the NT Event Monitor table), it checks the status (stats) of each Windows event log for its current length and the time that it was last updated. Thereafter, the CA eHealth SystemEDGE agent periodically scans each event log for additions or modifications since the last update. If the event log file has changed, the agent scans only the changes--not the entire event log--to see if a match exists for the specified filters.

If the agent finds a match, it sends an enterprise-specific SNMP ntEventMonMatch trap message to the configured management systems. For more information about the ntEventMonMatch trap, see ntEventMonMatch Trap in the chapter "Private Enterprise Traps."

Search Criteria

Each Windows Event Monitor table entry instructs the agent to search for matches based on the criteria described in the following table:

Event Log

Specifies the name of the event log. This value can be any of the following:

- Application
- System
- Security
- DirService (for Directory Service)
- DnsServer (for DNS Service)
- FileRepService (for File Replication Service)

Event Type

Specifies the type of event. Types 1 through 5 are defined by Windows as the following:

- error(1)
- warning(2)
- information(3)
- success(4)
- failure(5)

Type all(6) indicates that the agent should match all event types.

Event Source

Specifies the name of the program or module that generated the event. The agent uses regular expressions to match this field.

Event Description

Describes the event. The agent uses regular expressions to match this field.

The CA eHealth SystemEDGE agent generates an SNMP trap message when it finds a match based on all four criteria. This matching is similar to a Boolean AND operation.

NT Event Monitor Table

For each entry in the NT Event Monitor table, the table provides information such as the following:

- Event log that the agent is monitoring
- Regular expression for which the log is being monitored
- Number of times that a trap has been sent because a match was found
- Time at which the last trap was sent
- Log entry that caused the last match

Columns of the NT Event Monitor Table

The following table describes the columns of the NT Event Monitor table. For more information about the NT Event Monitor table, see the Systems Management MIB `empire.asn1` in the `doc` subdirectory of the CA eHealth SystemEDGE agent's installation.

ntEventMonIndex

Specifies the row of the table in which this entry exists.

Permissions: Read-only

ntEventMonLog

Specifies the integer that designates which event log to monitor. The following are possible values:

- Application(1)
- Security(2)
- System(3)
- Directory Service(4)
- DNS Service(5)
- File Replication Service(6)

Permissions: Read-write

ntEventMonTime

Specifies the time, based on `sysUpTime`, at which the event occurred.

Permissions: Read-only

ntEventMonMatches

Specifies the number of times that a match occurred for this entry and the agent sent a trap.

Permissions: Read-only

ntEventMonTypeLastMatch

Specifies the number that identifies the event type of the last event that matched the search criteria. Types 1 through 5 are defined by Windows as the following:

- error(1)
- warning(2)
- information(3)
- success(4)
- failure(5)

Type noMatch(6) indicates that there has not yet been a matching event for this monitoring entry.

Permissions: Read-only

ntEventMonTypeFilter

Specifies the number that identifies the event type to match for this entry. Types 1 through 5 are defined by Windows as the following:

- error(1)
- warning(2)
- information(3)
- success(4)
- failure(5)

Type all(6) indicates that the agent should match all event types.

Permissions: Read-write

ntEventMonSrcLastMatch

Identifies the Event Source of the last event log entry that matched this monitor entry. The Event Source is usually the name of the program that generated the event. Each time a match occurs, this variable is updated.

Permissions: Read-only

ntEventMonSrcFilter

Specifies the regular expression to apply to the Event Source when scanning the events for matches.

Permissions: Read-write

ntEventMonDescLastMatch

Specifies the last event log entry that matched this monitor entry. Each time a match occurs, this variable is updated.

Permissions: Read-only

ntEventMonDescFilter

Specifies the regular expression to apply to the Event Description when scanning the events for matches.

Permissions: Read-write

ntEventMonStatus

Specifies the SNMPv2 RowStatus, which can be one of the following:

- active(1)
- notInService(2)
- notReady(3)

Permissions: Read-write

ntEventMonDescr

Specifies a quoted string, 0 to 128 characters in length, that typically contains a description of the events being monitored and a severity level for this event.

Permissions: Read-write

ntEventMonAction

Specifies a quoted string, 0 to 256 characters in length, that specifies the full path of the command (with any parameters) which runs when a match is found and a trap is sent. If the string is empty, the agent invokes no action for this entry.

Permissions: Read-write

ntEventMonFlags

Specifies the unsigned integer flags value indicating additional behavior that this row should follow during the course of its operation. By default, this field is assigned the hexadecimal value 0x00. For more information about this field, see NT Event Monitor Table Flags.

Optimizing Row Creation

You can use the following MIB objects with the NT Event Monitor table to optimize row creation:

ntEventUnusedIndex

Returns an unused index number for the NT Event Monitor table when you perform an SNMP Get on the variable.

ntEventMatchDescr

Determines the index number that corresponds to a particular entry description. Perform an SNMP Set of this MIB object to cause the agent to search through entries in the NT Event Monitor table and put the index value of the last entry whose description matches in the ntEventMatchIndex MIB object.

ntEventMatchIndex

Matches a particular entry description with its index number when used with ntEventMatchDescr.

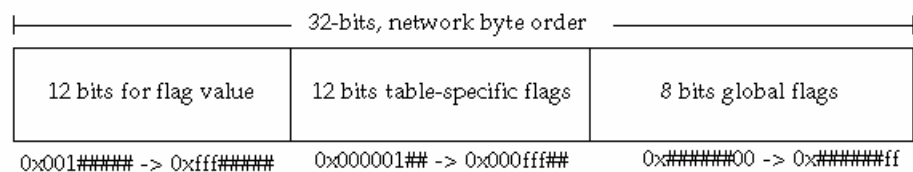
NT Event Monitor Table Flags

You can use the ntEventMonFlags column in the NT Event Monitor Table to specify additional behavioral semantics for the corresponding NT Event Monitor table row. By default, the NT Event Monitor table row does the following:

- Attempts to reinitialize itself
- Sends SNMP traps
- Logs events with the syslog utility
- Invokes actions (if they are configured)

You can set different flag bits to alter these defaults. The CA eHealth SystemEDGE agent interprets all flags in hexadecimal (base 16) notation.

The following illustration shows the composition of the NT Event Monitor table flags field (ntEventmonFlags).



The flags value consists of three fields:

- **Field 1:** Common table flags defined for all of the self-monitoring tables. This portion is the low-order 8 bits of the flags.
- **Field 2:** Table-specific flags defined separately for each self-monitoring table. This field defines the next 12 low-order bits after the common table flags. For Windows event monitoring, there are currently no table-specific flags defined.
- **Field 3:** Flags value reserves the 12 high-order bits for an integer value for use with table-specific flags. For Windows event monitoring, there are currently no table-specific flags defined.

The following sections define each flag bit. You can combine flag values through logical OR operations.

The following list describes the common table flags for the self-monitoring tables:

0x00000001

Disables running of actions for this entry.

0x00000002

Disables sending of SNMP traps for this entry. This flag bit overrides any other flag bit with respect to traps.

0x00000004

Disables attempts to reinitialize this entry. By default, if the monitored event log is unavailable, the CA eHealth SystemEDGE agent periodically tries to reinitialize this table entry. Setting this bit disables automatic reinitialization.

0x00000008

Disables logging of events for this entry through the syslog facility. Setting this bit does not affect the sending of traps or execution of actions. On Windows systems, the agent will not log the event to the agent's log file (sysedge.log). When events occur frequently, it is useful to disable event logging.

0x00000010

Sends continuous ntEventMonEntryNotReady traps for this entry every time the agent attempts to reinitialize event log monitoring and fails. The agent's default behavior is to send a single ntEventMonEntryNotReady trap when the event log that it is monitoring ceases to exist or when an error accessing that event log occurs. The agent periodically attempts to reinitialize the entry. Enabling this feature causes the agent to send an additional ntEventMonEntryNotReady trap each time reinitialization fails.

0x00000020

Disables the passing of CA eHealth SystemEDGE arguments to action scripts or programs. CA eHealth SystemEDGE typically passes default action parameters that indicate the trap type, description field, and so on. This flag disables the passing of those arguments. For more information about action parameters, see NT Event Monitor Table Actions.

0x00000040

Disables sending of notReady traps for this entry.

0x00000100

Pre-appends the Event ID number to the vent description field. The identifier is added using the form [#] where # is the identifier. The purpose of this flag is to facilitate searches for specific Event IDs. For example, to specify that only Event ID 528 should match, specify this flag with the ntEventMonDescFilter set to [528].

0x00000200

Sends every event except the specified event. When you enable this flag, the agent applies matches to expressions and does the following: if the match is true, it sets it to false, and if the match is false, it sets it to true. If the results are true, the agent performs the specified action or sends a trap.

Note: Use caution in setting this flag.

0x####0000

The NT Event Monitor Table does not use this flag value field; setting it has no effect on the NT Event Monitor Table operation or on any of the supported flag bits.

NT Event Monitor Table Actions

The CA eHealth SystemEDGE agent provides several default parameters to the action commands. These parameters are in addition to any parameters you specify in the action string and are passed on the command line after those that you specify. The default parameters are the same as the parameters provided in the SNMP traps sent for the NT Event Monitor table.

The following list describes the default parameters for NT Event Monitor table actions:

trapType

Specifies the type of trap sent, such as a `ntEventMonMatchEvent` or `ntEventMonNotReadyEvent`.

ntEventMonLog

Specifies the event log that the agent is monitoring.

ntEventMonTypeLastMatch

Specifies the type of event that generated this trap.

ntEventMonTime

Specifies the time that this event was generated.

ntEventMonSrcLastMatch

Specifies the source of the event that generated this trap.

ntEventMonDescLastMatch

Specifies a description of the event that generated this trap.

ntEventMonDesc

Specifies a description of this entry.

ntEventMonIndex

Specifies the index for this entry.

ntEventMonFlags




Specifies the flags field, in hexadecimal notation (for example, 0x0000), for this table entry.

For more information about traps sent by the CA eHealth SystemEDGE agent, see the chapter "Private Enterprise Traps."

The CA eHealth SystemEDGE agent logs action-command invocation errors to the sysedge.log file. For information about starting the agent with its debugging options turned on, see Configuring Support for Agent Debugging in the chapter "Configuring the CA eHealth SystemEDGE Agent."

View the NT Event Monitor Table with CA eHealth AdvantEDGE View

If you are using CA eHealth AdvantEDGE View, you can query a system for NT Event Monitor table information by selecting the system you want to monitor from the System list, selecting NT Event Monitoring from the Configuration list, and clicking the Configuration icon. For more information, see the CA eHealth AdvantEDGE View Web Help. The following illustration shows a sample CA eHealth AdvantEDGE View Monitor table:

| Index | Description | Monitor Time | Num Matches | Log | Type Filter | Src Filter | Descr Filter | Last Type Match | Last Src Match | Last Descr Match | Action | Flags | Row Status |
|--------------|--------------|------------------|-------------|-----------------|-------------|------------|--------------|-----------------|----------------|--|-------------|-------|---|
| 30 | test | 17 days, 5:18:36 | 124 | application (1) | error (1) | * | * | error(1) | Userenv | Windows cannot determine the user or computer name. Return value (1722). | (no action) | 0x20 |  |
| 500 | Monitor blah | 0:00:00 | 0 | application (1) | error (1) | blah | blah | noMatch (6) | (no match) | (no match) | (no action) | 0x20 |  |
| 12345 | eHealth test | 0:00:00 | 0 | application (1) | error (1) | * | * | noMatch (6) | (no match) | (no match) | (no action) | 0x20 |  |
| | | | | | | | | | | | | | |

Add NT Event Monitor Entry

Configuring the NT Event Monitor Table

You can control which event logs the CA eHealth SystemEDGE agent monitors by adding, deleting, or modifying entries in the NT Event Monitor table.

You can configure the NT Event Monitor table in one of these ways:

- Dynamically. Use SNMP commands from a management system, such as CA eHealth AdvantEDGE View, to modify the table. For more information, see *Dynamic Configuration During Operation*.
- At start-up initialization. Specify the entries for the NT Event Monitor table in the CA eHealth SystemEDGE agent configuration file, `sysedge.cf`. For more information, see *Initial Configuration During Start-Up*.

Dynamic Configuration During Operation

You can use your NMS platform to issue SNMP Set request messages to the CA eHealth SystemEDGE agent to modify the entries in the NT Event Monitor Table. NT Event Monitor table entries are saved to the `sysedge.mon` configuration file so that any changes made during the operation of the agent are preserved across agent and system restarts.

The agent uses the SNMPv2 SMI Row Status textual convention for creating, deleting, and modifying rows in the table.

Configuration file directives in the `sysedge.cf` file take precedence over entries in `sysedge.mon`. For example, if a NT Event Monitor table entry is in `sysedge.mon` at index 10, and a configuration file directive is added to `sysedge.cf` for index 10 of the NT Event Monitor table, the entry defined in `sysedge.cf` overwrites the entry from `sysedge.mon`.

Initial Configuration During Start-Up

On start-up, the CA eHealth SystemEDGE agent reads the `sysedge.cf` configuration file and uses the `watch ntevent` keyword to specify initial entries to the NT Event Monitor Table. You can specify event logs that you want the agent to monitor by adding appropriate entries for the files, types, and expressions you want to monitor to the `sysedge.cf` file.

watch ntevent Directive--Add Entries to ntEventMonTable

You can use the watch ntevent directive to add entries to the ntEventMonTable as follows:

```
watch ntevent ntEventMonIndex ntEventMonFlags ntEventMonLog ntEventMonTypeFilter  
'ntEventMonSrcFilter' 'ntEventMonDescFilter' 'ntEventMonDescr' 'ntEventMonAction'
```

ntEventMonIndex

Specifies the row number of the entry to be created in the NT Event Monitor Table.

ntEventMonFlags

Specifies the hexadecimal flags (for example, 0x00001) that direct the additional behavioral semantics of this entry.

ntEventMonLog

Specifies the event log to monitor. This value can be one of the following:

- Application
- System
- Security

ntEventMonTypeFilter

Specifies the event type to match for this entry. The following are valid types:

- Error
- Warning
- Information
- Success
- Failure
- All

Type All indicates that all event types should match.

'ntEventMonSrcFilt'

Specifies the regular expression to apply when scanning the Event Source attribute for each event in the log. You must enclose this value in single quotation marks ('.'). For more information about specifying regular expressions, see the UNIX man page on `egrep(1)`.

'ntEventMonDescFilt'

Specifies the regular expression to apply when scanning the Event Description attribute for each event in the log. You must enclose this value in single quotation marks ('.'). For more information about specifying regular expressions, see the UNIX man page on `egrep(1)`.

'ntEventMonDescr'

Specifies a description for this entry. This value is a quoted string, 0 to 128 characters in length, and contains a description of the file being monitored and a severity level for this event.

'ntEventMonAction'

Specifies a quoted string, 0 to 256 characters in length, that specifies the full path of the command (with any parameters) which runs when the entry is matched and the agent sends a trap. If the string is empty or not specified, the agent invokes no action for this entry.

watch ntevent Directive Examples

This section includes examples for using the watch ntevent directive.

Example: Search the Application Log for Web Server Messages

The following example adds a new entry to the agent's NT Event Monitor table at table index 1 to search the Application log for messages from the http Web server application:

```
watch ntevent 1 0x00 Application All 'http' '.*' 'Web Server messages' "
```

Example: Search the Security Log for Failure Events

The following example adds a new entry to the agent's NT Event Monitor table at table index 2 to search the Security log for Failure events that indicate login failures:

```
watch ntevent 2 0x00 Security Failure '.*' '.*' 'Access Failure - WARNING' "
```

Example: Search the Application Log for Specific Events

The following example adds a new entry to the agent's NT Event Monitor Table at table index 3 to search the Application log for events with Event ID 277:

```
watch ntevent 3 0x0100 Application All '.*' '[277\]' 'Event ID 277' "
```

0x0100

Adds the Event ID to the description. [277\] is the description field that the agent will attempt to match.

The backslash character (\) is required because brackets ([]) are special characters for regular expression matching.

edgework Utility--Monitor Windows Events

To facilitate Windows event monitoring, CA eHealth SystemEDGE includes the edgework command-line utility. This utility acts in a manager role to configure entries in the NT Event Monitor table and list entries currently in the table. The edgework utility is located in the bin subdirectory of the CA eHealth SystemEDGE agent's installation.

Although the NT Event monitoring capability is provided on Windows only, you can configure it from any supported platform using the edgework utility. That is, you can configure event monitoring on a Windows system by using edgework from a Solaris system.

You can use edgework to add, delete, set the status of, or list entries in the NT Event Monitor table.

You can use the edgework utility as follows:

```
edgework [-h hostname | ip_addr] [-p port] [-c community]
        [-v 1 | 2c | 3] [-u secName] [-s secLevel] [-n contextName]
        [-a authPassword] [-A MD5 | SHA]
        [-x privPassword] [-X DES | AES | 3DES]
        [-m FIPS_mode]
        [-r retries]
        [-t timeout] [-d logLevel] [-f logFile]
        [-o] [facility] [command]
```

-h *hostname* | *ipaddr*

Specifies the hostname or IP address of the system on which the agent is running. Accepts IPv4 and IPv6 addresses.

Default: localhost

-p *port*

Specifies the UDP port that the agent is running on (for example, 1691).

Default: 161

-c *community*

Specifies a community string that the agent uses. Valid for SNMPv1 and SNMPv2c only.

Note: A read-write community string has to be specified for snmpset.

Default: public

-v 1 | 2c | 3

Indicates the version of SNMP that the agent is running. Specify 1 for SNMPv1, 2c for SNMPv2c, or 3 for SNMPv3.

Default: 1

-u *secName*

Specifies the name of the SNMPv3 secure user.

Default: none

-s *secLevel*

Specifies one of the following security levels for SNMPv3 communication:

- 1 - noAuthNoPriv
- 2 - AuthNoPriv
- 3 - AuthPriv (SNMPv3 only)

Default: none

-n *contextName*

Specifies the context name used by the agent if it is configured as SNMPv3.

Note: This option is not required for SNMPv3 communication.

Default: none

-a *authPassword*

Specifies the authentication password if the agent is configured for SNMPv3 with secLevel 2 (AuthNoPriv) or 3 (AuthPriv).

Default: none

-A MD5 | SHA

Specifies the authentication protocol to be used by SNMPv3. This is required if the SNMPv3 user is configured with secLevel 2 (AuthNoPriv) or 3 (AuthPriv). Currently only MD5 (Message Digest Algorithm) and SHA (Secure Hash Algorithm, if the agent is configured for SNMPv3 with secLevel 2 (AuthNoPriv) or 3 (AuthPriv)) are used.

Default: MD5

-x *privPassword*

Specifies the privacy (encryption) password if the agent is configured for SNMPv3 with secLevel 3 (AuthPriv).

Default: none

-X DES | AES | 3DES

Specifies the privacy protocol if the SNMPv3 user is configured with secLevel 3 (AuthPriv). Specify DES for Data Encryption Standard, AES for Advanced Encryption Standard using cryptographic keys of 128 bits (AES128), and 3DES for Triple Data Encryption Standard.

Default: none

-m FIPS_mode

Controls the FIPS mode of operation. Accepted values are 0, 1, and 2.

0

Indicates Non-FIPS mode.

1

Indicates FIPS co-existence mode.

2

Indicates FIPS only mode.

-r *retries*

Specifies the number of retries.

Default: 3

-t *timeout*

Specifies the duration before the SNMP receiver considers the request as timed out.

Default: 10 seconds

-d logLevel

Specifies the log level of the SNMP messages. Accepted values are 0 to 5.

0

Logs fatal messages.

1

Logs critical messages.

2

Logs warning messages.

3

Logs informational messages.

4

Logs all of the messages.

5

Logs all of the messages including debugging messages.

Default: 0

-f logfile

Specifies the name of the log file that contains error and debug information. The default log file name for most of the utilities is `sysedge_utility.log`

Default: none

facility

Specifies a facility to use. Supported values are the following:

- process
- logfile
- ntevent
- procgroup

command

Specifies the command and associated arguments. Supported commands are the following:

- add
- setstatus
- delete
- list

Note: For more information about the commands, see *edgwatch Commands for Windows Event Monitoring*.

edgwatch Commands for Windows Event Monitoring

This section describes the *edgwatch* commands for the NT Event Monitor table:

```
add ntEventMonIndex ntEventMonFlags EventMonLog ntEventMonTypeFilter
"ntEventMonSrcFilter" "ntEventMonDescFilter" "ntEventMonDescr" "ntEventMonAction"
setstatus ntEventMonIndex status
delete ntEventMonIndex
list
```

ntEventMonIndex

Specifies the row in the NT Event Monitor table. Rows are indexed starting at 1. An index of 0 is not permitted. Because SNMP does not include a Create PDU type, new table entries are created as side effects of setting the columnar values for non-existent rows. Therefore, this value is required for add operations to specify the index (of an unused row) to use for row creation.

ntEventMonFlags

Specifies the hexadecimal flags (for example, 0x00001) that direct the additional behavior for this entry.

ntEventMonLog

Specifies the event log to monitor, which can be one of the following:

- Application
- System
- Security
- DirService (for Directory Service)
- DnsServer (for DNS Service)
- FileRepService (for File Replication Service)

ntEventMonTypeFilter

Specifies the event type to match for this entry. The following are valid types:

- Error
- Warning
- Information
- Success
- Failure

- All

Type All indicates that all event types should match.

"ntEventMonSrcFilt"

Specifies the regular expression to apply when scanning the Event Source attribute for each event in the log. You must enclose this value in quotation marks ("..").

"ntEventMonDescFilt"

Specifies the regular expression to apply when scanning the Event Description attribute for each event in the log. You must enclose this value in quotation marks ("..").

"ntEventMonDescr"

Specifies a quoted string, 0 to 128 characters in length, that contains a description of the file being monitored and a severity level for this event.

"ntEventMonAction"

Specifies a quoted string, 0 to 256 characters in length, that specifies the full path of the command (with any parameters) that runs when the entry is matched and a trap is sent. If the string is empty or not specified, no action will be taken.

Status

Specifies the Status textual convention value to use in setting the status of a row in the NT Event Monitor table when used with the setstatus operation. The Status parameter can take on one of the following values. Values can be either the assigned integer values or the actual spelled out status text. You specify these values when setstatus is specified.

- active(1)
- notInService(2)
- notReady(3)
- destroy(6)

Sample Uses of edgewatch for Monitoring Windows Events

This section provides examples for using the edgewatch utility to monitor Windows events.

Example: List Entries in the NT Event Monitor Table

The following example lists the contents of the agent's NT Event Monitor table:

```
edgewatch -v 1 -h fe80:ab01::901:bdef -c public -o ntevent list

edgewatch -v 2c -h 127.0.0.1 -c public -o ntevent list

edgewatch -v 3 -h fe80:ab01::901:bdef -s 3 -u userName -A authProtocol -a
authPassword -X encryptProtocol -x privPassword -o ntevent list
```

Example: Add an NT Event Monitor Entry

The following example adds a new entry to an agent's NT Event Monitor table at table index 5 to search for login failures on a Windows system.

```
edgewatch -v 1 -h 127.0.0.1 -c private -o ntevent add 5 0x0 Security Failure ".*"
".*" "Failed login attempt - WARNING" "\\local\\bin\\mail2admin.exe"

edgewatch -v 2c -h fe80:ab01::901:bdef -c private -o ntevent add 5 0x0 Security
Failure ".*" ".*" "Failed login attempt - WARNING" "\\local\\bin\\mail2admin.exe"

edgewatch -v 3 -h fe80:ab01::901:bdef -s 3 -u userName -A authProtocol -a
authPassword -X encryptProtocol -x privPassword -o ntevent add 5 0x0 Security
Failure ".*" ".*" "Failed login attempt - WARNING" "\\local\\bin\\mail2admin.exe"
```

This example also instructs the agent to run the \\local\\bin\\mail2admin.exe script when the agent finds a match.

Example: Delete an NT Event Monitor Entry

The following example deletes an entry from an agent's NT Event Monitor table at table index 5:

```
edgewatch -v 1 -h 127.0.0.1 -c private -o ntevent delete 5

edgewatch -v 2c -h fe80:ab01::901:bdef -c private -o ntevent delete 5

edgewatch -v 3 -h 127.0.0.1 -s 3 -u userName -A authProtocol -a authPassword -X
encryptProtocol -x privPassword -o ntevent delete 5
```

Example: Disable an NT Event Monitor Entry

The following example disables the NT Event Monitor table entry at table index 5 by setting that entry's status to `notInService(2)`. The entry will remain in the table, but the agent will not scan the event log for matches unless the entry's status is returned to `active(1)`:

```
edgewatch -v 1 -h 127.0.0.1 -c private ntevent setstatus 5 2
```

```
edgewatch -v 2c -h fe80:ab01::901:bdef -c private ntevent setstatus 5 2
```

```
edgewatch -v 3 -h 127.0.0.1 -s 3 -u userName -A authProtocol -a authPassword -X  
encryptProtocol -x privPassword -o ntevent setstatus 5 2
```

2

Corresponds to the Row Status textual convention value `notInService(2)`.

Removing NT Event Monitoring Entries

To stop the self-monitoring of a particular Windows event log, you must remove that entry from the NT Event Monitor table. NT Event Monitor table entries are stored in the `sysedge.mon` file to make sure that they will not be lost when the CA eHealth SystemEDGE agent restarts. In addition, the `watch ntevent` directives in the `sysedge.cf` file will create an NT Event Monitor entry whenever the CA eHealth SystemEDGE agent starts. This row creation results in a new NT Event Monitor table entry that will be stored in `sysedge.mon`. Thus, permanent removal of an NT Event Monitor Table entry requires two steps:

1. Remove the entry from the `sysedge.cf` file.
2. Remove the entry from the NT Event Monitor table.

Removing Entries from the `sysedge.cf` File

If you configured an entry by adding a `watch ntevent` directive to the `sysedge.cf` file, you must remove it from that file as part of removing the entry from the table. If you do not remove the `sysedge.cf` directive, the entry will be recreated the next time the CA eHealth SystemEDGE agent is restarted.

Removing Entries with the edgemon Utility

To remove an entry from the NT Event Monitor Table, use the edgemon utility to delete the entry. The following examples delete row 14 from the NT Event Monitor table on host 143.45.0.12. After deletion the row will be removed both from memory and from the sysedge.mon file.

```
edgemon -v 1 -h 143.45.0.12 -c private -o ntevent delete 14
```

```
edgemon -v 2c -h 143.45.0.12 -c private -o ntevent delete 14
```

```
edgemon -v 3 -h fe80:ab01::901:bdef -s 3 -u userName -A authProtocol -a  
authPassword -X encryptProtocol -x privPassword -o ntevent delete 14
```

Remove Entries Manually

In some cases, you cannot use the edgemon utility to delete NT Event Monitor Table entries. For example, if you have configured the CA eHealth SystemEDGE agent to disallow SNMP Set operations, the edgemon utility will not work. In this case, you must remove the entry from the NT Event Monitor Table by editing the sysedge.mon file and removing the entry from it. Because this is an active file, you must stop the CA eHealth SystemEDGE agent before you edit it. For more information about the format of this file, see the appendix "Adding Self-Monitoring Entries to the sysedge.mon File."

To delete row 14 manually

1. Stop the CA eHealth SystemEDGE agent.
2. Open sysedge.mon for editing, delete the entry for nteventmon row 14, and save the file.
3. Open sysedge.cf for editing, delete the entry for nteventmon row 14 if it exists, and save the file.
4. Restart the CA eHealth SystemEDGE agent.

Chapter 14: Configuring History Collection

This chapter describes the CA eHealth SystemEDGE agent's history-sampling capability. It also explains how you can instruct the agent to monitor and store the values of MIB variables over time for future retrieval by a manager.

This section contains the following topics:

[History Collection](#) (see page 311)

[History Control Table and the Data Table](#) (see page 312)

[View the History Control Table with CA eHealth AdvantEDGE View](#) (see page 316)

[Configuring the History Control Table](#) (see page 317)

History Collection

The CA eHealth SystemEDGE agent can track the values of various integer-based MIB objects (counters, gauges, and so on) over time and store them for later retrieval. This functionality, commonly referred to as history sampling, can greatly reduce the amount of management station polling across the network.

Instead of having to continuously poll to collect the value of a MIB variable over time, the manager can instruct the agent to sample and store the values. The management system can contact the agent periodically to upload the complete history of samples. The agent will continue to sample and store the specified MIB values even during periods of network outage when the management system cannot communicate with the agent.

History Sampling

The agent uses two SNMP MIB tables to provide the history capability:

- History Control table
A control table for defining the data-collection functions.
- History table
A data table for storing the actual data samples.

The control table enables you to dynamically configure the agent to sample and store the values of any integer-based MIB variable under its control. The data table stores the values for future retrieval.

To perform baselining and trend analysis, you can configure the CA eHealth SystemEDGE agent to monitor and store the values for swapCapacity, for example, by configuring the agent to sample the value every 5 minutes, and to store the most recent 144 samples. Then, once every 12 hours, the Management System can upload the entire 144 samples to obtain the values for swapCapacity that were collected during the preceding 12-hour period. (The swapCapacity variable of the Systems Management MIB specifies the percentage of swap space currently in use).

History Control Table and the Data Table

The History Control table contains parameters that describe the data that the agent will sample and store in the History table. Each row of the History Control table assigns values to the parameters (columnar objects) of the table and thereby defines a specific data-collection function. One or more rows (stored samples) in the History table are associated with that single control row.

Each control table row is assigned a unique value of empireHistoryCtrlIndex. A row defines the data-collection function by specifying the object-instance to be sampled, how often to sample (in multiples of 30 seconds), and the number of samples to keep (buckets). Associated with each data-collection function (row of the control table) is a set of rows of the History table. Each row of the History table, which is also named a bucket, holds the value of the specified MIB object that was gathered during one sampling interval.

As each sampling interval occurs, the agent adds a new row to the History table with the same empireHistoryIndex as the other rows for this data-collection function. Each new row corresponds to the single row in the History Control table, and has an empireHistorySampleIndex which is one greater than the SampleIndex of the previous sample.

Columns of the History Control Table

The following list describes the columns of the History Control table:

empireHistoryCtrlIndex

Specifies an integer (1 to MAXINT) that uniquely identifies the entry in the table.

empireHistoryCtrlDescr

Describes the data-collection function defined by this entry.

empireHistoryCtrlInterval

Specifies an integer value indicating how often (in seconds) the agent should sample the MIB variable.

Note: The interval must be a multiple of 30 seconds.

empireHistoryCtrlObjID

Specifies the complete object-instance identifier of the MIB variable to be sampled.

Note: You must include the *instance* portion of the object identifier (for example, .0 for scalars). The object instance must exist and must be contained within the Systems Management MIB.

For example, any supported (INTEGER-based) object in MIB-II, the Host Resources MIB, or the Systems Management MIB is valid. Objects should be of integer type including counter, gauge, integer, or enumerated integer.

empireHistoryCtrlObjType

Specifies the ASN1/SNMP type of the MIB variable that the agent is sampling.

empireHistoryCtrlBucketsReq

Specifies the requested number of discrete samples to be saved in the History table. Depending on available resources, the agent sets the empireHistoryBucketsGrant column as close to this value as possible.

empireHistoryCtrlBucketsGrant

Specifies the actual number of discrete samples that the agent will save in the History table for the data-collection function defined by this entry. The agent will keep the most recent BucketsGrant number of samples.

empireHistoryCtrlLastCall

Specifies the last time, based on sysUptime, that a sample was taken on behalf of this entry.

empireHistoryCtrlCreateTime

Specifies the time, based on sysUptime, at which this history sampling function was created.

empireHistoryCtrlStatus

Specifies the status of the entry, which can be one of the following:

- active
- notInService
- notReady
- createAndGo
- createAndWait

- destroy

These values are defined in the SNMPv2 SMI RowStatus textual convention.

Setting the status to destroy(6) causes the agent to discontinue history sampling for this entry, and to delete both this row and the corresponding data sample rows in the History table.

Note: For more information about the History Control table, see the specification `empire.asn1` in the `doc` subdirectory of the CA eHealth SystemEDGE agent's installation and the chapter "Systems Management MIB."

Columns of the History Table

Note: For more information about the data-storage table, see the specification (`empire.asn1` in the `doc` subdirectory of the CA eHealth SystemEDGE agent's installation) and the chapter "Systems Management MIB."

Following are the columns of the History table:

empireHistoryIndex

Identifies the row in the History Control table of which this sample is a part. It has the same value as the `empireHistoryCtrlIndex` for the corresponding entry in the control table.

empireHistorySampleIndex

Specifies the index that uniquely identifies the particular sample this represents among all samples associated with the same history control entry. This index starts at 1 and increases by one as each new sample is stored.

empireHistoryStartTime

Specifies the time, based on `sysUptime`, at which the first sample was taken.

empireHistorySampleTime

Specifies the time, based on `sysUptime`, at which this particular sample was taken.

empireHistoryValue

Specifies the current value of the MIB variable taken at this sample.

Optimizing Row Creation

The following table describes the scalar MIB objects that you can use with the History table to optimize row creation:

histCtrlUnusedIndex

Returns an unused index number for the History table when you perform an SNMP Get on the variable.

Permissions: Read-only

histCtrlMatchDescr

Determines the index number that corresponds to a particular entry description. Perform an SNMP Set of this MIB object to cause the agent to search through entries in the History table and put the index value of the last entry whose description matches in the histCtrlMatchIndex MIB object.

Permissions: Read-write

histCtrlMatchIndex

Matches a particular entry description with its index number when used with histCtrlMatchDescr.











Permissions: Read-only

View the History Control Table with CA eHealth AdvantEDGE View

If you are using CA eHealth AdvantEDGE View, you can query a system for History information by selecting the system you want to monitor from the System list, selecting History Collection from the Configuration list, and clicking the Configuration icon.

Note: For more information, see the *CA eHealth AdvantEDGE View Web Help*.

The following illustration shows the sample CA eHealth AdvantEDGE View History Control table:

| Index | Description | Interval (secs) | ObjID | Type | Requested | Granted | Last Call | Create | Row Status |
|--|----------------------|-----------------|--|---------|-----------|---------|-------------------|---------|---|
| 10  | Swap Cap | 60 | swapCapacity 1.3.6.1.4.1.546.1.1.7.8.30.0 | Integer | 20 | 20 | 49 days, 18:11:06 | 0:00:03 |  |
| 11  | test oid | 60 | totalSwapSpace 1.3.6.1.4.1.546.1.1.7.8.29.0 | Integer | 60 | 60 | 49 days, 18:11:06 | 0:00:03 |  |
| 12  | test oid | 60 | swapCapacity 1.3.6.1.4.1.546.1.1.7.8.30.0 | Integer | 60 | 60 | 49 days, 18:11:06 | 0:00:03 |  |
| 15  | Run Queue Length | 30 | runQLen 1.3.6.1.4.1.546.1.1.7.8.4.0 | Gauge | 100 | 100 | 49 days, 18:11:36 | 0:00:03 |  |
| 1000  | swapCapacity history | 60 | swapCapacity 1.3.6.1.4.1.546.1.1.7.8.30.0 | Integer | 60 | 60 | 49 days, 18:11:06 | 0:00:03 |  |
| Add History Entry | | | | | | | | | |

Configuring the History Control Table

You can control which MIB objects the CA eHealth SystemEDGE agent stores in the History table by adding, deleting, or modifying entries in the History Control table.

You can configure the History Control table in one of these ways:

- **Dynamically**

Use SNMP commands from a management system, such as CA eHealth AdvantEDGE View, to modify the table.

Note: For more information, see Dynamic Configuration During Operation in the chapter “Configuring History Collection.”

- **At start-up initialization**

Specify the entries for the History Control table in the CA eHealth SystemEDGE agent configuration file `sysedge.cf`.

Note: For more information, see Initial Configuration During Start Up.

Initial Configuration During Start-Up

On startup, the CA eHealth SystemEDGE agent reads the `sysedge.cf` configuration file and uses the `emphistory` directive to specify initial entries to the History Control table. You can specify MIB object for the agent to monitor by adding appropriate entries to `sysedge.cf`.

emphistory Command--Add Entries to History Control Table

You can use the `emphistory` keyword in the configuration file to add entries in the History Control table as follows:

```
emphistory index interval objid buckets 'description'
```

index

Specifies the row number in which the agent will create the entry.

interval

Specifies the interval (in seconds) at which the agent will sample the object's value. The interval must be a multiple of 30 seconds.

objid

Specifies the object instance within the agent's MIB whose value should be sampled. You can specify the OID using the complete dotted-decimal value (for example, 1.3.6.1.2.1.25.1.5.0) or the symbolic MIB name (for example, `hrSystemNumUsers.0`). In both cases, you must specify the object instance, which is typically zero for non-tabular MIB variables.

buckets

Specifies the number of buckets, or samples, that the agent will store internally. The agent stores the last buckets number of samples. As the agent takes each new sample, it deletes the oldest sample.

'description'

Specifies a quoted string (0 to 128 characters in length) that indicates the description field for this entry.

emphistory Command Example

The following entry instructs the agent to sample the value of the object MemInUse through table index 15 every 120 seconds and to store the last 60 samples:

```
emphistory 15 120 memInUse.0 60 'MemInUse history'
```

Dynamic Configuration During Operation

You can use the emphistory command-line utility to configure entries in the History Control table and to retrieve data samples from the History Control table dynamically. The emphistory utility is located in the bin subdirectory of the CA eHealth SystemEDGE agent's installation. You can also add history table entries to the agent configuration file manually.

Note: For more information about how to add history table entries to the agent configuration file, see [emphistory Utility--Manage Entries in History Control Table](#).

emphistory Utility--Manage Entries in History Control Table

Use the emphistory utility from the command line to add, delete, set the status of, or list entries in the History Control table. You can also use it (with the dump operation) to retrieve stored data samples from the History table.

You can use the emphistory utility as follows:

```
emphistory [-h hostname | ip_addr] [-p port] [-c community]
          [-v 1 | 2c | 3] [-u secName] [-s secLevel] [-n contextName]
          [-a authPassword] [-A MD5 | SHA]
          [-x privPassword] [-X DES | AES | 3DES]
          [-m FIPS_mode]
          [-r retries]
          [-t timeout] [-d logLevel] [-f logFile]
          [-o] [command]
```

-h *hostname* | *ipaddr*

Specifies the hostname or IP address of the system on which the agent is running. Accepts IPv4 and IPv6 addresses.

Default: localhost

-p *port*

Specifies the UDP port that the agent is running on (for example, 1691).

Default: 161

-c *community*

Specifies a community string that the agent uses. Valid for SNMPv1 and SNMPv2c only.

Note: A read-write community string has to be specified for snmpset.

Default: public

-v 1 | 2c | 3

Indicates the version of SNMP that the agent is running. Specify 1 for SNMPv1, 2c for SNMPv2c, or 3 for SNMPv3.

Default: 1

-u *secName*

Specifies the name of the SNMPv3 secure user.

Default: none

-s *secLevel*

Specifies one of the following security levels for SNMPv3 communication:

- 1 - noAuthNoPriv
- 2 - AuthNoPriv

- 3 - AuthPriv (SNMPv3 only)

Default: none

-n contextName

Specifies the context name used by the agent if it is configured as SNMPv3.

Note: This option is not required for SNMPv3 communication.

Default: none

-a authPassword

Specifies the authentication password if the agent is configured for SNMPv3 with secLevel 2 (AuthNoPriv) or 3 (AuthPriv).

Default: none

-A MD5 | SHA

Specifies the authentication protocol to be used by SNMPv3. This is required if the SNMPv3 user is configured with secLevel 2 (AuthNoPriv) or 3 (AuthPriv). Currently only MD5 (Message Digest Algorithm) and SHA (Secure Hash Algorithm, if the agent is configured for SNMPv3 with secLevel 2 (AuthNoPriv) or 3 (AuthPriv)) are used.

Default: MD5

-x privPassword

Specifies the privacy (encryption) password if the agent is configured for SNMPv3 with secLevel 3 (AuthPriv).

Default: none

-X DES | AES | 3DES

Specifies the privacy protocol if the SNMPv3 user is configured with secLevel 3 (AuthPriv). Specify DES for Data Encryption Standard, AES for Advanced Encryption Standard using cryptographic keys of 128 bits (AES128), and 3DES for Triple Data Encryption Standard.

Default: none

-m FIPS_mode

Controls the FIPS mode of operation. Accepted values are 0, 1, and 2.

0

Indicates Non-FIPS mode.

1

Indicates FIPS co-existence mode.

2

Indicates FIPS only mode.

-r *retries*

Specifies the number of retries.

Default: 3

-t *timeout*

Specifies the duration before the SNMP receiver considers the request as timed out.

Default: 10 seconds

-d *logLevel*

Specifies the log level of the SNMP messages. Accepted values are 0 to 5.

0

Logs fatal messages.

1

Logs critical messages.

2

Logs warning messages.

3

Logs informational messages.

4

Logs all of the messages.

5

Logs all of the messages including debugging messages.

Default: 0

-f *logfile*

Specifies the name of the log file that contains error and debug information. The default log file name for most of the utilities is `sysedge_utility.log`

Default: none

command

Specifies the command and associated arguments. Supported commands are the following:

- add
- setstatus
- delete
- dump
- list

Note: For more information about the above commands, see *emphistory Commands for Managing Entries in History Control Table*.

Important! The following usage of the *emphistory* utility is deprecated. The use of the above argument format is strongly recommended, as the following format will not be supported in the future:

```
emphistory add ipaddr[:port] commstr ctrlIndex objid interval buckets
['description']
emphistory delete ipaddr[:port] commstr ctrlIndex
emphistory list ipaddr[:port] commstr
emphistory setstatus ipaddr[:port] commstr ctrlIndex status
emphistory dump ipaddr[:port] commstr ctrlIndex
```

emphistory Commands for Managing Entries in the History Control Table

The *emphistory* command specifies the type of command to be carried out from the available list of commands: add, setstatus, delete, dump, and list.

Depending on the type of command, you may be required to specify additional parameters to complete the command. The syntax for the commands is as follows:

```
add [index] [objid] [interval] [buckets] ["description"]

setstatus [index] [status]

delete [index]

dump [index]

list
```

index

Specifies the row number for this entry in the agent's History Control table.

Rows are indexed starting at 1. When you specify the dump operation, the agent retrieves all data samples from the History table that correspond to the entry in the History Control table identified by index. If you specify an index value of -1 for dump operations, the agent retrieves all of the contents of the History table.

You must specify an index value for add operations to specify the particular MIB table index of an unused row to use for row creation.

objid

Specifies the complete object-instance identifier of the MIB variable that the agent will sample and store in the History Control table. You can specify the OID using the complete dotted-decimal value (for example, 1.3.6.1.2.1.25.1.5.0) or the symbolic MIB name (for example, hrSystemNumUsers.0). You must provide the instance portion of the OID (that is, .0 for scalars). The object-instance must exist and must be of integer type (which includes counter, gauge, integer, or enumerated integer).

interval

Specifies the interval in seconds between successive samples; this value must be a multiple of 30 seconds.

buckets

Specifies the number of discrete data samples to be saved in the History table on behalf of this control entry. Each sample, named a bucket, contains the snapshot value of the MIB variable, the time at which the sample was created, the sample index, and an index that corresponds to the entry in the History Control table that defines the data-collection function. The History table stores the most recent buckets number of samples.

status

Specifies the RowStatus textual convention value to use in setting the status of a row in the History Control table when used with the setstatus operation. The Status parameter can be one of the following values:

- (1)active
- (2)notInService
- (6)destroy

Setting the row status to destroy(6) causes the agent to discontinue history sampling for that entry, and to delete the row in the History Control table and the corresponding data sample rows in the History table.

'description'

(Optional). Specifies a description for the row. If you specify a value, emphistory uses the supplied string instead of the default History Control entry description string. If you include a description, you must enclose it within single quotation marks (' '). The description must be less than 128 characters, not including the quotation marks. Longer strings are truncated.

Important! The following usage of the emphistory utility is deprecated. We strongly encourage using the new argument, as the following format will not be supported in the future.

```
emphistory add ipaddr[:port] commstr ctrlIndex objid interval buckets  
['description']
```

emphistory Utility Examples

This section provides examples for using the emphistory utility. The agent is assumed to be running with default port 161.

Example: List Entries in the History Control Table

The following examples list the contents of the agent's History Control table:

```
emphistory -h 127.0.0.1 -v 1 -c public -o list
```

```
emphistory -h 127.0.0.1 -v 3 -u username -s 3 -A MD5 -a authPassword -X DES -x  
encryptPassword -o list
```

Example: Add a History Control Entry

The following examples add a new control entry at table index 5 to the agent's History Control table. This control entry instructs the agent to sample the ifInOctets.1 MIB object-instance every 60 seconds and to store the most recent 10 samples:

```
emphistory -h 127.0.0.1 -v 1 -c private -o add 5 ifInOctets.1 60 10
```

```
emphistory -h 127.0.0.1 -v 3 -u username -s 3 -A MD5 -a authPassword -X DES -x  
encryptPassword -o add 5 ifInOctets.1 60 10
```

Example: Delete a History Control Entry

The following entries delete the History Control table entry at table index 3:

```
emphistory -h 127.0.0.1 -v 1 -c private -o delete 3
```

```
emphistory -h 127.0.0.1 -v 3 -u username -s 3 -A MD5 -a authPassword -X DES -x  
encryptPassword -o setstatus 3 destroy
```

Note: These entries also instruct the agent to delete the stored data samples in the History table that correspond to this control entry.

Example: Set the Row Status of a Control Entry

The following examples disable the control entry in the History Control table at table index 5, but save the corresponding stored samples in History table:

```
emphistory -h 127.0.0.1 -v 1 -c private -o setstatus 5 2
```

```
emphistory -h 127.0.0.1 -v 3 -u username -s 3 -A MD5 -a authPassword -X DES -x  
encryptPassword -o setstatus 5 2
```

2

Corresponds to the RowStatus textual convention value notInService(2).

Example: Retrieve Stored Data Samples

The following examples retrieve all the stored data samples that correspond to the data-collection function defined in row 5 of the History Control table:

```
emphistory -h 127.0.0.1 -v 1 -c private -o dump 5
```

```
emphistory -h 127.0.0.1 -v 3 -u username -s 3 -A MD5 -a authPassword -X DES -x  
encryptPassword -o dump 5
```

The following examples retrieve all the stored samples for all control entries using -1. This command retrieves the entire contents of the History table:

```
emphistory -h 127.0.0.1 -v 1 -c private -o dump -1
```

```
emphistory -h 127.0.0.1 -v 3 -u username -s 3 -A MD5 -a authPassword -X DES -x  
encryptPassword -o dump -1
```

Example: Deprecated Old Usage Examples

`emphistory delete ipaddr[:port] commstr ctrlIndex`

`emphistory list ipaddr[:port] commstr`

`emphistory setstatus ipaddr[:port] commstr ctrlIndex status`

`emphistory dump ipaddr[:port] commstr ctrlIndex`

Chapter 15: Adding Custom MIB Objects

The CA eHealth SystemEDGE agent provides a mechanism for extending the Systems Management MIB to include information about your systems and applications. Using this feature, you can extend the agent to manage your system environment more effectively. You can also design application-specific MIB variables to manage your applications using SNMP without implementing SNMP support within the application source.

This section contains the following topics:

[Systems Management MIB Extension Group](#) (see page 327)

[Features of the Extension Group](#) (see page 328)

[Configuring Extension Variables](#) (see page 329)

[Extension Examples](#) (see page 330)

[Writing Extension Scripts](#) (see page 332)

[Using Extension Variables with Your Management Software](#) (see page 333)

[Recommendations for Using Extensions](#) (see page 334)

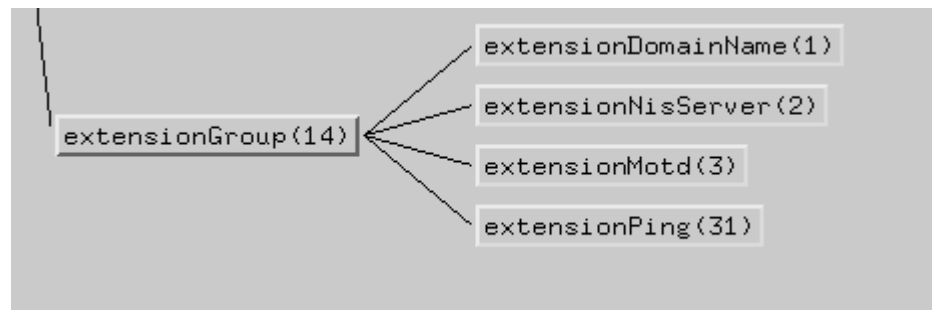
Systems Management MIB Extension Group

The CA eHealth SystemEDGE agent provides this extension support through the Extension group (extensionGroup) of the Systems Management MIB. This group contains unspecified scalar MIB variables that you can configure. In response to a SNMP GetRequest for one of these variables, the CA eHealth SystemEDGE agent invokes the command that you specify for the variable and returns the value that the command returns. Using the SNMP Set operation, you can also pass parameters to a command.

The Extension group is supported on both UNIX and Windows systems; on Windows, however, you may also configure the agent to report on performance and configuration data available in the Windows registry. To support this reporting, the CA eHealth SystemEDGE agent also provides a Windows registry extension group.

Note: For more information about this group, see the chapter “Configuring Windows Event Monitoring.”

The following illustration shows four sample extension variables distributed with the agent:



Features of the Extension Group

The Extension Group of the Systems Management MIB is located at OID 1.3.6.1.4.1.546.14. This group provides space for 2^{32} user-defined scalar MIB variables. These new variables are numbered 1 through 2^{32} and are referenced as any other scalar MIB object. For example, extension object 1 is referred to in Get and Set request messages as 1.3.6.1.4.1.546.14.1.0. In all cases, extension variables use the object-instance identifier of .0.

An extension variable can be any valid base SNMP type, including the following:

- Integer
- Counter
- Gauge
- Octetstring
- TimeTicks
- ObjectId
- IPAddress

You can specify variables as Read-Only or Read-Write.

Note: The CA eHealth SystemEDGE agent logs extension command invocations at syslog level LOG_DEBUG. It logs extension-command invocation errors at syslog level LOG_WARNING.

For more information about configuring the syslog utility, see the appendix “Using the syslog Facility.” For more information about starting the agent with its debugging options turned on, see the chapter “Starting the CA eHealth SystemEDGE Agent.”

Configuring Extension Variables

You can configure extension variables in the CA eHealth SystemEDGE agent configuration file, `sysedge.cf`. On startup, the agent configures these values and verifies whether the program or command associated with the variable is executable. Within the configuration file, the keyword `extension` defines an extension variable. The next section describes how to use this keyword.

extension Keyword--Add Entries to the Extension Group

You can use the extension keyword to add entries in the extension group as follows:

```
extension LeafNumber Type Access 'Command'
```

LeafNumber

Specifies an extension variable number in the range of 1 through 2³² defined by this entry.

Type

Specifies the SNMP type for this entry. The supported types are as follows:

- Integer
- Counter
- Gauge
- Octetstring
- TimeTicks
- Objectid
- IPAddress

Access

Specifies the access type, which can be either Read-Only or Read-Write.

'Command'

Specifies a quoted string, 0 to 256 characters in length, that specifies the full path of the command (with any parameters) which runs when this variable is accessed through either a Get, GetNext or Set request. If the command file is not currently accessible, the configuration of this variable will fail.

Additional Parameters

In addition to the parameters that you specify in the configuration file as part of the extension command, the CA eHealth SystemEDGE agent passes some additional parameters to help you decide how to treat the request. These parameters are passed after any parameters that you specified in quotation marks as part of the extension statement.

The following list describes the additional parameters that can be passed to the extension command:

Leafnumber

Specifies an integer value that represents the leaf number of this variable (1 through 232). If you have a single script that supports multiple extension values you can use this parameter to determine which variable is being requested.

Request-Type

Indicates the type of SNMP request, which can be one of the following:

- Get
- GetNext
- Set

The request type is passed with all letters capitalized.

Set-Value

Specifies a string that contains the value that passed in a Set request. Use this string in your extension program to modify the current value of the variable in such a way that a future Get or GetNext can return this value.

Extension Examples

The CA eHealth SystemEDGE agent includes several sample extension variables. These examples are defined in the sample sysedge.cf file. The scripts that implement these examples are included in the contrib subdirectory of the CA eHealth SystemEDGE agent's installation. These variables are also defined in the Systems Management MIB (empire.asn1 in the doc subdirectory of the CA eHealth SystemEDGE agent's installation).

Important! Before you add your own extensions, carefully review the examples in this chapter and in the Systems Management MIB. For clarity, these examples include the appropriate configuration file extension commands.

You can add these extensions to your `/etc/sysedge.cf` (UNIX) or `%SystemRoot%\System32\sysedge.cf` (Windows) file to make them available to the CA eHealth SystemEDGE agent.

Example: DNS Domain (UNIX Only)

The following extension object returns the DNS domain name of the underlying system (as opposed to the Network Information System [NIS] domain name):

```
extension 1 OctetString ReadOnly /opt/EMPSysedge/contrib/getextension.sh
```

The instance-identifier of this object is 1.3.6.1.4.1.546.14.1.0.

Example: NIS Domain Name (UNIX Only)

The following extension object returns the NIS domain name of the underlying system (as opposed to the DNS domain name):

```
extension 2 OctetString ReadOnly /opt/EMPSysedge/contrib/getextension.sh
```

The instance-identifier of this object is 1.3.6.1.4.1.546.14.2.0.

Example: Remote Ping (UNIX and Windows)

Use the following extension objects to instruct the CA eHealth SystemEDGE agent to ping a remote host from the host where the agent is running. This can be a useful tool for diagnosing network connectivity problems and is a good example of how to use SNMP Set operations with extension variables.

For UNIX systems, enter the following in the `/etc/sysedge.cf` file:

```
extension 31 OctetString ReadWrite /opt/EMPSysedge/contrib/ping.sh
```

For Windows systems, enter the following in the `%SystemRoot%\System32\sysedge.cf` file:

```
extension 31 OctetString ReadWrite c:\sysedge\contrib\ping.bat
```

The instance-identifier of this object in both examples is 1.3.6.1.4.1.546.14.31.0.

To use this feature, first set the variable with the name or IP address of the destination you would like to ping. Then, when you get the variable, the agent returns the output from the ping attempt.

Writing Extension Scripts

The CA eHealth SystemEDGE agent places very few constraints on the operation of extension scripts. It does, however, require the following:

- All operations (Set, Get, or GetNext) must have output. The output of Set invocations should echo the value that was actually Set while the output of Get and GetNext should be the object's value. If there is no output, the SNMP query will fail.
- Output from extension scripts is only parsed up through the first newline character. If the first character is a newline, the output is considered NULL, causing the SNMP query to fail, and returning an error.

Note: Because the CA eHealth SystemEDGE agent runs as root or administrator, make sure that all commands and scripts use absolute pathnames, are fully debugged, and contain no ambiguous code or unnecessary options.

On Windows systems, you can use batch files for writing extension scripts. The agent can directly run those batch files. However, batch file functionality is severely limited. Use Perl and other scripting languages for Windows instead.

Testing Your Script: An Example

After you create an extension script, test it at the command line. The following example creates an OctetString extension on a UNIX system, tests its output, and then uses SNMP to return the value from the script.

To test the script

1. Add the following line to `/etc/sysedge.cf`:

```
extension 1 OctetString ReadWrite /opt/EMPSysedge/debugext.sh
```

Note: This example tests an extension script that is an OctetString. It is valid for UNIX operating systems. On Windows systems, you must call the interpreter in your action script command. For example, enter `perl.exe myscript.pl`.

2. Enter the following at the command line to create a file named `myset.txt`:

```
echo "1.3.6.1.4.1.546.14.1.0 04 debugSetString" > myset.txt
```

You now have a file named `myset.txt`. You are setting the OctetString(04) "debugSetString" to the OID 1.3.6.1.4.1.546.14.1.0.

3. Issue the SNMP Set by entering the following:

```
./snmpset private 127.0.0.1 < myset.txt
```

4. Retrieve the value of your debugSetString by entering the following:

```
./snmpget public 127.0.0.1 1.3.6.1.4.1.546.14.1.0 debugSetString
```

5. Test the setup by entering the following at the command line:

```
./debugext.sh 1 SET debugSetString2
```

Should return the value as being set as output

```
./debugext.sh 1 GET
```

Should return the value as set in the earlier SET call

Using Extension Variables with Your Management Software

All methods for incorporating extension variables into your management system software (for example, Cabletron Spectrum, HP OpenView, Sun Enterprise Manager, and so on) require you to edit and import MIB specification files. This guide does not discuss details of importing MIB specification files into management system software, but it does describe the two overall strategies that exist for incorporating extension variables:

- Edit `empire.asn1` to include the extension variables defined for your site.
- Edit a separate MIB file to include the extension variables defined for your site.

How to Edit `empire.asn1` for Extension Variables

Follow these steps to add your own extension variables to the Systems Management MIB (`empire.asn1`):

1. Create and debug the relevant extension scripts, and then configure them in the agent's configuration file, `sysedge.cf`, to include them.
2. Edit `empire.asn1` to include new extension MIB variables that exist under the `extensionGroup`.
3. Perform a test compile of `empire.asn1` to ensure that there are no syntax errors. This procedure is specific to your management system and MIB compiler.
4. Import the new `empire.asn1` file into your management system software. This procedure is specific to your management system and MIB compiler.

Note: If you do not reimport the MIB file, your management system software will not be able to access the new extension MIB objects.

How to Edit a Separate MIB Specification for Extension Variables

You can put extension variables in a separate MIB specification file for ease of updating. You can save time and effort by making changes to a MIB specification file, which is much smaller than `empire.asn1`; this avoids recompiling and reloading the entire `empire.asn1` file.

To add your extension variables to a separate MIB specification file

1. Create and debug the relevant extension scripts, and configure them in the agent's configuration file, `sysedge.cf`, to include them.
2. Edit your own MIB specification file (for example, `extensions.asn1`) to include new extension MIB variables that exist under the Systems Management Extension group.
3. Perform a test compile of `extensions.asn1` to check that there are no syntax errors. This procedure is specific to your management system and MIB compiler.
4. Import the `extensions.asn1` file into your NMS software. This procedure is specific to your management system and MIB compiler.

Note: If you do not reimport the MIB file, your management system software will not be able to access the new extension MIB objects.

Recommendations for Using Extensions

When you create CA eHealth SystemEDGE extensions, follow these guidelines:

- Do not edit `empire.asn1`. Use a separate MIB specification.
- Use CA eHealth SystemEDGE debugging (log file monitoring) to find output, plug-in, and extension problems.
- Use 128 characters or less for extension entries to meet the Windows limitation on command line length. Remember that the length includes paths, commands, and arguments.
- Do not use batch files. Instead use Windows scripting, VBScript, Perl, C, Shell, and so on.
- Watch the fork/exec time for new extensions. Extensions are synchronous and can block other actions.

Chapter 16: Adding Windows Registry and Performance MIB Objects

The CA eHealth SystemEDGE agent provides a powerful mechanism for extending the Systems Management MIB to include information from the Windows registry and performance counters. This information includes configuration data (which is typically viewed using the Windows program regedit) and performance data (which is typically viewed using the Windows program perfmon).

Using this feature, you can customize the CA eHealth SystemEDGE agent to return additional configuration and performance data for your systems and applications. For example, you can use the CA eHealth SystemEDGE agent to make many application registry entries that specify the configuration of the application available through SNMP. In addition, you can access performance statistics for many applications through the CA eHealth SystemEDGE agent.

This section contains the following topics:

[Systems Management MIB ntRegPerf Group](#) (see page 335)

[Windows Registry and Performance Functionality](#) (see page 336)

[Configuring Windows Registry and Performance Variables](#) (see page 339)

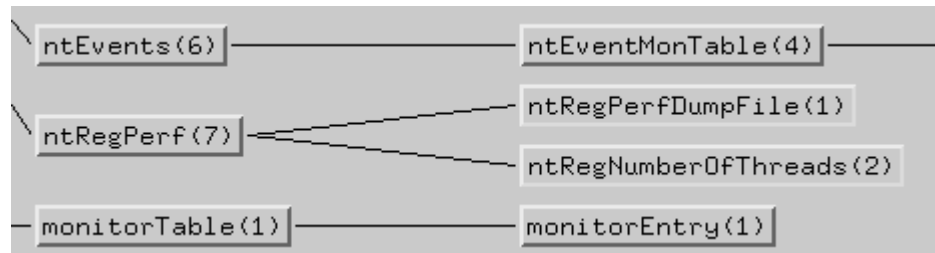
[Windows Registry and Performance Examples](#) (see page 341)

[Using Windows Registry and Performance Variables with Your Management Software](#) (see page 342)

Systems Management MIB ntRegPerf Group

The CA eHealth SystemEDGE agent provides the support for these additional configuration and performance parameters in the ntRegPerf group of the Systems Management MIB. This group contains 128 unspecified scalar MIB variables that you can configure. In response to a SNMP Get request for one of these variables, the CA eHealth SystemEDGE agent will read the Windows registry and return the value obtained.

The following illustration shows two sample ntRegPerf variables distributed with the CA eHealth SystemEDGE agent:



Windows Registry and Performance Functionality

The ntRegPerf group of the Systems Management MIB is located at OID 1.3.6.1.4.1.546.5.7. This group provides a space for up to 128 user-defined scalar MIB variables. These new variables are numbered 1 through 128 and are referenced just like any other scalar MIB object. For example, ntRegPerf object 1 is referred to in Get request messages as 1.3.6.1.4.1.546.5.7.1.0. In all cases, ntRegPerf variables use the object-instance identifier of .0.

An ntRegPerf variable can be any valid base SNMP type, including the following:

- Integer
- Counter
- Gauge
- OctetString
- TimeTicks
- ObjectId
- IPAddress

Registry Data

The CA eHealth SystemEDGE agent provides data from the standard configuration registry. This data is indexed by both a key name and a value. For example, the key SYSTEM\CurrentControlSet\Control\CrashControl and the value DumpFile identify a text string that describes the location of the system dump file. Only LOCAL_MACHINE registry data is supported. The following list matches the registry data types that the CA eHealth SystemEDGE agent supports with the preferred SNMP type.

REG_DWORD

Integer

REG_SZ

OctetString

REG_EXPAND_SZ

OctetString

REG_MULT_SZ

OctetString; only the first string is returned

Performance Data

The CA eHealth SystemEDGE agent provides access to performance counters in the performance registry by specifying the object and counter names. For instance, the object NWLink NetBIOS and counter Bytes Total/sec identifies the NetBIOS byte-counter statistic.

When using performance counters, you must carefully interpret the data. At this time, there are at least 27 different counter types in Windows. Most counters do not actually report the data in the format you would assume based on the name of the counter. Instead, most values are raw counters that require you to do some post-processing, such as dividing the difference of two samples by the elapsed time. If you see a counter name like Bytes Total/sec, the data you are really getting is a raw counter that can be used to calculate this rate value.

Familiarize yourself with Windows Performance counters before using this feature of the CA eHealth SystemEDGE agent. For more information, refer to the section on optimizing Windows in the Windows Resource Kit.

The following list matches the 4-byte performance data types that the CA eHealth SystemEDGE agent supports with the preferred SNMP types.

PERF_COUNTER_COUNTER

Counter or integer

PERF_COUNTER_RAWCOUNT

Counter, integer, or gauge

PERF_COUNTER_RAWCOUNT_HEX

Counter, integer, or gauge

PERF_SAMPLE_COUNTER

Counter, integer, or gauge

The following list matches the 8-byte performance data types that the CA eHealth SystemEDGE agent supports with the preferred SNMP types. Because SNMPv1 supports only 4-byte values, CA eHealth SystemEDGE will return only the least significant 4-bytes of data.

PERF_COUNTER_TIMER

Counter, integer, or gauge

PERF_COUNTER_BULK_COUNT

Counter, integer, or gauge

PERF_COUNTER_LARGE_RAWCOUNT

Counter, integer, or gauge

PERF_COUNTER_LARGE_RAWCOUNT_HEX

Counter, integer, or gauge

PERF_COUNTER_TIMER_INV

Counter, integer, or gauge

PERF_AVERAGE_BULK

Counter, integer, or gauge

PERF_100SEC_TIMER

Counter or integer

PERF_100SEC_TIMER_INV

Counter or integer

PERF_COUNTER_MULTI_TIMER

Counter or integer

PERF_COUNTER_MULTI_TIMER_INV

Counter or integer

PERF_100NSEC_MULTI_TIMER

Counter or integer

PERF_100NSEC_MULTI_TIMER_INV

Counter, integer, or gauge

PERF_ELAPSED_TIME

Integer

PERF_RAW_FRACTION

Integer or gauge

Unsupported Performance Data Types

Several counters with multiple samples and internal data is difficult to present in a single value and require significant post-processing. For that reason, CA eHealth SystemEDGE does not support the following counter types:

- PERF_COUNTER_QUEUELEN_TYPE
- PERF_COUNTER_TEXT
- PERF_COUNTER_NODATA
- PERF_SAMPLE_BASE
- PERF_AVERAGE_TIMER
- PERF_AVERAGE_BASE
- PERF_COUNTER_MULTI_BASE
- PERF_RAW_BASE
- PERF_COUNTER_HISTOGRAM_TYPE

Configuring Windows Registry and Performance Variables

Windows Registry and Performance variables are configured in the CA eHealth SystemEDGE agent configuration file, `sysedge.cf`. When the CA eHealth SystemEDGE agent starts, it configures these values and verifies that the value associated with each variable is accessible. If not, it prints an error message to `sysedge.log` and does not create the MIB object.

ntRegPerf Keyword--Add Entries to the ntregperf Group

Within the configuration file, the keyword `ntregperf` defines the `ntRegPerf` variable. You can use the `ntRegPerf` keyword to add entries in the `ntregperf` group as follows:

```
ntregperf LeafNumber Type Registry 'Key' 'Value'
```

or

```
ntregperf LeafNumber Type Performance 'Object' 'Counter' 'PerfInstance'
```

LeafNumber

Defines the `ntRegPerf` variable number, in the range of 1 through 128.

Type

Specifies the SNMP type for this entry, which can be one of the following:

- Integer
- Counter
- Gauge
- Octetstring
- TimeTicks
- Objectid
- IPAddress

Registry

Selects a configuration registry entry.

Performance

Selects a configuration registry entry.

'Key'

Specifies a quoted string, 0 to 512 characters in length, that specifies the registry key to be accessed.

'Value'

Specifies a quoted string, 0 to 128 characters in length, that specifies the registry key to be accessed.

'Object'

Specifies a quoted string, 0 to 512 characters in length, that specifies the performance object to be accessed.

'Counter'

Specifies a quoted string, 0 to 128 characters in length, that specifies the object's performance counter value to be accessed.

'PerfInstance'

Specifies the performance counter instance to be accessed; it should be equivalent to that listed in the Windows perfmon utility.

Windows Registry and Performance Examples

The CA eHealth SystemEDGE agent includes several sample ntRegPerf variables. These examples are defined in the sample sysedge.cf file. Before you add your own ntRegPerf extension, study these examples and their definitions in the Systems Management MIB (empire.asn1 in the doc subdirectory of the CA eHealth SystemEDGE agent's installation). For clarity, these examples include the appropriate configuration file ntRegPerf commands.

Example: CrashControl DumpFile

The following ntRegPerf object returns the path to the dump file:

```
ntregperf 1 OctetString Registry 'SYSTEM\CurrentControlSet\Control\CrashControl'
'DumpFile'
```

The object instance-identifier of this object is 1.3.6.1.4.1.546.5.7.1.0.

Example: Total Number of Threads

The following ntRegPerf object returns the total number of threads currently available in the system:

```
ntregperf 2 Gauge Performance 'Objects' 'Threads' '1'
```

The object instance-identifier of this object is 1.3.6.1.4.1.546.5.7.2.0.

Example: TCP Segments Sent/Sec

The following ntRegPerf object returns the total number of TCP segments that were transmitted by the system:

```
ntregperf 3 Counter Performance 'TCP' 'Segments Sent/sec' '1'
```

The object instance-identifier of this object is 1.3.6.1.4.1.546.5.7.3.0.

Using Windows Registry and Performance Variables with Your Management Software

There are several methods for incorporating ntRegPerf variables into your management system software (for example, CA Spectrum, HP OpenView, Sun Enterprise Manager, and so on), all of which require editing and importing MIB specification files. While the details of importing MIB specification files into management system software are beyond the scope of this guide, two overall strategies exist for incorporating ntRegPerf variables:

- Edit empire.asn1 to include the ntRegPerf variables defined for your site.
- Edit a separate MIB file to include the ntRegPerf variables defined for your site.

How to Edit empire.asn1 for ntRegPerf Variables

You can add your own ntRegPerf variables to the Systems Management MIB.

To add your own ntRegPerf variables to the Systems Management MIB

1. Create and debug the relevant ntRegPerf entries in the agent's configuration file, sysedge.cf, to include them.
2. Edit empire.asn1 to include new ntRegPerf MIB variables.
3. Perform a test compile of empire.asn1 to ensure there are no syntax errors. This procedure is specific to your management system and its corresponding MIB compiler.
4. Import the new empire.asn1 file into your management system software. This procedure is specific to your management system and MIB compiler.

Note: If you do not reimport the MIB file, your management system software will not be able to access the new MIB objects.

How to Add a Separate MIB Specification for ntRegPerf Variables

You can put ntRegPerf variables in a separate MIB specification file for ease of updating. You can save time and effort by making changes to a MIB specification file, which is much smaller than empire.asn1. This avoids recompiling and reloading the entire empire.asn1 file.

To create and debug the relevant ntRegPerf entries in the agent's configuration file, sysedge.cf, to include them

1. Edit your own MIB specification file (for example, ntregperf.asn1) to include the new ntRegPerf MIB variables that exist under the Systems Management MIB ntregperfGroup.
2. Perform a test compile of the ntregperf.asn1 file to make sure there are no syntax errors. This procedure is specific to your management system and its corresponding MIB compiler.
3. Import the ntregperf.asn1 file to your management system software. This procedure is specific to your network management station and its corresponding MIB compiler.

Note: If you do not reimport the MIB file, your management system software will not be able to access the new MIB objects.

Chapter 17: Deploying the CA eHealth SystemEDGE Agent

This chapter describes deployment options for the CA eHealth SystemEDGE agent.

This section contains the following topics:

[Introduction](#) (see page 345)

[Deploy CA eHealth SystemEDGE with CA eHealth AdvantEDGE View](#) (see page 346)

[Deploy CA eHealth SystemEDGE from the Web](#) (see page 347)

[Deploy CA eHealth SystemEDGE through Email](#) (see page 348)

[Third-Party Deployment Tools](#) (see page 348)

[How to Automate Deployment](#) (see page 348)

Introduction

Deploying the CA eHealth SystemEDGE agents can be challenging in large distributed environments. This chapter provides guidelines and suggestions for automating the deployment of the CA eHealth SystemEDGE agent.

For small numbers of systems, manual deployment may be advantageous because it requires little configuration or preparation. However, as the number of systems and locations grows, the effort required to manually deploy new software grows exponentially.

CA eHealth SystemEDGE includes a sample set of scripts that you can use to help automate its deployment. The scripts are in the contrib subdirectory of the CA eHealth SystemEDGE agent's installation. The contrib subdirectory also includes an ntdist.pl script. You can run this script from the command line on Windows systems, using Perl with the Win32 extension.

CA eHealth SystemEDGE also provides the following deployment options:

- Using CA eHealth AdvantEDGE View Agent Deployment
- Downloading the agent from a Web page
- Distributing the agent through email

Deploy CA eHealth SystemEDGE with CA eHealth AdvantEDGE View

If you are using CA eHealth SystemEDGE with CA eHealth AdvantEDGE View on Windows systems, you can automatically deploy CA eHealth SystemEDGE agents and CA eHealth AIMs through CA eHealth AdvantEDGE View Agent Deployment.

To deploy CA eHealth SystemEDGE and CA eHealth AIMs from CA eHealth AdvantEDGE View

1. Click Administration.

CA eHealth AdvantEDGE View displays the Administration page.

2. Click Agent Deployment.

CA eHealth AdvantEDGE View displays the AdvantEDGE View: Agent Deployment form.

Note: For more information about completing this form, see the *CA eHealth AdvantEDGE View User Guide* or the *CA eHealth AdvantEDGE View Web Help*.

How the Automated Deployment Works

CA eHealth AdvantEDGE View obtains a list of target systems to deploy, including information about deployment options for each system, checks for local files, and verifies that the correct files for deployment exist on the system from which you will deploy the software. If the deployment setup is invalid, CA eHealth AdvantEDGE View stops the deployment.

CA eHealth AdvantEDGE View next checks each target system to make sure it meets the deployment criteria. If the target system meets the installation criteria, CA eHealth AdvantEDGE View copies the installation files (based on the options you specified on the Agent Deployment form) to the target systems. If the system does not meet the installation criteria, CA eHealth AdvantEDGE View displays an error on the Deployment Results Summary and moves on to the next target system.

Note: For more information about the criteria, see the *CA eHealth AdvantEDGE View User Guide*.

Deploy CA eHealth SystemEDGE from the Web

You can also deploy the CA eHealth SystemEDGE agent, CA eHealth AIMs, and other modules to your systems from a Web page in CA eHealth AdvantEDGE View or from the CA eHealth Software Downloads Web page at <http://support.concord.com/support/secure/software/>.

To access the CA eHealth AdvantEDGE View download page

1. Click Agent Deployment.
CA eHealth AdvantEDGE View displays the SystemEDGE Deployment form.
2. Click the link in the Agent Downloads section of the form.
The Agent Downloads page appears and displays the products available for deployment.
3. Click the README link next to the package for installation instructions and the Package link to download the installation package.
4. Click Latest downloads from CA to display the Software Downloads page of the CA eHealth Support Web site:
<http://support.concord.com/support/secure/index.asp>.

Note: For more information about accessing the Software Downloads page, see the procedure that follows.

To access the Software Downloads page of the CA eHealth Support Web site

1. In your Web browser, go to <http://support.concord.com>.
2. Enter your user name and password, and then click Login.
3. Select downloads.
The Software Downloads page appears.
4. Select the product that you want to download.
A Web page appears, listing the available versions of that product.
5. Click Instructions for installation instructions and Software Package to begin downloading the CA eHealth SystemEDGE agent or CA eHealth AIMs to your system.

Deploy CA eHealth SystemEDGE through Email

You can deploy the CA eHealth SystemEDGE agent, CA eHealth AIMs, and other modules to your users through email. To do so, copy the installation package file, the *CA eHealth SystemEDGE Release Notes* and the *CA eHealth SystemEDGE User Guide* from your CA eHealth SystemEDGE installation media, and email them to the user. You can modify the readme.txt file as necessary for your deployment.

Third-Party Deployment Tools

For new systems, most vendors provide automated installation tools for installing third-party software at operating system installation time.

The following list describes the recommended programs for automating operating system installation:

- Sun JumpStart (Migration Kit): <http://www.sun.com>
- HP Ignite-UX: <http://www.hp.com/>
- Microsoft System Management Server (SMS): <http://microsoft.com/>
- Symantec Ghost: <http://www.symantec.com/>
- PowerQuest Drive Image: <http://www.powerquest.com/>
- Red Hat Kick Start: <http://www.redhat.com/>

The following list describes the recommended software deployment tools:

- Tivoli: <http://www.tivoli.com>
- Microsoft System Management Server: <http://www.microsoft.com/>
- HPOV Software Distributor:
<http://www.hp.com/openview/products/softdist.html>

Even if you do not want to implement a software distribution system, you can take some steps to help automate deployment of the CA eHealth SystemEDGE agent in a distributed environment.

How to Automate Deployment

Automating software deployment involves the following four steps:

1. Making software available to remote systems.
2. Installing software on remote systems.
3. Configuring software for distributed systems.

Making Software Available to Remote Systems

You can make software available to remote systems in a variety of ways, including through protocols, remote file systems, and email.

Using Protocols to Distribute Software

This section discusses the protocols that you can use to distribute software.

- **Hypertext Transfer Protocol (HTTP)**

HTTP is the set of rules for exchanging files (text, graphic images, sound, video, and other multimedia files) on the World Wide Web (WWW). Relative to the TCP/IP suite of protocols (which are the basis for information exchange on the Internet), HTTP is an application protocol. You can easily distribute CA eHealth SystemEDGE agents through the Web by placing them on a Web page for downloading and installation.
- **File Transfer Protocol (FTP)**

FTP, a standard Internet protocol, is the simplest way to exchange files between computers on the Internet. Like HTTP, which transfers Web pages and related files, and SMTP, which transfers email, FTP is an application protocol that uses TCP/IP. FTP is commonly used to transfer Web page files from the computer where they were created to the computer that acts as their server. FTP is also commonly used for downloading programs and other files to your computer from other servers.
- **Remote Copy Protocol (RCP)**

RCP enables you to integrate remote copy operations into your applications. An application can copy files between the local and remote system, or from one remote system to another. RCP commands support recursive file copying and can preserve the original time and date attributes of the file.
- **Remote Distribution (RDIST)**

RDIST maintains identical copies of files over multiple hosts. It preserves the owner, group, mode, and modification time of files and can update the running programs. Almost all versions of UNIX include RDIST, but most include a very old version, sometimes referred to as 4.2BSD rdist, rdist classic, or rdist version 3.
- **Network File System (NFS)**

NFS is a client/server file-sharing protocol that lets you view and optionally store and update files on a remote computer as though they were on your own computer. To use NFS, your system must have an NFS client, and the other computer must have the NFS server. Both systems must also have TCP/UDP/IP installed.

Note: Microsoft Windows includes client and server SMB protocol support.

- Server Message Block Protocol (SMB)

SMB provides a method for client applications on one computer to read and write to files on, and to request services from, server programs in a computer network. You can use SMB over the Internet on top of its TCP/IP protocol, or on top of other network protocols, such as IPX and NetBEUI.

Using the SMB protocol, an application (or the user of an application) can access files at a remote server, including the other resources, such as printers, mailslots, and named pipe. Thus, a client application can read, create, and update files on the remote server. It can also communicate with a server program set up to receive an SMB client request.

Installing Software on Remote Systems

You can install software on remote systems in several ways. We recommend that you use a third-party deployment system for a large preinstalled base. You can also use the operating system-specific installation packages included with the distribution media and copy the required files.

Configuring Software for Distributed Systems

You can configure software for a larger distributed system in several ways, but in all cases, it is desirable to accomplish this remotely. You can generate a cookie-cutter configuration file that can be used by all systems or classes of systems (for example, all Windows 2003 systems or all UltraSPARC systems), and then use scripts to copy specific configuration data to remote systems or include these configuration files with the CA eHealth SystemEDGE files when using third-party deployment tools.

Note: The CA eHealth AdvantEDGE View interface enables configuration of individual agents or groups of agents. For more information, see the *CA eHealth AdvantEDGE View Web Help*. To access the *CA eHealth AdvantEDGE View Web Help*, open the CA eHealth Advantage View console and click the question mark (?) icon on the top right of the screen.

You can also update the CA eHealth SystemEDGE configuration file remotely through SNMP Set commands. Programs like CA eHealth AdvantEDGE View, CA eHealth, HP OpenView, and other SNMP-compliant software can configure the CA eHealth SystemEDGE agent remotely.

Security Issues

System security is a complex problem that involves tradeoffs between usability and system integrity. Making a system more secure often infringes on the usability and the ease of use of the underlying system. Using an SNMP agent such as the CA eHealth SystemEDGE agent requires some policy decisions about what functionality to permit or restrict. This section identifies some of the security issues that you should consider when deploying the CA eHealth SystemEDGE agent.

Those security issues center on four main points:

- Integrity of the underlying system
- SNMP communities
- Scripts and commands invoked by the CA eHealth SystemEDGE agent
- MIB groups

Security of the underlying system is important for restricting access to the CA eHealth SystemEDGE agent configuration files (configuration, monitor, and license) to only those users who are authorized to read and write them. Both read and write access to these files should be restricted:

- Read access can provide information about valid community names and their respective permissions (read-only or read-write).
- Write access to these files permits modification of SNMPv3 USM security information, community names, associated privileges, and self-monitoring table entries, which can run arbitrary commands.

The CA eHealth SystemEDGE agent currently supports SNMPv1, SNMPv2c, and SNMPv3. Communities (for SNMPv1 and SNMPv2c), like a password in some respects, are transmitted in clear-text in SNMP PDUs. Consequently, community names can be vulnerable to packet snooping. Communities have read-only or read-write permissions associated with them, permitting the inspection of or inspection or alteration of MIB variables respectively. The CA eHealth SystemEDGE agent can attach IP-address based access-control lists to communities, but IP-spoofing can circumvent them. CA recommends migrating to CA eHealth SystemEDGE's SNMPv3 USM security model which offers encrypted communication.

Note: For more information, see *Configuring Access Communities* in the chapter "Configuring the CA eHealth SystemEDGE Agent." For more information about SNMPv3 USM security, see the appendix "SNMPv3 in CA eHealth SystemEDGE".

Security Issues with Extension Variables and Action Commands

Because the CA eHealth SystemEDGE agent runs as root or administrator, you must be careful when using extension variables and action commands. More specifically, you should take steps to check that all commands and scripts use absolute pathnames, are fully debugged, and contain no ambiguous code or unnecessary options. You can also specify that the agent run all subprograms (for example, actions, extension objects, and so on) as with users and groups other than root.

Note: For more information about configuring user and group permissions for subprograms, see *Configuring User/Group Permission for Subprograms (UNIX Only)* in the chapter “Configuring the CA eHealth SystemEDGE Agent.”

For more information about using extension variables, see the chapter “Adding Custom MIB Objects.” For more information about actions that you can use with the CA eHealth SystemEDGE agent, see the chapters “Configuring Threshold Monitoring” and “Configuring Log File Monitoring.”

Security Issues with MIBs

MIB groups within the Host Resources and Systems Management MIBs may permit access to information deemed inappropriate by local system policies. Before you deploy the CA eHealth SystemEDGE agent, examine these MIBs to make sure they do not violate local security policies.

Note: For more information about the Systems Management and Host Resources MIBs, see the chapters “Systems Management MIB” and “Host Resources MIB.”

In particular, verify the settings for access to users, groups, and Who Table information, as well as remote-shell execution support. You can turn off support for these capabilities through the CA eHealth SystemEDGE agent configuration file (`sysedge.cf`).

Note: For more information, see *Configuring Support for User and Group Information* and *Configuring Support for Who Table Information* in the chapter “Configuring the CA eHealth SystemEDGE Agent.”

Chapter 18: Command Line Utilities

This chapter describes the usage and syntax of the command line utilities provided in CA eHealth SystemEDGE.

This section contains the following topics:

[SNMP Command Line Utilities](#) (see page 353)

[Additional Command Line Utilities](#) (see page 403)

SNMP Command Line Utilities

CA eHealth SystemEDGE utilities traditionally (as with previous releases) supported SNMPv1 communication only. With CA eHealth SystemEDGE 4.3.0, all of the SNMP utilities can communicate using SNMPv1, SNMPv2c, and SNMPv3. Utilities can work in IPv4 and IPv6 environments. These utilities reside in the bin sub-directory of the CA eHealth SystemEDGE installation directory.

Usage for all SNMP based utilities has been updated. All SNMP utilities accept SNMPv1, SNMPv2c, and SNMPv3 parameters while retaining their old usage (as in releases earlier than SystemEDGE 4.3). The new usage format is strongly encouraged, as the old usage format will not be supported in the future. Any automated scripts or programs using the old argument format should be updated to the new usage format.

The following SNMP command line utilities are described in this section:

- diagsysedge.exe
- edgemon
- edgewatch
- emphistory
- se_enc
- sendtrap
- snmpget
- snmpset
- sysvariable
- walktree
- xtrapmon

The information passed by these utilities should match the information stored in the agent's configuration files. Utilities (except for xtrapmon) provided in the CA eHealth SystemEDGE distribution do not use any configuration files for their operation.

The agent's SNMPv1/SNMPv2c configuration file, `sysedge.cf`, defines the read-only and read-write community strings used in SNMPv1/SNMPv2c communication.

For more information about `sysedge.cf`, see the chapter "Configuring the CA eHealth SystemEDGE Agent".

The agent retrieves the information about all valid SNMPv3 users, their respective SNMPv3 security configurations (the user's security level), and if it is applicable for a user, the user's authorization and privacy security information from the SNMPv3 configuration file, `sysedgeV3.cf`.

For more information about `sysedgeV3.cf`, see the Configuring SNMPv3 in the appendix "SNMPv3 in CA eHealth SystemEDGE".

diagsysedge.exe Utility--Troubleshooting the Agent

Use `diagsysedge.exe` to verify if the agent is running and to obtain information about the agent that you can use for troubleshooting. This utility generates a report in the output file `diagsysedge.txt` in the system temporary directory `/tmp` for UNIX and `%TEMP%` for Windows.

This utility has the following format:

```
diagsysedge.exe
  [-p port]
  [-c community]
  [-v 1 | 2c | 3]
  [-u secName]
  [-s secLevel]
  [-n contextName]
  [-a authPassword] [-A MD5 | SHA]
  [-x privPassword] [-X DES | AES | 3DES]
  [-m FIPS_mode]
  [-r retries]
  [-t timeout]
  [-d logLevel]
  [-f logFile]
  [-V]
  [OPTIONS]
```

-p *port*

Specifies the UDP port that the agent is running on (for example, 1691).

Default: 161

-c *community*

Specifies a community string that the agent uses. Valid for SNMPv1 and SNMPv2c only.

Note: A read-write community string has to be specified for snmpset.

Default: public

-v 1 | 2c | 3

Indicates the version of SNMP that the agent is running. Specify 1 for SNMPv1, 2c for SNMPv2c, or 3 for SNMPv3.

Default: 1

-u *secName*

Specifies the name of the SNMPv3 secure user.

Default: none

-s *secLevel*

Specifies one of the following security levels for SNMPv3 communication:

- 1 - noAuthNoPriv
- 2 - AuthNoPriv
- 3 - AuthPriv (SNMPv3 only)

Default: none

-n *contextName*

Specifies the context name used by the agent if it is configured as SNMPv3.

Note: This option is not required for SNMPv3 communication.

Default: none

-a *authPassword*

Specifies the authentication password if the agent is configured for SNMPv3 with secLevel 2 (AuthNoPriv) or 3 (AuthPriv).

Default: none

-A MD5 | SHA

Specifies the authentication protocol to be used by SNMPv3. This is required if the SNMPv3 user is configured with secLevel 2 (AuthNoPriv) or 3 (AuthPriv). Currently only MD5 (Message Digest Algorithm) and SHA (Secure Hash Algorithm, if the agent is configured for SNMPv3 with secLevel 2 (AuthNoPriv) or 3 (AuthPriv)) are used.

Default: MD5

-x *privPassword*

Specifies the privacy (encryption) password if the agent is configured for SNMPv3 with secLevel 3 (AuthPriv).

Default: none

-X DES | AES | 3DES

Specifies the privacy protocol if the SNMPv3 user is configured with secLevel 3 (AuthPriv). Specify DES for Data Encryption Standard, AES for Advanced Encryption Standard using cryptographic keys of 128 bits (AES128), and 3DES for Triple Data Encryption Standard.

Default: none

-m FIPS_mode

Controls the FIPS mode of operation. Accepted values are 0, 1, and 2.

0

Indicates Non-FIPS mode.

1

Indicates FIPS co-existence mode.

2

Indicates FIPS only mode.

-r *retries*

Specifies the number of retries.

Default: 3

-t *timeout*

Specifies the duration before the SNMP receiver considers the request as timed out.

Default: 10 seconds

-d logLevel

Specifies the log level of the SNMP messages. Accepted values are 0 to 5.

0

Logs fatal messages.

1

Logs critical messages.

2

Logs warning messages.

3

Logs informational messages.

4

Logs all of the messages.

5

Logs all of the messages including debugging messages.

Default: 0

-f logfile

Specifies the name of the log file that contains error and debug information. The default log file name for most of the utilities is sysedge_utility.log

Default: none

-V

Generates detailed and verbose information.

OPTIONS

Specifies one of the following options:

-B

Basic Query. Queries if the SystemEDGE Agent is running.

-H

Displays the usage message.

-L

Views the diagsysedge.txt output file if it is already generated.

-S

Generates a report of the SNMP commands only.

-O

Generates a report of the system commands only.

Notes:

- You can run diagsysedge.exe without any arguments.
- diagsysedge.exe cannot query remote hosts.

Examples

diagsysedge.exe

diagsysedge.exe -p 2009 -u userv3 -A SHA -a osa -X AES -x osp -v 3 -s 3 -t 30

diagsysedge.exe -B

diagsysedge.exe -B -p 2009 -c admin -v 1

diagsysedge.exe -0

diagsysedge.exe -0 -p 2009 -c private

diagsysedge.exe -S

diagsysedge.exe -S -p 2009 -u userv3 -A SHA -a osa -X AES -x osp -v 3 -s 3

diagsysedge.exe -L -p 2009

Old Usage

Important! The following old usage is deprecated. The use of the above argument format is strongly encouraged, as the old argument format will not be supported in the future.

diagsysedge.exe IP:PORT,TIMEOUT COMMUNITY [OPTIONS]

edgemon Utility--Monitor Thresholds

edgemon is a command-line utility that automatically configures the CA eHealth SystemEDGE agent to monitor a MIB variable that you specify. With this utility, you can specify the following:

- MIB variable, either by name or by object-identifier
- Threshold value and comparison operator
- Flags
- Description
- Optional action
- Optional "Superseded By Index"

The edgemon utility issues a SNMP set request to create the appropriate entry in the agent's self-monitoring table.

This utility has the following format:

```
edgemon
[-h hostname | ip_addr]
[-p port]
[-c community]
[-v 1 | 2c | 3]
[-u secName]
[-s secLevel]
[-n contextName]
[-a authPassword] [-A MD5 | SHA]
[-x privPassword] [-X DES | AES | 3DES]
[-m FIPS_mode]
[-r retries]
[-t timeout]
[-d logLevel]
[-f logFile]
[-o] [command]
```

-h *hostname* | *ipaddr*

Specifies the hostname or IP address of the system on which the agent is running. Accepts IPv4 and IPv6 addresses.

Default: localhost

-p *port*

Specifies the UDP port that the agent is running on (for example, 1691).

Default: 161

-c *community*

Specifies a community string that the agent uses. Valid for SNMPv1 and SNMPv2c only.

Note: A read-write community string has to be specified for snmpset.

Default: public

-v 1 | 2c | 3

Indicates the version of SNMP that the agent is running. Specify 1 for SNMPv1, 2c for SNMPv2c, or 3 for SNMPv3.

Default: 1

-u *secName*

Specifies the name of the SNMPv3 secure user.

Default: none

-s *secLevel*

Specifies one of the following security levels for SNMPv3 communication:

- 1 - noAuthNoPriv
- 2 - AuthNoPriv
- 3 - AuthPriv (SNMPv3 only)

Default: none

-n *contextName*

Specifies the context name used by the agent if it is configured as SNMPv3.

Note: This option is not required for SNMPv3 communication.

Default: none

-a *authPassword*

Specifies the authentication password if the agent is configured for SNMPv3 with secLevel 2 (AuthNoPriv) or 3 (AuthPriv).

Default: none

-A MD5 | SHA

Specifies the authentication protocol to be used by SNMPv3. This is required if the SNMPv3 user is configured with secLevel 2 (AuthNoPriv) or 3 (AuthPriv). Currently only MD5 (Message Digest Algorithm) and SHA (Secure Hash Algorithm, if the agent is configured for SNMPv3 with secLevel 2 (AuthNoPriv) or 3 (AuthPriv)) are used.

Default: MD5

-x *privPassword*

Specifies the privacy (encryption) password if the agent is configured for SNMPv3 with secLevel 3 (AuthPriv).

Default: none

-X *DES | AES | 3DES*

Specifies the privacy protocol if the SNMPv3 user is configured with secLevel 3 (AuthPriv). Specify DES for Data Encryption Standard, AES for Advanced Encryption Standard using cryptographic keys of 128 bits (AES128), and 3DES for Triple Data Encryption Standard.

Default: none

-m *FIPS_mode*

Controls the FIPS mode of operation. Accepted values are 0, 1, and 2.

0

Indicates Non-FIPS mode.

1

Indicates FIPS co-existence mode.

2

Indicates FIPS only mode.

-r *retries*

Specifies the number of retries.

Default: 3

-t *timeout*

Specifies the duration before the SNMP receiver considers the request as timed out.

Default: 10 seconds

-d logLevel

Specifies the log level of the SNMP messages. Accepted values are 0 to 5.

0

Logs fatal messages.

1

Logs critical messages.

2

Logs warning messages.

3

Logs informational messages.

4

Logs all of the messages.

5

Logs all of the messages including debugging messages.

Default: 0

-f logfile

Specifies the name of the log file that contains error and debug information. The default log file name for most of the utilities is `sysedge_utility.log`

Default: none

-o command

Specifies the command and associated arguments. Supported commands include the following:

- `oid` (for monitoring an object)
- `filesystem` (for monitoring a file system)
- `list` (for listing the current Monitor table entries)
- `setstatus` (for setting the status of a Monitor table entry)
- `delete` (for deleting a Monitor table entry)

Note: For more information about the above commands, see the `edgemon` Commands for Threshold Monitoring in the chapter "Configuring Threshold Monitoring".

Note: For detailed `edgemon` examples see the `edgemon` Examples in the chapter "Configuring Threshold Monitoring".

Old Usage

Important! The following old usage is deprecated. The use of the above argument format is strongly encouraged, as the old argument format will not be supported in the future.

```
edgemon ipaddr[:port][,timeout] commstr [command]
```

edgemon Utility--Monitor Processes

edgemon is a command-line utility that automatically configures the CA eHealth SystemEDGE agent to monitor processes, log files, and Windows event logs. After you specify the particular process, log file, or Windows event log and the associated arguments, the edgemon utility issues a SNMP set request to create the appropriate entry in the agent's process monitoring, nt event monitoring, log file monitoring, and process group monitoring tables.

This utility has the following format:

```
edgemon
[-h hostname | ip_addr]
[-p port]
[-c community]
[-v 1 | 2c | 3]
[-u secName]
[-s secLevel]
[-n contextName]
[-a authPassword] [-A MD5 | SHA]
[-x privPassword] [-X DES | AES | 3DES]
[-m FIPS_mode]
[-r retries]
[-t timeout]
[-d logLevel]
[-f logFile]
[-o facility command]
```

-h hostname | ipaddr

Specifies the hostname or IP address of the system on which the agent is running. Accepts IPv4 and IPv6 addresses.

Default: localhost

-p port

Specifies the UDP port that the agent is running on (for example, 1691).

Default: 161

-c community

Specifies a community string that the agent uses. Valid for SNMPv1 and SNMPv2c only.

Note: A read-write community string has to be specified for snmpset.

Default: public

-v 1 | 2c | 3

Indicates the version of SNMP that the agent is running. Specify 1 for SNMPv1, 2c for SNMPv2c, or 3 for SNMPv3.

Default: 1

-u secName

Specifies the name of the SNMPv3 secure user.

Default: none

-s secLevel

Specifies one of the following security levels for SNMPv3 communication:

- 1 - noAuthNoPriv
- 2 - AuthNoPriv
- 3 - AuthPriv (SNMPv3 only)

Default: none

-n contextName

Specifies the context name used by the agent if it is configured as SNMPv3.

Note: This option is not required for SNMPv3 communication.

Default: none

-a authPassword

Specifies the authentication password if the agent is configured for SNMPv3 with secLevel 2 (AuthNoPriv) or 3 (AuthPriv).

Default: none

-A MD5 | SHA

Specifies the authentication protocol to be used by SNMPv3. This is required if the SNMPv3 user is configured with secLevel 2 (AuthNoPriv) or 3 (AuthPriv). Currently only MD5 (Message Digest Algorithm) and SHA (Secure Hash Algorithm, if the agent is configured for SNMPv3 with secLevel 2 (AuthNoPriv) or 3 (AuthPriv)) are used.

Default: MD5

-x *privPassword*

Specifies the privacy (encryption) password if the agent is configured for SNMPv3 with secLevel 3 (AuthPriv).

Default: none

-X *DES | AES | 3DES*

Specifies the privacy protocol if the SNMPv3 user is configured with secLevel 3 (AuthPriv). Specify DES for Data Encryption Standard, AES for Advanced Encryption Standard using cryptographic keys of 128 bits (AES128), and 3DES for Triple Data Encryption Standard.

Default: none

-m *FIPS_mode*

Controls the FIPS mode of operation. Accepted values are 0, 1, and 2.

0

Indicates Non-FIPS mode.

1

Indicates FIPS co-existence mode.

2

Indicates FIPS only mode.

-r *retries*

Specifies the number of retries.

Default: 3

-t *timeout*

Specifies the duration before the SNMP receiver considers the request as timed out.

Default: 10 seconds

-d *logLevel*

Specifies the log level of the SNMP messages. Accepted values are 0 to 5.

0

Logs fatal messages.

1

Logs critical messages.

2

Logs warning messages.

3

Logs informational messages.

4

Logs all of the messages.

5

Logs all of the messages including debugging messages.

Default: 0

-f *logfile*

Specifies the name of the log file that contains error and debug information. The default log file name for most of the utilities is `sysedge_utility.log`

Default: none

facility

Specifies a facility to use. Supported values are the following:

- process
- logfile
- ntevent
- procgroup

command

Specifies the command and associated arguments. Supported commands are the following:

- add
- setstatus
- delete
- list
- dump

Note: For more information about the above commands, see edgework Commands for Process Monitoring in the chapter "Configuring Process and Service Monitoring".

Note: For detailed edgework examples see edgework Examples in the chapter "Configuring Process and Service Monitoring".

Old Usage

Important! The following old usage is deprecated. The use of the above argument format is strongly encouraged, as the old argument format will not be supported in the future.

```
edgework hostname [:port][,timeout] community facility command
```

emphistory Utility--Manage Entries in History Control Table

emphistory is a command-line utility that can configure entries in the History Control table and retrieve data samples from the History Control table dynamically.

This utility has the following format:

```
emphistory
  [-h hostname | ip_addr]
  [-p port]
  [-c community]
  [-v 1 | 2c | 3]
  [-u secName]
  [-s secLevel]
  [-n contextName]
  [-a authPassword] [-A MD5 | SHA]
  [-x privPassword] [-X DES | AES | 3DES]
  [-m FIPS_mode]
  [-r retries]
  [-t timeout]
  [-d logLevel]
  [-f logFile]
  [-o] [command]
```

-h *hostname* | *ipaddr*

Specifies the hostname or IP address of the system on which the agent is running. Accepts IPv4 and IPv6 addresses.

Default: localhost

-p *port*

Specifies the UDP port that the agent is running on (for example, 1691).

Default: 161

-c *community*

Specifies a community string that the agent uses. Valid for SNMPv1 and SNMPv2c only.

Note: A read-write community string has to be specified for snmpset.

Default: public

-v 1 | 2c | 3

Indicates the version of SNMP that the agent is running. Specify 1 for SNMPv1, 2c for SNMPv2c, or 3 for SNMPv3.

Default: 1

-u *secName*

Specifies the name of the SNMPv3 secure user.

Default: none

-s *secLevel*

Specifies one of the following security levels for SNMPv3 communication:

- 1 - noAuthNoPriv
- 2 - AuthNoPriv
- 3 - AuthPriv (SNMPv3 only)

Default: none

-n *contextName*

Specifies the context name used by the agent if it is configured as SNMPv3.

Note: This option is not required for SNMPv3 communication.

Default: none

-a *authPassword*

Specifies the authentication password if the agent is configured for SNMPv3 with secLevel 2 (AuthNoPriv) or 3 (AuthPriv).

Default: none

-A MD5 | SHA

Specifies the authentication protocol to be used by SNMPv3. This is required if the SNMPv3 user is configured with secLevel 2 (AuthNoPriv) or 3 (AuthPriv). Currently only MD5 (Message Digest Algorithm) and SHA (Secure Hash Algorithm, if the agent is configured for SNMPv3 with secLevel 2 (AuthNoPriv) or 3 (AuthPriv)) are used.

Default: MD5

-x *privPassword*

Specifies the privacy (encryption) password if the agent is configured for SNMPv3 with secLevel 3 (AuthPriv).

Default: none

-X DES | AES | 3DES

Specifies the privacy protocol if the SNMPv3 user is configured with secLevel 3 (AuthPriv). Specify DES for Data Encryption Standard, AES for Advanced Encryption Standard using cryptographic keys of 128 bits (AES128), and 3DES for Triple Data Encryption Standard.

Default: none

-m *FIPS_mode*

Controls the FIPS mode of operation. Accepted values are 0, 1, and 2.

0

Indicates Non-FIPS mode.

1

Indicates FIPS co-existence mode.

2

Indicates FIPS only mode.

-r *retries*

Specifies the number of retries.

Default: 3

-t *timeout*

Specifies the duration before the SNMP receiver considers the request as timed out.

Default: 10 seconds

-d logLevel

Specifies the log level of the SNMP messages. Accepted values are 0 to 5.

0

Logs fatal messages.

1

Logs critical messages.

2

Logs warning messages.

3

Logs informational messages.

4

Logs all of the messages.

5

Logs all of the messages including debugging messages.

Default: 0

-f logfile

Specifies the name of the log file that contains error and debug information. The default log file name for most of the utilities is `sysedge_utility.log`

Default: none

command

Specifies the command and associated arguments. Supported commands are the following:

- add
- setstatus
- delete
- dump
- list

Note: For more information about the above commands, see `emphistory` Commands for Managing Entries in History Control Table in the chapter "Configuring History Collection".

Note: For detailed `emphistory` examples see `emphistory` Examples in the chapter "Configuring History Collection".

Old Usage

Important! The following old usage is deprecated. The use of the above argument format is strongly encouraged, as the old argument format will not be supported in the future.

```
emphistory add ipaddr[:port] commstr ctrlIndex objid interval buckets  
['description']
```

```
emphistory delete ipaddr[:port] commstr ctrlIndex
```

```
emphistory list ipaddr[:port] commstr
```

```
emphistory setstatus ipaddr[:port] commstr ctrlIndex status
```

```
emphistory dump ipaddr[:port] commstr ctrlIndex
```

se_enc Utility--Encrypt the SNMPV3 Configuration File

You encrypt the SNMPv3 configuration file using the se_enc utility.

For information about how to use this utility to encrypt the SNMPv3 configuration file, see Encrypt the SNMPv3 Configuration File in the appendix "SNMPv3 in CA eHealth SystemEDGE".

sendtrap Utility--Send a SNMP UDP Trap

sendtrap sends a SNMP trap PDU from the node you are on to any node on your network. By default, SNMP trap PDUs are sent to the SNMP Trap port (UDP/162) on the specified host. The sendtrap utility can send SNMPv1, SNMPv2c, and SNMPv3 UDP traps.

This utility has the following format:

```
sendtrap
  [-f from_addr | from_host]
  [-h dest_addr | dest_host]
  [-i] [-r retries]
  [-p port]
  [-c community]
  [-v 1 | 2c | 3]
  [-u secName]
  [-s secLevel]
  [-n contextName]
  [-a authPassword] [-A MD5 | SHA]
  [-x privPassword] [-X DES | AES | 3DES]
  [-m FIPS_mode]
  [-t timeout]
  [-d logLevel]
  [enterprise-oid] [trap-type] [subtype] [data-oid] [oid-type] [oid-value]
```

-f from_addr | from_host

Changes the source address in the SNMP Trap PDU. The default value is an IP address of the host that is executing sendtrap.

-h dest_addr | dest_host

Specifies the destination host name or IP address to which the trap is being sent.

-i

Sends inform requests (INFORM REQUEST) and waits for acknowledgement. These are also known as confirmed traps. Specify -i only if the -v (trap version) argument is 2c (SNMPv2c) or 3 (SNMPv3).

-r retries

Specifies the number of retries to deliver an inform request until it is acknowledged.

-p port

Specifies the UDP port that the agent is running on (for example, 1691).

Default: 161

-c community

Specifies a community string that the agent uses. Valid for SNMPv1 and SNMPv2c only.

Note: A read-write community string has to be specified for snmpset.

Default: public

-v 1 | 2c | 3

Indicates the version of SNMP that the agent is running. Specify 1 for SNMPv1, 2c for SNMPv2c, or 3 for SNMPv3.

Default: 1

-u secName

Specifies the name of the SNMPv3 secure user.

Default: none

-s secLevel

Specifies one of the following security levels for SNMPv3 communication:

- 1 - noAuthNoPriv
- 2 - AuthNoPriv
- 3 - AuthPriv (SNMPv3 only)

Default: none

-n contextName

Specifies the context name used by the agent if it is configured as SNMPv3.

Note: This option is not required for SNMPv3 communication.

Default: none

-a authPassword

Specifies the authentication password if the agent is configured for SNMPv3 with secLevel 2 (AuthNoPriv) or 3 (AuthPriv).

Default: none

-A MD5 | SHA

Specifies the authentication protocol to be used by SNMPv3. This is required if the SNMPv3 user is configured with secLevel 2 (AuthNoPriv) or 3 (AuthPriv). Currently only MD5 (Message Digest Algorithm) and SHA (Secure Hash Algorithm, if the agent is configured for SNMPv3 with secLevel 2 (AuthNoPriv) or 3 (AuthPriv)) are used.

Default: MD5

-x *privPassword*

Specifies the privacy (encryption) password if the agent is configured for SNMPv3 with secLevel 3 (AuthPriv).

Default: none

-X DES | AES | 3DES

Specifies the privacy protocol if the SNMPv3 user is configured with secLevel 3 (AuthPriv). Specify DES for Data Encryption Standard, AES for Advanced Encryption Standard using cryptographic keys of 128 bits (AES128), and 3DES for Triple Data Encryption Standard.

Default: none

Note: Install the CA eHealth Advanced Encryption package if you specify AES or 3DES. For more information, see the appendix "CA eHealth Advanced Encryption".

-m FIPS_mode

Controls the FIPS mode of operation. Accepted values are 0, 1, and 2.

0

Indicates Non-FIPS mode.

1

Indicates FIPS co-existence mode.

2

Indicates FIPS only mode.

Note: FIPS mode requires you to install the CA eHealth Advanced Encryption package. For more information, see the appendix "CA eHealth Advanced Encryption".

-t *timeout*

Specifies the duration before the SNMP receiver considers the request as timed out.

Default: 10 seconds

-d logLevel

Specifies the log level of the SNMP messages. Accepted values are 0 to 5.

0

Logs fatal messages.

1

Logs critical messages.

2

Logs warning messages.

3

Logs informational messages.

4

Logs all of the messages.

5

Logs all of the messages including debugging messages.

Default: 0

enterprise-oid

Specifies the top level enterprise object identifier that represents this trap.

trap-type

Specifies the generic trap type in the SNMP Trap PDU. Defined in RFC 1157, this field can accept one of the following values (integers 0 - 6):

- 0 - cold start
- 1 - warm start
- 2 - link down
- 3 - link up
- 4 - authentication failure
- 5 - EGP Neighbor Loss
- 6 - enterprise specific

Values less than 0 (zero) cause sendtrap to print an error message and exit. Values greater than 6 cause sendtrap to issue a warning message.

subtype

Specifies an enterprise-specific trap subtype. An accepted value for this field is an integer. You should only specify this if the trap type is 6 (Enterprise specific trap).

data-oid

Specifies the Object Identifier (OID) that is included in the SNMP Trap PDU.

oid-type

Specifies the type of the OID value to be set. OID type can be one of the following:

-i

integer

-o

octet string. Valid on character strings, binary and string IPv4 and IPv6 addresses, and string host names.

-s

string

-d

object identifier

-a

IPv4 address only

-c

counter value

-C

64 bit counter value

-g

gauge

-t

time ticks

oid-value

Specifies the value of the OID to be set. The type of the OID value should match OID-type.

Notes:

- The default port number for sendtrap is 162.
- The enterprise OID, trap type and data OID pair should be the last in the argument list.
- You can specify multiple data OID pairs (commonly referred as varbinds) separated by a blank space. All of the varbinds are then associated with the same enterprise oid, trap type, and trap sub-type.

- There is no limit to the number of varbinds in a single trap message.
- If sendtrap is sending a SNMPv3 trap, the information passed by sendtrap should match the information stored in the agent's SNMPv3 configuration file of the receiver. You do not need any configuration files to run sendtrap.

sendtrap Examples

```
./sendtrap -h box1.domain.com -f from.domain.com -v 2c -c admin -p 1692 1.2.3.4 6  
1023 1.3.6.1.2.1.2.2.1.1.1 -i 3 1.3.7.8.9.10.11 -s "Trap value"
```

box1.domain.com

Sends the trap to this host name.

from.domain.com

Sends the trap with from.domain.com in the from address in the trap PDU.

1692

Specifies the port number to send the trap.

1.2.3.4

Specifies the enterprise OID.

6

Specifies the trap type (enterprise specific trap).

1023

Specifies the enterprise specific trap sub-type.

1.3.6.1.2.1.2.2.1.1.1 and 1.3.7.8.9.10.11

Specify the data OIDs.

-i and -s

Specify the oid types.

3 and "Trap value"

Specify the OID values of the respective OIDs.

The following example sends an authentication failure (trap_type: 4) SNMPv1 trap with varbinds 1.3.6.1.2.1.2.2.1.1.1 and 1.3.7.8.9.10.11 to port 162 on the host with IP address Ea2f:fe90:abcd:0000:230:a2f:200:ad01:

```
./sendtrap -h Ea2f:fe90:abcd:0000:230:a2f:200:ad01 -v 1 -c admin 1.2.3.4 4  
1.3.6.1.2.1.2.2.1.1.1 -i 3 1.3.7.8.9.10.11 -s "Trap value"
```

The following example sends an authentication failure (trap_type: 4) SNMPv3 Inform request with varbinds 1.3.6.1.2.1.2.2.1.1.1 and 1.3.7.8.9.10.11 to port 162 on the host with IP address 130.10.100.101. It waits for acknowledgement with a timeout of 30 seconds and retries 2 times.

```
./sendtrap -h 130.10.100.101 -p 2009 -i -r 2 -t 30 -u userlv3 -A SHA -a osa -X  
AES -x osp -v 3 -s 3 -t 30 -h 130.10.100.101 1.2.3.4 4 1.3.6.1.2.1.2.2.1.1.1 -i 3  
1.3.7.8.9.10.11 -s "Trap value"
```

Old Usage

Important! The following old usage is deprecated. The use of the above argument format is strongly encouraged, as the old argument format will not be supported in the future.

```
sendtrap host TrapType SpecificType {EnterpriseOid} [varbinds]
```

Old Usage Examples

This section includes sample filters that you can add to a file. Add these filters to your CA eHealth SystemEDGE configuration file to perform the actions that they describe.

Example: Send an Enterprise-Specific Trap 4 PDU

This example sends an enterprise-specific Trap 4 PDU (without variable bindings) to the local host:

```
sendtrap 127.0.0.1 6 4 < /dev/null
```

Example: Send a MIB-II linkup(3) Trap

This example sends a MIB-II linkUp(3) Trap PDU to the local host with a single variable binding that contains the integer 1 for a Windows system:

```
sendtrap 127.0.0.1 3 0  
1.3.6.1.2.1.2.2.1.1.1 integer 1  
^Z
```

Note: For a UNIX system, use the ^d end-of-file character instead of ^Z.

Example: Redirect Variable Bindings from stdin into sendtrap

This example redirects variable bindings from stdin into sendtrap:

```
sendtrap 127.0.0.1 6 321 <<!  
1.3.6.1.2.1.4.20.1.1.5.5.5.5 ipaddr 5.5.5.5  
1.3.6.1.2.1.4.20.1.1.6.6.6.6 ipaddr 6.6.6.6  
1.3.6.1.2.1.4.20.1.1.127.0.0.1 ipaddr 127.0.0.1!
```

Note: Invoke this command in the UNIX shell /bin/sh. Input/output redirection is specific to each shell. For information about redirecting variable bindings with other shells, consult the man pages for those shells.

If you want to invoke `sendtrap` from within another C program, see the `call-sendtrap.c` (UNIX), or `callsend.c` (Windows) sample file included in the `scripts` subdirectory. These scripts show how to correctly invoke `sendtrap` and pass the requested variable bindings.

varbinds--Specify Variable Bindings for sendtrap

You can specify optional variable bindings as standard input to `sendtrap`. Variable bindings are data fields in the SNMP Trap PDU. Each variable binding associates a particular object instance with its current value and contains an object-identifier, an object type, and a value. Variable bindings are passed as input to `sendtrap` as ASCII character strings. The `sendtrap` utility converts them to SNMPv1 format.

You must enter each variable binding on a separate input line. The variable-bindings list is terminated by an end-of-file (EOF) character (^Z for Windows systems, or ^d for UNIX systems). If you do not want to provide variable bindings to `sendtrap`, redirect input from `/dev/null` or a zero-length file.

The OIDs are specified in dotted-notation format (for example, 1.3.6), and types are indicated from a set of constant, case-insensitive strings. The type may be one of the following:

- `ipaddr`
- `cntr`
- `gauge`
- `timeticks`
- `integer`
- `string`
- `objid`

Values are dependent on the type and are converted appropriately to internal format. If `sendtrap` encounters conversion errors, it skips the current variable binding, rather than abandoning trap generation.

You can script this utility to redirect variable bindings from standard input (`stdin`) to `sendtrap`.

Notes:

- If you are not using input from a file, you must provide the end-of-file character for each `sendtrap` command. Use ^Z for Windows systems or ^d for UNIX systems.
- The maximum number of `varbinds` that you can specify in a single trap is 100.

SpecificType

Specifies the integer to use in the enterprise-specific trap type field in the Trap PDU. SpecificType values less than 0 cause sendtrap to print an error message and exit.

Note: The sendtrap utility reports 0 for the Trap PDU's time-stamp field because it cannot know the real value. Due to internal limits, sendtrap can send a maximum of 32 variable bindings in a single Trap PDU. You must be able to represent object values as an ASCII character string to enable sendtrap to read, convert, and send them within Trap PDUs. sendtrap does not recognize or convert ASCII strings for the TrapType or SpecificType arguments. You can specify only integers for these fields.

snmpget Utility--Retrieve an OID value

snmpget retrieves the value of a specific instance of a MIB attribute using SNMP. The OID of the attribute is printed followed by the value of the OID.

This utility has the following format:

```
snmpget
  [-h hostname | ip_addr]
  [-p port]
  [-c community]
  [-v 1 | 2c | 3]
  [-u secName]
  [-s secLevel]
  [-n contextName]
  [-a authPassword] [-A MD5 | SHA]
  [-x privPassword] [-X DES | AES | 3DES]
  [-m FIPS_mode]
  [-r retries]
  [-t timeout]
  [-d logLevel]
  [-f logFile]
  [-b]
  [-o OID]
```

-h *hostname* | *ipaddr*

Specifies the hostname or IP address of the system on which the agent is running. Accepts IPv4 and IPv6 addresses.

Default: localhost

-p *port*

Specifies the UDP port that the agent is running on (for example, 1691).

Default: 161

-c *community*

Specifies a community string that the agent uses. Valid for SNMPv1 and SNMPv2c only.

Note: A read-write community string has to be specified for snmpset.

Default: public

-v 1 | 2c | 3

Indicates the version of SNMP that the agent is running. Specify 1 for SNMPv1, 2c for SNMPv2c, or 3 for SNMPv3.

Default: 1

-u *secName*

Specifies the name of the SNMPv3 secure user.

Default: none

-s *secLevel*

Specifies one of the following security levels for SNMPv3 communication:

- 1 - noAuthNoPriv
- 2 - AuthNoPriv
- 3 - AuthPriv (SNMPv3 only)

Default: none

-n *contextName*

Specifies the context name used by the agent if it is configured as SNMPv3.

Note: This option is not required for SNMPv3 communication.

Default: none

-a *authPassword*

Specifies the authentication password if the agent is configured for SNMPv3 with secLevel 2 (AuthNoPriv) or 3 (AuthPriv).

Default: none

-A MD5 | SHA

Specifies the authentication protocol to be used by SNMPv3. This is required if the SNMPv3 user is configured with secLevel 2 (AuthNoPriv) or 3 (AuthPriv). Currently only MD5 (Message Digest Algorithm) and SHA (Secure Hash Algorithm, if the agent is configured for SNMPv3 with secLevel 2 (AuthNoPriv) or 3 (AuthPriv)) are used.

Default: MD5

-x *privPassword*

Specifies the privacy (encryption) password if the agent is configured for SNMPv3 with secLevel 3 (AuthPriv).

Default: none

-X *DES | AES | 3DES*

Specifies the privacy protocol if the SNMPv3 user is configured with secLevel 3 (AuthPriv). Specify DES for Data Encryption Standard, AES for Advanced Encryption Standard using cryptographic keys of 128 bits (AES128), and 3DES for Triple Data Encryption Standard.

Default: none

-m *FIPS_mode*

Controls the FIPS mode of operation. Accepted values are 0, 1, and 2.

0

Indicates Non-FIPS mode.

1

Indicates FIPS co-existence mode.

2

Indicates FIPS only mode.

-r *retries*

Specifies the number of retries.

Default: 3

-t *timeout*

Specifies the duration before the SNMP receiver considers the request as timed out.

Default: 10 seconds

-d logLevel

Specifies the log level of the SNMP messages. Accepted values are 0 to 5.

0

Logs fatal messages.

1

Logs critical messages.

2

Logs warning messages.

3

Logs informational messages.

4

Logs all of the messages.

5

Logs all of the messages including debugging messages.

Default: 0

-f logfile

Specifies the name of the log file that contains error and debug information. The default log file name for most of the utilities is `sysedge_utility.log`

Default: none

-b

Displays the value in hexadecimal format. Applies to `snmpget`, `snmpset`, and `walktree` utilities only.

-o OID

Specifies the object identifier (OID) to be set or queried for the `snmpget`, `snmpset`, and `walktree` utilities.

Default: none

Notes:

- The default port number for `snmpget` is 161.
- OID must be the last argument for `snmpget`.
- You can query multiple OIDs (separated by a blank space) in a single `snmpget` call. Examples are provided below for reference.

snmpget Examples

```
snmpget -o 1.3.6.1.2.1.1.1.0
```

```
snmpget -h box1.domain.com -o 1.3.6.1.2.1.2.1.0 1.3.6.1.2.1.1.4.0
1.3.6.1.2.1.2.2.1.2.1 1.3.6.1.2.1.1.1.0

snmpget -p 2009 -c admin -v 1 -o 1.3.6.1.2.1.1.4.0

snmpget -c admin -v 2c -h Ea2f:fe90:abcd:0000:230:a2f:200:ad01 -b -o
1.3.6.1.2.1.1.1.0 1.3.6.1.2.1.2.2.1.2.1

snmpget -p 2009 -u user3v3 -v 3 -s 1 -o 1.3.6.1.2.1.2.1.0 1.3.6.1.2.1.2.2.1.1.1
1.3.6.1.2.1.2.2.1.2.1

snmpget -h 130.10.100.101 -p 2009 -u user2v3 -A SHA -a osa -v 3 -s 2 -m 2 -o
1.3.6.1.2.1.2.1.0 1.3.6.1.2.1.1.4.0

snmpget -p 2009 -u user1v3 -A SHA -a osa -X AES -x osp -v 3 -s 3 -t 30 -o
1.3.6.1.2.1.2.1.0 1.3.6.1.2.1.2.2.1.1.1 1.3.6.1.2.1.2.2.1.2.1 1.3.6.1.2.1.1.4.0
```

Old Usage

Important! The following old usage is deprecated. The use of the above argument format is strongly encouraged, as the old argument format will not be supported in the future.

```
snmpget community ipaddr[:port][,timeout] instance-id [-s]
```

snmpset Utility--Set Value of an OID

snmpset sets the value a specific instance of a MIB attribute using SNMP.

This utility has the following format:

```
snmpset
  [-h hostname | ip_addr]
  [-p port]
  [-c community]
  [-v 1 | 2c | 3]
  [-u secName]
  [-s secLevel]
  [-n contextName]
  [-a authPassword] [-A MD5 | SHA]
  [-x privPassword] [-X DES | AES | 3DES]
  [-m FIPS_mode]
  [-r retries]
  [-t timeout]
  [-d logLevel]
  [-f logFile]
  [-b ]
  [-o] [OID] [OID-type] [OID-value]
```


-h *hostname* | *ipaddr*

Specifies the hostname or IP address of the system on which the agent is running. Accepts IPv4 and IPv6 addresses.

Default: localhost

-p *port*

Specifies the UDP port that the agent is running on (for example, 1691).

Default: 161

-c *community*

Specifies a community string that the agent uses. Valid for SNMPv1 and SNMPv2c only.

Note: A read-write community string has to be specified for snmpset.

Default: public

-v 1 | 2c | 3

Indicates the version of SNMP that the agent is running. Specify 1 for SNMPv1, 2c for SNMPv2c, or 3 for SNMPv3.

Default: 1

-u *secName*

Specifies the name of the SNMPv3 secure user.

Default: none

-s *secLevel*

Specifies one of the following security levels for SNMPv3 communication:

- 1 - noAuthNoPriv
- 2 - AuthNoPriv
- 3 - AuthPriv (SNMPv3 only)

Default: none

-n *contextName*

Specifies the context name used by the agent if it is configured as SNMPv3.

Note: This option is not required for SNMPv3 communication.

Default: none

-a *authPassword*

Specifies the authentication password if the agent is configured for SNMPv3 with secLevel 2 (AuthNoPriv) or 3 (AuthPriv).

Default: none

-A MD5 | SHA

Specifies the authentication protocol to be used by SNMPv3. This is required if the SNMPv3 user is configured with secLevel 2 (AuthNoPriv) or 3 (AuthPriv). Currently only MD5 (Message Digest Algorithm) and SHA (Secure Hash Algorithm, if the agent is configured for SNMPv3 with secLevel 2 (AuthNoPriv) or 3 (AuthPriv)) are used.

Default: MD5

-x *privPassword*

Specifies the privacy (encryption) password if the agent is configured for SNMPv3 with secLevel 3 (AuthPriv).

Default: none

-X DES | AES | 3DES

Specifies the privacy protocol if the SNMPv3 user is configured with secLevel 3 (AuthPriv). Specify DES for Data Encryption Standard, AES for Advanced Encryption Standard using cryptographic keys of 128 bits (AES128), and 3DES for Triple Data Encryption Standard.

Default: none

-m FIPS_mode

Controls the FIPS mode of operation. Accepted values are 0, 1, and 2.

0

Indicates Non-FIPS mode.

1

Indicates FIPS co-existence mode.

2

Indicates FIPS only mode.

-r *retries*

Specifies the number of retries.

Default: 3

-t *timeout*

Specifies the duration before the SNMP receiver considers the request as timed out.

Default: 10 seconds

-d *logLevel*

Specifies the log level of the SNMP messages. Accepted values are 0 to 5.

0

Logs fatal messages.

1

Logs critical messages.

2

Logs warning messages.

3

Logs informational messages.

4

Logs all of the messages.

5

Logs all of the messages including debugging messages.

Default: 0

-f *logfile*

Specifies the name of the log file that contains error and debug information. The default log file name for most of the utilities is `sysedge_utility.log`

Default: none

-b

Displays the value in hexadecimal format. Applies to `snmpget`, `snmpset`, and `walktree` utilities only.

-o *OID*

Specifies the object identifier (OID) to be set or queried for the `snmpget`, `snmpset`, and `walktree` utilities.

Default: none

OID-type

Specifies the type of the OID value that you are setting. The following are possible values:

-i

integer

-o

octet

-s

string

-d

object identifier

-a

IP address

-c

counter value

-C

64 bit counter value

-g

gauge

-t

time ticks

-x

hexadecimal-encoded binary data (opaque)

OID-value

Specifies the value of the OID you want to set. The type of the OID value should match OID-type.

Notes:

- The default port number for snmpset is 161.
- To set an OID value, a read-write community string for SNMPv1/SNMPv2c is required and a read-write SNMPv3 user is required for SNMPv3.
- The [OID] [OID-type] [OID-value] pair (commonly referred as varbinds) should be the last arguments for snmpset, and you should specify them in this order.
- You can give multiple OID pairs a single snmpset call. Examples are provided below for reference.

Examples

```
snmpset -c admin -o 1.3.6.1.2.1.1.4.0 -s "syscontact update"
```

```
snmpset -h Ea2f:fe90:abcd:0000:230:a2f:200:ad01 -c admin -o 1.3.6.1.2.1.1.4.0 -s  
"syscontact update" 1.3.6.1.2.1.1.6.0 -s "syslocation update"
```

```
snmpset -h box1.domain.com -p 2009 -c admin -v 1 -o 1.3.6.1.2.1.1.4.0 -s  
"syscontact update" 1.3.6.1.2.1.1.6.0 -s "syslocation update"
```

```
snmpset -c admin -v 2c -h Ea2f:fe90:abcd:0000:230:a2f:200:ad01 -o  
1.3.6.1.2.1.1.6.0 -s "syslocation update"
```

```
snmpset -p 2009 -u user3v3 -v 3 -s 1 -o 1.3.6.1.2.1.1.4.0 -s "syscontact update"  
1.3.6.1.2.1.1.6.0 -s "syslocation update"
```

```
snmpset -p 2009 -u user2v3 -A SHA -a osa -v 3 -s 2 -m 2 -o 1.3.6.1.2.1.1.6.0 -s  
"syslocation update"
```

```
snmpset -p 2009 -h 130.10.100.101 -u user1v3 -A SHA -a osa -X AES -x osp -v 3 -s  
3 -t 30 -o 1.3.6.1.2.1.1.4.0 -s "syscontact update" 1.3.6.1.2.1.1.6.0 -s  
"syslocation update"
```

Old Usage

Important! The following old usage is deprecated. The use of the above argument format is strongly encouraged, as the old argument format will not be supported in the future.

```
snmpset comm-str ipaddr[:port] {varbinds}  
    varbind = { OID Hex-Type Value }  
    type = 40 (ipaddr) | 41 (counter) | 42 (gauge)  
           43 (TimeTick) | 02 (int) | 04 (octetstring)  
           06 (Objid)
```

sysvariable Utility--Retrieve a System Value

sysvariable retrieves the value of a specific system value using SNMP. This utility is a snmpget with the specific OID. sysvariable is a simple way to retrieve system values without having to know the numeric OID name.

This utility has the following format:

```
sysvariable
[-h hostname | ip_addr]
[-p port]
[-c community]
[-v 1 | 2c | 3]
[-u secName]
[-s secLevel]
[-n contextName]
[-a authPassword] [-A MD5 | SHA]
[-x privPassword] [-X DES | AES | 3DES]
[-m FIPS_mode]
[-r retries]
[-t timeout]
[-d logLevel]
[-f logFile]
[-V]
[-o Variable]
```

-h *hostname* | *ipaddr*

Specifies the hostname or IP address of the system on which the agent is running. Accepts IPv4 and IPv6 addresses.

Default: localhost

-p *port*

Specifies the UDP port that the agent is running on (for example, 1691).

Default: 161

-c *community*

Specifies a community string that the agent uses. Valid for SNMPv1 and SNMPv2c only.

Note: A read-write community string has to be specified for snmpset.

Default: public

-v 1 | 2c | 3

Indicates the version of SNMP that the agent is running. Specify 1 for SNMPv1, 2c for SNMPv2c, or 3 for SNMPv3.

Default: 1

-u *secName*

Specifies the name of the SNMPv3 secure user.

Default: none

-n *contextName*

Specifies the context name used by the agent if it is configured as SNMPv3.

Note: This option is not required for SNMPv3 communication.

Default: none

-a *authPassword*

Specifies the authentication password if the agent is configured for SNMPv3 with secLevel 2 (AuthNoPriv) or 3 (AuthPriv).

Default: none

-A MD5 | SHA

Specifies the authentication protocol to be used by SNMPv3. This is required if the SNMPv3 user is configured with secLevel 2 (AuthNoPriv) or 3 (AuthPriv). Currently only MD5 (Message Digest Algorithm) and SHA (Secure Hash Algorithm, if the agent is configured for SNMPv3 with secLevel 2 (AuthNoPriv) or 3 (AuthPriv)) are used.

Default: MD5

-x *privPassword*

Specifies the privacy (encryption) password if the agent is configured for SNMPv3 with secLevel 3 (AuthPriv).

Default: none

-X DES | AES | 3DES

Specifies the privacy protocol if the SNMPv3 user is configured with secLevel 3 (AuthPriv). Specify DES for Data Encryption Standard, AES for Advanced Encryption Standard using cryptographic keys of 128 bits (AES128), and 3DES for Triple Data Encryption Standard.

Default: none

-m FIPS_mode

Controls the FIPS mode of operation. Accepted values are 0, 1, and 2.

0

Indicates Non-FIPS mode.

1

Indicates FIPS co-existence mode.

2

Indicates FIPS only mode.

-r retries

Specifies the number of retries.

Default: 3

-t timeout

Specifies the duration before the SNMP receiver considers the request as timed out.

Default: 10 seconds

-d logLevel

Specifies the log level of the SNMP messages. Accepted values are 0 to 5.

0

Logs fatal messages.

1

Logs critical messages.

2

Logs warning messages.

3

Logs informational messages.

4

Logs all of the messages.

5

Logs all of the messages including debugging messages.

Default: 0

-f logfile

Specifies the name of the log file that contains error and debug information. The default log file name for most of the utilities is sysedge_utility.log

Default: none

-V

Generates detailed and verbose information.

Variable

Specifies the system value you want to display. This value can be any of the following:

sysuptime

Displays the system uptime.

sysnumusers

Displays the number of users.

sysprocess

Displays the number of processes.

nodename

Displays the host name.

memory

Displays the memory in kilobytes.

agentversion

Displays the SystemEDGE agent version.

systype

Displays the system release.

osversion

Displays the system version.

numcpu

Displays the number of CPUs.

virtualmemory

Displays the virtual memory in kilobytes.

totalswap

Displays the total swap space in kilobytes.

cpu1min

Displays the overall CPU busy percentage in the last minute.

cpu5min

Displays the overall CPU busy percentage in the last 5 minutes.

cpu15min

Displays the overall CPU busy percentage in the last 15 minutes.

avg1

Displays the 1 minute Load Average multiplied by 100.

avg5

Displays the 5 minute Load Average multiplied by 100.

avg15

Displays the 15 minute Load Average multiplied by 100.

openfiles

Displays the number of open files.

swapcap

Displays the swap capacity.

memcap

Displays the memory capacity.

meminusecap

Displays the memoryInUse capacity.

numneti

Displays the number of network interfaces.

sysedgemode

Displays the SystemEDGE mode.

Notes:

- The default port number for sysvariable is 161.
- Variable must be the last argument for sysvariable.
- You cannot query multiple variables in a single sysvariable call.

Examples

```
sysvariable -o sysprocess
```

```
sysvariable -h box1.domain.com -o agentversion
```

```
sysvariable -p 2009 -c admin -v 1 -o openfiles
```

```
sysvariable -c admin -v 2c -h Ea2f:fe90:abcd:0000:230:a2f:200:ad01 -o numcpu  
  
snmpget -p 2009 -u user3v3 -v 3 -s 1 -o memcap  
  
snmpget -h box1.domain.com -p 2009 -u user1v3 -A SHA -a osa -v 3 -s 2 -m 2 -o  
numneti  
  
snmpget -h 130.10.100.101 -p 2009 -u user2v3 -A SHA -a osa -X AES -x osp -v 3 -s  
3 -t 30 -o sysuptime
```

Old Usage

Important! The following old usage is deprecated. The use of the above argument format is strongly encouraged, as the old argument format will not be supported in the future.

```
sysvariable ipaddr[:port] commstr [options]
```

walktree Utility--Retrieve Values of OID Tree

walktree retrieves the value of every instance of every attribute that is defined in the MIB, from the specified OID through the last OID in the tree.

This utility has the following format:

```
walktree  
  [-h hostname | ip_addr]  
  [-p port]  
  [-c community]  
  [-v 1 | 2c | 3]  
  [-u secName]  
  [-s secLevel]  
  [-n contextName]  
  [-a authPassword] [-A MD5 | SHA]  
  [-x privPassword] [-X DES | AES | 3DES]  
  [-m FIPS_mode]  
  [-r retries]  
  [-t timeout]  
  [-d logLevel]  
  [-f logFile]  
  [-b]  
  [-o OID]
```

-h *hostname* | *ipaddr*

Specifies the hostname or IP address of the system on which the agent is running. Accepts IPv4 and IPv6 addresses.

Default: localhost

-p *port*

Specifies the UDP port that the agent is running on (for example, 1691).

Default: 161

-c *community*

Specifies a community string that the agent uses. Valid for SNMPv1 and SNMPv2c only.

Note: A read-write community string has to be specified for snmpset.

Default: public

-v *1* | *2c* | *3*

Indicates the version of SNMP that the agent is running. Specify 1 for SNMPv1, 2c for SNMPv2c, or 3 for SNMPv3.

Default: 1

-u *secName*

Specifies the name of the SNMPv3 secure user.

Default: none

-n *contextName*

Specifies the context name used by the agent if it is configured as SNMPv3.

Note: This option is not required for SNMPv3 communication.

Default: none

-a *authPassword*

Specifies the authentication password if the agent is configured for SNMPv3 with secLevel 2 (AuthNoPriv) or 3 (AuthPriv).

Default: none

-A *MD5* | *SHA*

Specifies the authentication protocol to be used by SNMPv3. This is required if the SNMPv3 user is configured with secLevel 2 (AuthNoPriv) or 3 (AuthPriv). Currently only MD5 (Message Digest Algorithm) and SHA (Secure Hash Algorithm, if the agent is configured for SNMPv3 with secLevel 2 (AuthNoPriv) or 3 (AuthPriv)) are used.

Default: MD5

-x *privPassword*

Specifies the privacy (encryption) password if the agent is configured for SNMPv3 with secLevel 3 (AuthPriv).

Default: none

-X *DES | AES | 3DES*

Specifies the privacy protocol if the SNMPv3 user is configured with secLevel 3 (AuthPriv). Specify DES for Data Encryption Standard, AES for Advanced Encryption Standard using cryptographic keys of 128 bits (AES128), and 3DES for Triple Data Encryption Standard.

Default: none

-m *FIPS_mode*

Controls the FIPS mode of operation. Accepted values are 0, 1, and 2.

0

Indicates Non-FIPS mode.

1

Indicates FIPS co-existence mode.

2

Indicates FIPS only mode.

-r *retries*

Specifies the number of retries.

Default: 3

-t *timeout*

Specifies the duration before the SNMP receiver considers the request as timed out.

Default: 10 seconds

-d logLevel

Specifies the log level of the SNMP messages. Accepted values are 0 to 5.

0

Logs fatal messages.

1

Logs critical messages.

2

Logs warning messages.

3

Logs informational messages.

4

Logs all of the messages.

5

Logs all of the messages including debugging messages.

Default: 0

-f logfile

Specifies the name of the log file that contains error and debug information. The default log file name for most of the utilities is `sysedge_utility.log`

Default: none

-b

Displays the value in hexadecimal format. Applies to `snmpget`, `snmpset`, and `walktree` utilities only.

-o OID

Specifies the object identifier (OID) to be set or queried for the `snmpget`, `snmpset`, and `walktree` utilities.

Default: none

Notes:

- The default port number for `walktree` is 161.
- OID must be the last argument for `snmpget`.
- You cannot query multiple OIDs in a single `walktree` call.

Examples

```
walktree -o 1.3.6.1.2.1.1
```

```
walktree -h box1.domain.com -p 2009 -c admin -v 1 -o 1.3.6.1.2.1.1.4

walktree -h Ea2f:fe90:abcd:0000:230:a2f:200:ad01 -c admin -v 2c -b -o
1.3.6.1.2.1.1

walktree -p 2009 -u user3v3 -v 3 -s 1 -h 130.10.100.101 -o 1.3.6.1.2.1.1

walktree -p 2009 -u user2v3 -A SHA -a osa -v 3 -s 2 -m 2 -o 1.3.6.1.2.1.1.4

walktree -p 2009 -u user1v3 -A SHA -a osa -X AES -x osp -v 3 -s 3 -t 30 -o
1.3.6.1.2.1.1
```

Old Usage

Important! The following old usage is deprecated. The use of the above argument format is strongly encouraged, as the old argument format will not be supported in the future.

```
walktree community ipaddr[:port][,timeout] mibpath outfile numRetries
```

xtrapmon Utility--Capture SNMP Traps

xtrapmon captures SNMP traps sent to a given UDP port on a system and displays the information contained in those traps. It can accept SNMPv1, SNMPv2c, and SNMPv3 traps and can function in IPv4 or IPv6 networks.

This utility has the following format:

```
xtrapmon
[-T]
[-p port]
[-e SNMPV3_config_file]
[-m FIPS_mode] [-l traps-log-file]
[-k debug_level]
[-h]
```

-T

(UNIX only) Runs xtrapmon in text mode. This option displays trap messages to the screen (stderr) and suppresses launching X windows popup dialogs.

-p port

Specifies the port number that xtrapmon listens for traps.

Default: 162

-e SNMPV3_config_file

Specifies the absolute path of the SNMPv3 configuration file. xtrapmon uses the default sysedgeV3.cf from the config sub-directory of the CA eHealth SystemEDGE installation directory.

-m FIPS_mode

Specifies the xtrapmon FIPS mode of operation. Accepted values are 0, 1, and 2.

0

Indicates non-FIPS mode.

1

Indicates FIPS co-existence mode.

2

Indicates FIPS only mode.

Note: FIPS mode requires installation of the CA eHealth Advanced Encryption package.

-l traps-log-file

Specifies the absolute path of the log file name to log the received traps.

-k debug_level

Specifies the log level of the SNMP messages to be logged in the xtrapmon.log file. Note that the usage of -k in xtrapmon differs from the other utilities, which use -d for the SNMP messages log level. Accepted values are 0 to 5.

0

Logs fatal messages.

1

Logs critical messages.

2

Logs warning messages.

3

Logs information messages.

4

Logs all of the messages.

5

Logs all of the messages including debugging messages.

-h

Displays the usage message for xtrapmon.

/?

Displays the usage message for xtrapmon (Windows only).

Examples

The following example starts xtrapmon on the default port 162 and the default SNMPv3 configuration file sysedgeV3.cf on UNIX and Windows:

```
xtrapmon
```

The following example starts xtrapmon on the non-default port 2091 using the SNMPv3 configuration file usersnmpv3.cf in FIPS only mode and logs the traps to the file usertraplog.txt:

UNIX

```
./xtrapmon -p 2091 -e /usr/temp/usersnmpv3.cf -m 2 -l /usr/temp/usertraplog.txt
```

Windows

```
xtrapmon -p 2091 -e \usr\temp\usersnmpv3.cf -m 2 -l \usr\temp\usertraplog.txt
```

The following example starts xtrapmon in text mode (UNIX only):

```
./xtrapmon -T -p 2091
```

The following example starts xtrapmon in text mode and suppresses the trap messages to the screen but logs them to a file (UNIX only):

```
./xtrapmon -T -p 2091 -l /usr/temp/usertraplog.txt 2>/dev/null
```

xtrapmon on UNIX Systems

On UNIX systems, xtrapmon can run as an X window application that uses Motif 2.1 or later libraries, or as a text-based console application.

- xtrapmon starts as an X window application by default using X windows dialogs. xtrapmon has a static window with the total number of received traps and a copyright text. It additionally opens alert dialogs every time a trap is received with the trap information. You can discard (close) these additional trap dialogs once the trap information is reviewed.
- You can start xtrapmon in text mode using the -T option. This mode logs trap messages to the user terminal (stderr) where xtrapmon is started.

You must install Motif 2.1 (or later) libraries to run xtrapmon on UNIX systems.

xtrapmon on Windows Systems

On Windows systems, xtrapmon is a text based console application. If a trap is received, it displays the trap information on the console.

Authentication in xtrapmon

xtrapmon does not validate SNMPv1/v2c community strings. It displays any SNMPv1/v2c trap that is received on the xtrapmon UDP port (default 162).

You can start xtrapmon with SNMPv3 user information using the default sysedgeV3.cf in the config sub-directory. To configure SNMPv3 user information, see SNMPv3 Configuration in the appendix "SNMPv3 in CA eHealth SystemEDGE".

xtrapmon only accepts SNMPv3 traps that match the SNMPv3 user information that xtrapmon starts with.

Trap Report Data

xtrapmon displays the following data about traps that it captures in a report:

Time

Specifies the local time of the host that receives the trap (the host that is running xtrapmon) for SNMPv1 traps. Specifies the time specified in the packet from the host that is sending the traps for SNMPv2/v3 traps.

Agent address

Specifies the address of the host sending the trap.

Agent Type

Specifies the agent Object Identifier (OID) that identifies the agent.

Specific Trap

Specifies the specific trap type (Trap sub-type) when the trap is an enterprise specific (6) trap. If the trap is not an enterprise specific trap, a value of 0 (zero) displays.

Trap Type

Specifies the trap type of the received trap. Displayed values are 0 to 6.

0

Cold Start

1

Warm Start

2

Link Failure

3

Link Up

4

Authentication Failure

5

EGP Neighbor Lost

6

Vendor Specific (also known as Enterprise)

Additional Command Line Utilities

This section describes the command line utilities that are not SNMP based. These utilities are provided as sample programs to use as action items with CA eHealth SystemEDGE monitors. Actions are executed when a monitor table entry evaluates to True. Following are the utilities described in this section:

- bounce.exe (Windows only)
- checkfile.exe
- email.exe
- getver.exe (Windows only)
- nt4bigmem.exe (Windows only)
- restartproc.exe (Windows only)
- restartsvc.exe (Windows only)
- restartproc.sh (UNIX only)

bounce.exe Utility--Forcibly Reboot the System (Windows Only)

bounce.exe tries to shutdown or reboot the Windows system. It forcibly closes all applications that are currently open. It does not prompt you to save your work. This utility is available on Windows only.

This utility has the following format:

bounce.exe [OPTIONS]

OPTIONS

Specifies the available options for the command. The following options are available:

-h or /h , -? or /?

Displays the bounce.exe usage message.

-H

Displays helpful examples.

-r or /r

Reboots the system.

-s or /s

Shuts down the system.

-a or /a

Aborts the reboot or shutdown.

-t{n} or /t{n}

Sets the timeout to {n}.

Default: 20 sec

-d{message}

Displays a message before shutting down.

Examples

```
bounce.exe -r -t300 -d"You have 5 minutes to save your work before the system  
will be rebooted"
```

```
bounce.exe -a
```

checkfile.exe Utility--Display the File Size

checkfile.exe displays the file size of a file that you specify. If the file doesn't exist, it returns 0 (zero).

This utility has the following format:

```
checkfile.exe [OPTIONS] filename
```

OPTIONS

Specifies the available options for the command. The following options are available:

-v

Displays detailed (verbose) information. This option must precede the -s option.

-sb

Displays the file size in bytes.

-sk

Displays the file size in kilobytes.

-sm

Displays the file size in megabytes.

-sg

Displays the file size in gigabytes.

-h

Displays the checkfile.exe usage message.

filename

Specifies the absolute path name of the file whose size you want to display.

Examples

Windows

```
checkfile.exe -sk C:\pagefile.sys
```

UNIX

```
checkfile.exe -sm /var/log/syslog
```

email.exe Utility--Send an Email

email.exe sends an email. Use this utility as an action to send an email based on SystemEDGE monitoring or CA eHealth TrapEXPLODER activity. email.exe can work in both IPV4 and IPv6 environments.

This utility has the following format:

```
email.exe
  [-v]
  [-s]
  [-a]
  [-r smtp-server]
  [-xhdrf filename ]
  source-addr
  dest-addr
  [subject]
  [message]
```

-v

Enables verbose mode, and prints useful information to the screen.

-s

Lets you add more text to the message body by typing it on the terminal (stdin). Use ^D (CTRL+D) on UNIX to finish the message and ^Z (CTRL+Z) on Windows. Note that you still need to specify subject or message arguments when this option is specified.

-a

Removes the requirement to specify a subject or a message body. This option is most useful when used with a SystemEDGE action.

-r *smtp_server*

Lets you specify the mail server name to use when sending the email. By default, the program looks up the MX record of the host in the 'To' address, and tries to send the email that way. However, if the program is unable to connect to the destination mail exchanger (due to a firewall), you can send the message through the local mail server, specified by the -r option.

-xhdrf *filename*

Adds user defined information from a file to the email header. You can use this option to specify ISO character set information.

source-addr

Specifies the source email address in the format user@domain.

dest-addr

Specifies the destination email address in the format user@domain.

subject

Specifies the subject text of the email. Enclose the subject text in single quotes if the text contains more than one word.

message

Specifies the message body of the email. Enclose the message text in single quotes if the text contains more than one word.

Examples

```
email.exe -r mail.foo.com source@foo.com dest@foo.com 'email subject' 'email message'
```

```
email.exe -v -r mail.foo.com source@foo.com dest@foo.com 'email subject' 'email message'
```

```
email.exe -v -s -r mail.foo.com source@foo.com dest@foo.com 'email subject' 'email message'
```

getver.exe Utility--Display File Information (Windows Only)

getver.exe displays the product name, company name, file version, private build, and file description for a file that you specify if this information can be found. This utility is available on Windows only.

This utility has the following format:

```
getver.exe [OPTIONS] -f <filename>
```

OPTIONS

Specifies the available options for the command. The following options are available:

-h or /h, -? or /?

Displays the getver.exe usage message.

-H or /H

Displays helpful examples.

-p

Displays the product name if the file is found.

-c

Displays the company name if the file is found.

-v

Displays the file version if the file is found.

-b

Displays the private build version if the file is found.

-d

Displays the file description if the file is found.

-f <filename>

Specifies the absolute path name of the file whose information you want to display.

Examples

```
getver.exe -p -c -v -b -d -f C:\winnt\system32\kernel32.dll
```

```
getver.exe -f c:\sysedge\sysedge.dll
```

nt4bigmem.exe Utility--Display Memory Information (Windows Only)

nt4bigmem.exe displays the total memory and free (available) memory. This utility is available on Windows only.

This utility has the following format:

```
nt4bigmem.exe
```

restartproc.exe Utility--Restart a Process (Windows Only)

restartproc.exe can restart a NT process (program) through an action invocation. This utility is mainly used as an action for a process monitor table entry. Before running this utility, ensure that the corresponding process monitoring entry is already created. This utility is available on Windows only.

This utility has the following format:

```
restartproc <arglist>
```

arglist

Consists of the following 11 arguments. You must provide all of the following arguments for successful execution:

arg 1

Specifies the program binary to restart.

arg 2

Specifies the empire trap type.

arg 3

Specifies the process monitoring entry index.

arg 4

Specifies the process monitoring entry description.

arg 5

Specifies the process monitoring entry attribute.

arg 6

Specifies the process monitoring entry operator.

arg 7

Specifies the process monitoring entry current value.

arg 8

Specifies the process monitoring entry threshold value.

arg 9

Specifies the process monitoring entry flags.

arg 10

Specifies the process monitoring entry regular expression.

arg 11

Specifies the process monitoring entry current process id.

Examples

The following example invokes restartproc.exe in a sysedge.cf entry:

```
watch process procAlive 'testapp|TESTAPP' 1000 0x0 30 'testapp restart'  
'C:\sysedge\bin\restartproc.exe c:\testapp.exe'
```

You can also invoke restartproc.exe on the command line. The following example restarts process ID 3024:

```
c:\sysedge\bin\restartproc.exe c:\testapp.exe 10 1000 "testapp restart" 1 3 6 4  
0x0 "testapp|TESTAPP" 3024
```

restartsvc.exe Utility--Restart a Service (Windows Only)

restartsvc.exe restarts a NT service through an action invocation. This utility is mainly used as an action for a process monitor table entry. Before running this utility, ensure that the corresponding process monitoring entry is already created. This utility is available on Windows only.

This utility has the following format:

```
restartsvc <arglist>
```

arglist

Consists of the following 11 arguments. You must provide all of the following arguments for successful execution:

arg 1

Specifies the NT service to restart.

arg 2

Specifies the empire trap type.

arg 3

Specifies the process monitoring entry index.

arg 4

Specifies the process monitoring entry description.

arg 5

Specifies the process monitoring entry attribute.

arg 6

Specifies the process monitoring entry operator.

arg 7

Specifies the process monitoring entry current value.

arg 8

Specifies the process monitoring entry threshold value.

arg 9

Specifies the process monitoring entry flags.

arg 10

Specifies the process monitoring entry regular expression.

arg 11

Specifies the process monitoring entry service index from the NT Service table.

Examples

The following example invokes restartsvc.exe in a sysedge.cf entry:

```
watch process procAlive 'testsvc|TESTSVC' 2000 0x8 30 'testsvc restart'  
'C:\sysedge\bin\restartsvc.exe c:\testsvc.exe'
```

You can also invoke restartsvc.exe on the command line. The following example restarts process ID 125:

```
c:\sysedge\bin\restartsvc.exe c:\testsvc.exe 10 2000 "testsvc restart" 1 3 6 4  
0x8 "testsvc|TESTSVC" 125
```

restartproc.sh Utility--Restart a Process (UNIX Only)

restartproc.sh restarts a process through an action invocation. This utility is mainly used as an action for a process monitor table entry. Before running this utility, ensure that the corresponding process monitoring entry is already created. This utility is available on UNIX only.

This utility has the following format:

```
restartproc <arglist>
```

arglist

Consists of the following two arguments. You must provide both arguments for successful execution when running the utility on the command line:

arg 1

Specifies the program to restart.

arg 2

Specifies the integer value of 10 that is equivalent to PROCSTOP_TRAP.

Note: arg 2 is not required when it is defined as the action item for the process monitor entry. See the example below.

Examples

The following example invokes restartproc.sh in a sysedge.cf entry:

```
watch process procAlive 'lpd' 123 0x8 60 'Printer Service Alive'  
'/opt/EMPsysedge/bin/restartproc.sh /usr/bin/lpd'
```

You can also invoke `restartproc.sh` on the command line as follows:

```
c:\sysedge\bin\restartproc.sh /usr/bin/lpd 10
```

Chapter 19: Troubleshooting and Usage Suggestions

This chapter presents helpful tips for using the CA eHealth SystemEDGE agent. For explanations of the error and warning messages that the CA eHealth SystemEDGE agent and its associated utilities provide, see the appendix "Error Messages."

For the most current information, refer to the CA eHealth Support Web site at <http://support.concord.com>.

This section contains the following topics:

[Using diagsysedge.exe](#) (see page 413)

[Common Problems and Questions](#) (see page 415)

Using diagsysedge.exe

You can use the diagsysedge.exe program to verify that the agent is running and to obtain information about the agent that you can use for troubleshooting

The examples in this section assume that the agent is configured to use SNMPv1 or SNMPv2c and has a read community of public and is on port 161 or 1691.

Note: For more information about all of the options supported by diagsysedge.exe, see the chapter "Command Line Utilities".

Determine Whether the Agent Is Running

You can run diagsysedge.exe with the -B option to determine whether the agent is running.

To determine whether the agent is running

1. Change to the sysedge/bin directory.

2. Enter **./diagsysedge.exe -B**.

The command provides output similar to the following:

```
#./diagsysedge.exe -B
diagsysedge: (V1.03 - LINUX)
egyptian
2.6.9-11.ELsmp, Red Hat Enterprise Linux AS release 4 (Nahant Update 1)#1 SMP
Fri May 20 18:26:27 EDT 2005
```

If the agent is *not* running, you receive an error message that the SNMP operation failed. Check the port number, and if the agent is started on a port other than 161, specify -p <port> on the command line and try again, or you can attempt to start the agent again, using the instructions in the chapter "Starting the CA eHealth SystemEDGE Agent."

Obtain a Report for Troubleshooting

If you are encountering problems with your CA eHealth SystemEDGE agent, Technical Support may ask you to run diagsysedge.exe to generate a report that they can use for troubleshooting.

To generate a report that you can send to Technical Support

1. Change to the sysedge/bin directory.
2. Enter **./diagsysedge.exe**.

The command creates a diagsysedge.txt file that provides information similar to the following:

- Version of the diagsysedge.exe program
- Current date and time
- IP address of the system on which you are running the agent
- Operating system platform
- Community string (if the agent is configured for SNMPv1)
- Port on which the agent is running
- Timeout value
- Location of the sysedgeddiag.txt output file
- Contents of the sysedge.cf, sysedge.lic, and sysedge.mon files (from the /etc [UNIX] or %SystemRoot%\System32 directory [Windows])
- Information about registry settings that the setup program modified
- A detailed System Information report

If the agent is *not* running, you receive an error message that the SNMP operation failed. Attempt to start the agent again, using the instructions in this chapter for your operating system.

Common Problems and Questions

This section describes problems that might occur when you are installing and using the CA eHealth SystemEDGE agent. It also provides instructions for resolving these problems. This section is organized by observable symptoms of a problem, and questions for each symptom to help you isolate the cause.

Agent Not Responding to SNMP Requests

When the agent is not responding to SNMP requests, the NMS software cannot query the CA eHealth SystemEDGE agent. To resolve this problem, you must verify that the agent is running, that it is properly configured, and that the management system software is properly configured.

Is the SystemEDGE Agent Running?

You can verify that the agent is running in one of the following ways, depending on your platform.

To verify that the agent is running with netstat

On UNIX

1. Enter **netstat -a** and look for UDP/161, UDP/1691, or SNMP.
2. Enter **ps -aux** or **ps -aef** and look for sysedge.
3. Run walktree or snmpget.
4. Examine system log files for agent error messages.

On Windows

1. Select Start, Control Panel, double-click CA eHealth SystemEDGE, and view the status.
2. Open the Windows Task Manager and select the Processes tab. Look for the process name sysedge.exe. If this process appears in the list, CA eHealth SystemEDGE is running.
3. Select Start, Control Panel, Administrative Tools, Services, and look for SystemEDGE in the Services dialog.
4. Run walktree or snmpget.
5. Examine %SystemRoot%\system32\sysedge.log.

Is the SystemEDGE Agent Starting at System Initialization?

To verify that the agent is started at system initialization:

- Check the UNIX startup script.
- Make sure that the SystemEDGE service is configured for automatic startup.

Is the CA eHealth SystemEDGE Agent Responding to Queries?

Use the sysvariable utility to query a system and prove that the agent is responding to queries. You must specify the port number (unless you are using the default port of 161) for the CA eHealth SystemEDGE agent and the community string for your system if the CA eHealth SystemEDGE agent is configured to accept SNMPv1 or SNMPv2c communication. Otherwise, you must specify all necessary SNMPv3 communication (depending on the security level - Authpasswd, Authprotocol, Privpasswd, Privprotocol).

To display the operating system release number for a UNIX system where the CA eHealth SystemEDGE agent is on port 1691 and the community string is public, enter the following for SNMPv1:

```
#./sysvariable -h 127.0.0.1 -p 1691 -c public -v 1 -o systype
```

To display the operating system release number for a UNIX system where the CA eHealth SystemEDGE agent is on port 1691 and the community string is public, enter the following for SNMPv2c:

```
#./sysvariable -h 127.0.0.1 -p 1691 -c public -v 2c -o systype
```

To display the operating system release number for a UNIX system where the CA eHealth SystemEDGE agent is on port 1691 and the SecLevel is configured to AuthPriv(3) with secuser as the USM user name, authpasswd as the Authentication password, MD5 as the Authentication protocol, privpasswd as the Privacy password and DES as the Privacy protocol, enter the following for SNMPv3:

```
#./sysvariable -h 127.0.0.1 -p 1691 -v 3 -u secuser -s 3 -a authpasswd -A MD5 -x privpasswd -X DES -o systype
```

For a Windows system where the CA eHealth SystemEDGE agent is on port 1691 and the community string is public, enter the following for SNMPv1:

```
C:\sysedge\bin>sysvariable.exe -h 127.0.0.1 -p 1691 -c public -v 1 -o systype
```

For a Windows system where the CA eHealth SystemEDGE agent is on port 1691 and the community string is public, enter the following for SNMPv2c:

```
C:\sysedge\bin>sysvariable.exe -h 127.0.0.1 -p 1691 -c public -v 2c -o systype
```


For a Windows system where the CA eHealth SystemEDGE agent is on port 1691 and the SecLevel is configured to AuthPriv(3) with secuser as the USM user name, authpasswd as the Authentication password, MD5 as the Authentication protocol, privpasswd as the Privacy password and DES as the Privacy protocol, enter the following for SNMPv3:

```
C:\sysedge\bin>sysvariable.exe -h 127.0.0.1 -p 1691 -v 3 -u secuser -s 3 -a  
authpasswd -A MD5 -x privpasswd -X DES -o systype
```

Are the CA eHealth SystemEDGE Agent Binaries and Configuration Files Installed?

To verify that the correct agent binaries and configuration files are installed, see Agent Does Not Run on a Particular Operating System Version in this chapter.

Is the CA eHealth SystemEDGE Agent Co-Existing with Other Agents?

If you are using other agents with the CA eHealth SystemEDGE agent, verify their coexistence by entering the following on Windows:

```
setup.exe -c -v
```

Is the CA eHealth SystemEDGE Agent Configured Correctly?

To verify that the agent is configured correctly, check the access control lists. For SMPv2 and SNMPv2c, check the communities in sysedge.cf. For SNMPv3, check the USM security configuration in sysedgeV3.cf.

- On UNIX systems, examine /etc/sysedge.cf and /opt/EMPSysedge/config/sysedgeV3.cf for community strings.
- On Windows systems, verify access controls and SNMP v1 and SNMPv2c communities through the Network Control Panel. Verify USM security information in \opt\EMPSysedge\config\sysedgeV3.cf

Note: sysedgeV3.cf may be encrypted.

For more information, see se_enc Utility--Encrypt the SNMPv3 Configuration File in the chapter "Command Line Utilities".

Is the Management System Software Configured Correctly?

To verify that the management system software is correctly configured, check to see that it is querying the correct system and port number if the CA eHealth SystemEDGE agent is running on an alternate UDP port (for example, on UDP/1691).

Management System Not Receiving SNMP Trap Messages

If the management system is not receiving SNMP traps from the CA eHealth SystemEDGE agent, you must verify that the agent is sending Trap PDUs and that it is sending them to the correct addresses. If you have already verified those conditions, the problem is most likely due to misconfiguration of the management system.

Is the CA eHealth SystemEDGE Agent Running?

To verify that the agent is running, see Agent Not Responding to SNMP Requests in this chapter.

Is the CA eHealth SystemEDGE Agent Correctly Configured to Send Trap Messages?

To verify that the agent is configured to send trap messages to the appropriate addresses, do one of the following:

- For SNMPv1, verify the trap communities in the sysedge.cf file.
- For SNMPv3, verify the USM security configuration in the sysedgeV3.cf file.
- On UNIX systems, verify the trap communities in the sysedge.cf file. For example, make sure that only one IP address is specified for each trap community statement. For more information about trap community configuration, see the chapter "Configuring the CA eHealth SystemEDGE Agent."
- On Windows systems, verify the trap configuration (SNMPv1 only) through the Control Panel Network application. For more information, see Configuring Trap Communities in the chapter "Configuring the CA eHealth SystemEDGE Agent."

Is the SystemEDGE Agent Sending Traps?

To verify whether the SystemEDGE agent is sending traps, do the following:

- On UNIX systems, examine the syslog log files.
- On Windows systems, examine %SystemRoot%\system32\sysedge.log

Is the Management System Configured Correctly?

To verify the proper configuration of the management system, make sure that the empire.asn1 file has been imported into the management station (so that the management system knows the format of SNMP Trap PDUs generated by the CA eHealth SystemEDGE agent). For more information about troubleshooting the management system software, contact your management system vendor.

Agent Does Not Run on A Particular Operating System Version

If the CA eHealth SystemEDGE agent was installed on an operating system on which it does not run, the agent displays a message similar to the following when it starts for the first time after installation:

This agent binary is compiled for X, not Y

Because the CA eHealth SystemEDGE agent works closely with the underlying operating system, each release may require different binaries. For example, kernel data structures and application programmer interfaces may change from release to release, sometimes in subtle ways.

Therefore, for versions of Solaris 2.x, the CA eHealth SystemEDGE agent uses different binaries for each release. The error message indicates that you are attempting to run a CA eHealth SystemEDGE agent binary on an operating system release for which it is not compiled.

To make sure that your system is executing the correct CA eHealth SystemEDGE agent binary, examine the UNIX startup script and verify that it is selecting the appropriate binary.

Note: For more information the appropriate binary file for your operating system and a list of supported operating systems, see the *CA eHealth SystemEDGE Release Notes*.

Bind Failed: Address Already In Use

If a bind fails because the address is in use, the CA eHealth SystemEDGE agent displays a message similar to the following when it first starts:

```
sysedge: bind call failed: Address already in use  
sysedge: another agent is probably running on port X
```

This message most often occurs when another SNMP agent is already up and running and is bound to port UDP/161, which is the default, well-known SNMP agent port.

For more information about using the CA eHealth SystemEDGE agent with other SNMP agents, see the chapter "Using the CA eHealth SystemEDGE Agent with Other SNMP Agents." For more information about using CA eHealth SystemEDGE on an alternative UDP port, see the chapter "Starting the CA eHealth SystemEDGE Agent."

Update the Monitor Configuration File

The sysedge.mon monitor configuration file serves as non-volatile backing store to the agent's in-memory self-monitoring tables. When SNMP updates are made to any of the agent's in-memory self-monitoring tables, those tables are written to sysedge.mon. Consequently, the agent does not interpret updates to sysedge.mon that occur when it is running until it is restarted. Also, updates made to sysedge.mon while the agent is running may be lost if the agent writes its in-memory monitor table over the tables in sysedge.mon. In addition, configuration file directives in sysedge.cf take precedence over those in sysedge.mon.

To update the CA eHealth SystemEDGE agent's monitor configuration file

1. Open a command prompt.
2. Stop the agent.
 - On Windows, enter the following:
`net stop sysedge`
 - On Solaris, enter the following when you are logged in as root:
`/etc/init.d/sysedge stop`

Note: For other UNIX operating systems, enter the path and name for your CA eHealth SystemEDGE startup script. For a list of startup script names, see Service Startup Script for UNIX Systems in the chapter "Starting the CA eHealth SystemEDGE Agent".

3. Use a text editor to update sysedge.mon.
4. Restart the agent.
 - On Windows, enter the following:
`net start sysedge`
 - On Solaris, enter the following when you are logged in as root:
`/etc/init.d/sysedge start`

Note: For other UNIX operating systems, enter the path and name for your CA eHealth SystemEDGE startup script. For a list of startup script names, see Service Startup Script for UNIX Systems in the chapter "Starting the CA eHealth SystemEDGE Agent".

5. Verify that the updated sysedge.mon file does not contain any errors. To do so, examine the sysedge.log file on Windows or the syslog file output on UNIX for error messages pertaining to the monitor configuration file.

Note: You can update the sysedge.mon file directly, but you should not need to; configuration file directives exist to create supported types of sysedge.mon entries.

If Entries Are Not Being Added to sysedge.mon

If you are adding entries to sysedge.mon through SNMP and find that the file is not being updated, the sysedge.mon file may have changed at some point to have read-only permissions.

When sysedge.mon is read-only and you attempt to add new entries, the following occurs:

- sysedge.mon regains read-write permissions.
- Two new files are created:
 - sysedge.new contains the current content of sysedge.mon plus any new entries you added from the time it was set to read-only.
 - sysedge.old contains the read-only version of sysedge.mon.

To resolve the problem and commit your changes to sysedge.mon

1. Verify that the sysedge.mon, sysedge.new, and sysedge.old files exist in the *systemRoot\system32* directory.
2. Enter the following commands at the command line:

```
Attrib -r sysedge.*
Del systemRoot\system32\sysedge.old
Del systemRoot\system32\sysedge.mon
Move systemRoot\system32\sysedge.new sysedge.mon
```

How to Automatically Restart Processes

The CA eHealth SystemEDGE agent can perform actions when an entry in any of the agent's self-monitoring tables evaluate to True. You can use this capability to restart processes, clean up file systems, and so on.

To set the CA eHealth SystemEDGE agent to restart processes when they fail

1. Create an entry in the Monitor table to monitor a particular application or process. (You can create an entry through monprocess, through an SNMP management station, or by editing the sysedge.mon file in a text editor.)
2. Make sure that entry contains an action script. That action script is responsible for restarting the application and resetting the monitor-table entry for the application or process. The CA eHealth SystemEDGE agent distributions come with an example script, restartproc.sh, that performs this function.

When the process fails, the monitor table entry evaluates to True, sends an SNMP trap to the management system, and runs the action script. The action script restarts the application, waits a few seconds, and then calls monprocess to reinitialize the monitor-table entry so that the agent will resume monitoring the new process.

Implementing Trap Severity Levels

Currently, the trap messages generated by the CA eHealth SystemEDGE agent contain many variables, but not an explicit severity level; however, the trap messages generated by the self-monitoring tables contain an ASCII description field. You can implement severity levels if they are encoded within the corresponding description string for the particular table row in question.

Using this approach, you can assign severity levels like Critical or Warning, and you can include those severity levels within the table-description field. For example, you can use the description with file-system monitoring to indicate that a Critical event occurs when the file system is more than 95% full with a message similar to the following: "Critical: / filesystem over 95% full."

When management software receives a trap message, it can easily identify the severity level using the description string included in applicable trap messages. The description string provides an easily readable method for indicating severity rather than one that requires specialized management-station coding or modification.

Required and Recommended System Patches

The CA eHealth SystemEDGE agent requires few, if any, system patches for proper functioning. For the latest information about patches, see the *Release Notes*.

Appendix A: Error Messages

The error messages are organized based on their type into the following groups:

- CA eHealth SystemEDGE agent error messages
- Command-line utility error messages

This section contains the following topics:

[CA eHealth SystemEDGE Agent Error Messages](#) (see page 423)

[Command-line Utility Error Messages](#) (see page 471)

CA eHealth SystemEDGE Agent Error Messages

This section describes the error messages that you may get while using the CA eHealth SystemEDGE agent. Depending on the error, the message will either be printed to standard error or logged through the syslog utility. These error messages should be interpreted in conjunction with the `sysedge_snmp.log` in the `bin` subdirectory of the agent's installation.

Note: The messages in this section are sorted alphabetically.

action execution failed

Reason:

The CA eHealth SystemEDGE agent failed to run an action command due to an error.

Action:

Check the action script name and arguments.

authenFailure src: *ipaddress* community *filename*

Reason:

The CA eHealth SystemEDGE agent has sent an SNMP authentication failure trap message because it does not permit SNMP queries using a specific community *filename* from a specific *IP address*.

Action:

None. Information only.

bad ioconfig magic

Reason:

Upon startup, the CA eHealth SystemEDGE agent uses the available operating system support and attempts to discover the underlying system devices.

This error message indicates that the CA eHealth SystemEDGE agent could not read the HP-UX file etc/ioconfig and will therefore; attempt to discover the devices without the operating system's aid.

Action:

None. Information only.

bad pid filename for write_runStatus

Reason:

An attempt to change the host resources hrRunStatus variable for a process has failed due to an invalid Process ID (PID).

Action:

Double-check the PID value for the specific process and run the management operation again.

bad shellOutput directory filename

Reason:

The specified directory for remoteShell invocation output is invalid.

Action:

Put all the output from remoteShell invocations in /tmp or \tmp.

bad shellOutput file filename

Reason:

The designated remoteShell output file is invalid because it is either not a file (for example, it is a directory) or it is in an invalid directory (for example, not in /tmp folder).

Action:

None. Information only.

bind address *IPaddress* is not a valid ip address**Reason:**

The IP address that is specified using the bind_address keyword in the CA eHealth SystemEDGE configuration file (sysedge.cf) is invalid.

Reason:

Verify that the specified IP address is valid using tools like ping.

block size for *filename* is 0, using 1K**Reason:**

The CA eHealth SystemEDGE agent failed to determine a file system's underlying block size and is using 1000 bytes as a default block size for calculations in the Systems Management MIB Mounted Devices table.

Action:

None. Information only.

Cannot bind to socket, *socketNumber*, SNMP_SnmplistenInitialize (*socket*) call failed**Reason:**

The CA eHealth SystemEDGE agent attempted to bind to a UDP port and failed. If that UDP port is already in use, another SNMP agent is already running that system and is using that port.

This problem commonly occurs when multiple agents attempt to use the same UDP port (for example, UDP/161). The Internet Assigned Numbers Authority (IANA) has reserved UDP/1691 for the CA eHealth SystemEDGE agent.

Action:

You can configure the CA eHealth SystemEDGE agent to use that port by specifying it on the command line using the -p option.

cannot find title index**Valid on Windows systems only****Reason:**

The CA eHealth SystemEDGE agent could not initialize the relevant information to read a particular performance variable from the Windows registry.

Only this variable is affected. All other Windows variables should be supported.

Action:

None. Information only.

cannot locate disk pstat data, diskStatsTable not supported

Valid on HP-UX

Reason:

The CA eHealth SystemEDGE agent could not locate the appropriate disk statistics through an HP-UX application programmer interface (API).

Disk statistics for the particular drive are, therefore, not supported. Disk statistics for other drives, however, should be unaffected.

Action:

None. Information only.

cannot open kmem

Valid on UNIX

Reason:

The CA eHealth SystemEDGE agent reads the device kmem to retrieve certain kernel parameters and statistics. This error message indicates that the agent could not open the /dev/kmem device perhaps due to permissions problems.

Action:

Verify that the agent has read access to /dev/kmem.

cannot open socket for mib-2

Valid on UNIX

Reason:

The CA eHealth SystemEDGE agent uses a socket to obtain MIB-II statistics. This error message indicates that the agent could not obtain such a socket. Most likely, the agent will be unable to support MIB-II related statistics but should otherwise operate normally.

Action:

None. Information only.

caught SIGHUP**Reason:**

The CA eHealth SystemEDGE agent caught a UNIX hang-up signal and is continuing to operate normally.

Action:

None. Information only.

config file error in subprogram_user_name directive**Reason:**

The CA eHealth SystemEDGE agent configuration file contains an error in the subprogram directive. The directive will be ignored.

Action:

Fix the statement and restart the agent.

config syntax error, line *linenumber***Reason:**

The CA eHealth SystemEDGE agent has encountered a syntax error in its configuration file at line *linenumber*.

Action:

The offending line is ignored and the rest of the configuration file is parsed.

could not find a valid license for machine *machinename***Valid on Windows, UNIX****Reason:**

The CA eHealth SystemEDGE agent could not find a valid license for the system on which it is operating.

Action:

Check that the CA eHealth SystemEDGE agent license file is in the correct location and contains a valid license.

On Windows systems, the configuration file is located in the system root directory, %SystemRoot%\system32\. On UNIX systems, the CA eHealth SystemEDGE agent looks for the file in the /etc directory.

could not fork sub-shell

Reason:

The CA eHealth SystemEDGE agent could not create or invoke a sub-shell to process a remoteShell operation.

Action:

Check if the system process table is full. If it is full, additional processes cannot be created.

Could not open SNMPv3 configuration file: *file*

Reason:

CA SystemEDGE could not open the SNMPv3 configuration file.

Action:

Verify that the SNMPv3 configuration file exists. If you specify the SNMPv3 configuration file using the -e option, make sure that it is specified with an absolute pathname.

counterType *typename* not supported. Entry will not be added.

Reason:

The CA eHealth SystemEDGE agent could not support registry extension objects of type *typename* because the agent cannot understand it. (There is no way to map it to an SNMPv1 object type.) The agent will not support this registry extension object.

Action:

None. Information only.

CreateEvent for traps failed

Valid on Windows systems only

Reason:

The CA eHealth SystemEDGE agent must create an internal operating system event resource for use with sending SNMP traps. This error message indicates that the event creation has failed. Consequently, the CA eHealth SystemEDGE agent may not be able to send private-enterprise trap messages.

Action:

None. Information only.

createMutex failed for query mutex**Valid on Windows systems only****Reason:**

The CA eHealth SystemEDGE agent uses an operating system mutex resource to control access to structures shared with the master agent. Consequently, the CA eHealth SystemEDGE agent may not be able to operate properly.

Action:

None. Information only.

createMutex failed for result mutex**Valid on Windows systems only****Reason:**

The CA eHealth SystemEDGE agent uses an operating system mutex resource to control access to structures shared with the master agent. Consequently, the CA eHealth SystemEDGE agent may not be able to operate properly.

Action:

None. Information only.

/dev/lan is missing. SystemEDGE cannot continue.**Valid on HP-UX****Reason:**

The CA eHealth SystemEDGE agent requires a /dev/lan device to run on HP-UX systems. It will not start if it cannot find that device. Instead, it will log a message to syslog and exit.

Action:

None. Information only.

For more information about the /dev/lan device, see the HP-UX documentation.

discovering HPUX devices by hand

Valid on HP-UX

Reason:

The CA eHealth SystemEDGE agent discovers the devices manually rather than through automated techniques. This discovery method is used on older versions of HP-UX, which do not support automated methods.

Action:

None. Information only.

dkiotime read failed, no disk stats

Reason:

The CA eHealth SystemEDGE agent could not read a disk statistic structure of the kernel. Consequently, disk statistics may not be supported.

Action:

None. Information only.

empire_agent_init failed

Valid on Windows systems only

Reason:

The CA eHealth SystemEDGE agent failed to initialize due to licensing errors, invalid or non-existent configuration files, or some other problem.

Action:

Check the additional error messages printed by the CA eHealth SystemEDGE agent for the cause of the problem.

error parsing *errornumber* in monitor file, line *linenumber*

Reason:

The CA eHealth SystemEDGE agent encountered an error while parsing the sysedge.monitor configuration file. It prints the offending line number.

Action:

None. Information only.

event regcomp failed, *filename* *desc_filt***Reason:**

The CA eHealth SystemEDGE agent failed to compile a Windows event-monitoring regular expression because it was invalid.

Action:

Verify the regular expression and try the management operation again.

executing subprograms as group *groupname***Valid on UNIX****Reason:**

The CA eHealth SystemEDGE agent runs all the subprograms with permissions of the specific *groupname*, as specified in a configuration file directive.

Action:

None. Information only.

executing subprograms as user *username***Valid on UNIX****Reason:**

The CA eHealth SystemEDGE agent runs all the subprograms with permissions of the specific user, as specified in a configuration file directive.

Action:

None. Information only.

execv failed for action**Reason:**

The CA eHealth SystemEDGE agent failed to run the action command. Exec failures can occur if the command is invalid, if the binary or shell script no longer exists in the specified location, or if the execute permissions are not properly set.

Action:

Check the action command manually and, if appropriate, update the CA eHealth SystemEDGE agent configuration file, *sysedge.cf*.

execv failed for extension command

Reason:

The CA eHealth SystemEDGE agent failed to run the extension command. Exec failures can occur if the command is invalid, if the binary or shell script no longer exists in the specified location, or if the execute permissions are not properly set.

Action:

Check the extension command (for example, manually) and if appropriate, update the CA eHealth SystemEDGE agent configuration file.

extension command file *filename* is not a regular file

Reason:

The extension command file is not a valid executable or shell script.

Action:

Check the specified file and verify that it is not a directory, device, and so on.

For more information about extending the CA eHealth SystemEDGE agent, see the chapter "Adding Custom MIB Objects."

extension filename too long

Reason:

The extension command file name is too long. The CA eHealth SystemEDGE agent limits command file names to 256 characters.

Action:

Check the extension statement in the configuration file and restart the CA eHealth SystemEDGE agent.

For more information about extending the CA eHealth SystemEDGE agent, see the chapter "Adding Custom MIB Objects."

extension variable *variablename* already in use**Reason:**

The extension command variable specified in the configuration file is already in use. Extension commands must specify a number that has already been used.

Action:

Check the configuration file duplicate extension command numbers and restart the CA eHealth SystemEDGE agent.

For more information about extending the CA eHealth SystemEDGE agent, see the chapter "Adding Custom MIB Objects."

failed to add monitor entry index *entryname***Reason:**

The CA eHealth SystemEDGE agent failed to add a Monitor table entry. This error usually results from bad parameters, which are most often due to invalid intervals, Boolean operators, object identifiers, and so on.

Action:

Fix the monitor entry and restart the CA eHealth SystemEDGE agent.

For more information about the correct syntax for Monitor table entry commands, see the chapter "Configuring Threshold Monitoring."

failed to alloc anIDE struct**Reason:**

The CA eHealth SystemEDGE agent failed to allocate memory for an IDE device, most likely because the underlying system is low on memory. The CA eHealth SystemEDGE agent will continue to operate.

Action:

None. Information only.

failed to allocate history entry

Reason:

The CA eHealth SystemEDGE agent failed to allocate memory for a history table entry. It will therefore not create a history table entry and the configuration file statement will be ignored. The CA eHealth System EDGE agent will, however, continue to parse the remainder of the configuration file.

Action:

None. Information only.

failed to alloc space for monitor

Reason:

The CA eHealth SystemEDGE agent failed to allocate memory for its Monitor table, most likely because the underlying system is low on memory. The CA eHealth SystemEDGE agent will continue to operate, but it will not be able to perform self-monitoring.

Action:

None. Information only.

failed to alloc space for SNMPv3 config file

Reason:

The CA eHealth SystemEDGE agent could not allocate internal space for storing the file name of the SNMPv3 configuration file.

Action:

Verify that the system has enough memory resources and then restart the agent.

failed to create a trap session

Reason:

The CA eHealth SystemEDGE agent failed to allocate the resources necessary to send SNMP trap messages, most likely because of an error in the underlying system.

The CA eHealth SystemEDGE agent will continue to operate, but will not be able to send SNMP trap messages.

Action:

None. Information only.

failed to create timer event**Valid on Windows systems only****Reason:**

The CA eHealth SystemEDGE agent uses a timer construct to perform periodic processing. This message indicates that the agent failed to allocate such a resource. Consequently, the agent's self-monitoring capabilities may not function properly.

Action:

None. Information only.

failed to create timer event, filename**Valid on Windows systems only****Reason:**

The CA eHealth SystemEDGE agent failed to create a Windows timer event necessary for subagent initialization and operation. The problem is most likely due to an error on the underlying Windows system.

Action:

Reboot the Windows system.

failed to create trap pdu**Reason:**

The CA eHealth SystemEDGE agent failed to create an SNMP trap message for transmission to SNMP management systems. This error usually occurs when the underlying system is low on memory.

The CA eHealth SystemEDGE agent will not send the particular SNMP trap message, but it will continue to attempt to send subsequent SNMP trap messages.

Action:

None. Information only.

failed to get dkscinfo

Reason:

The CA eHealth SystemEDGE agent could not get a disk statistics structure out of the UNIX kernel. Consequently, disk statistics may not be supported for some or all disks.

Action:

None. Information only.

failed to get domain name

Reason:

The CA eHealth SystemEDGE agent could not determine the system's DNS domain name. This error usually occurs when the underlying system's DNS domain name is not configured. This error has little effect on the CA eHealth SystemEDGE agent's overall operation.

Action:

None. Information only.

failed to get service handle

Valid on Windows systems only

Reason:

The CA eHealth SystemEDGE agent could not obtain a Windows service handle. The service information is obtained through an operating-system specific resource termed a handle.

This message indicates that there was a problem obtaining a particular service's handle. The CA eHealth SystemEDGE agent will continue operation but may not be able to provide all information about a particular Windows service.

Action:

None. Information only.

failed to open /dev/netman**Valid on HP-UX****Reason:**

The CA eHealth SystemEDGE agent could not open the device /dev/netman, which enables the agent to support MIB-II. Running the CA eHealth SystemEDGE agent as a user other than the root user can cause this problem.

Action:

Verify that the /dev/netman file exists and that the CA eHealth SystemEDGE agent has permission to read it.

failed to open /system**Valid on HP-UX****Reason:**

The CA eHealth SystemEDGE agent failed to open the /system directory for inspection of locally installed software packages and patches. This problem can occur if you are running the CA eHealth SystemEDGE agent as a user other than the root user.

If you do not correct this error, the CA eHealth SystemEDGE agent will continue to operate but will not be able to support the Host Resources Installed Software table.

Action:

Check that the /system directory exists and that the CA eHealth SystemEDGE agent has permission to read it.

failed to open /var/adm/sw/products

Valid on HP-UX systems

Reason:

The CA eHealth SystemEDGE agent failed to open the /var/adm/sw/products directory for inspection of locally installed software packages. This problem can occur if you are running the CA eHealth SystemEDGE agent as a user other than the root user.

If you do not correct this error, the CA eHealth SystemEDGE agent will continue to operate but will not be able to support the Host Resources Installed Software table.

Action:

Verify that the /var/adm/sw/products directory exists and that the CA eHealth SystemEDGE agent has permission to read it.

failed to open /var/sadm/patch

Valid on Solaris 2.x

Reason:

The CA eHealth SystemEDGE agent failed to open the /var/sadm/patch directory for inspection of locally installed patches. This problem can occur if you are running the CA eHealth SystemEDGE agent as a user other than the root user.

If you do not correct this error, the CA eHealth SystemEDGE agent will continue to operate but will not be able to determine which patches have been installed on the underlying system.

Action:

Check that the /var/sadm/patch directory exists and that the CA eHealth SystemEDGE agent has permission to read it.

failed to open /var/sadm/pkg**Valid on Solaris 2.x****Reason:**

The CA eHealth SystemEDGE agent failed to open the /var/sadm/pkg directory for inspection of locally installed software packages. This problem can occur if you are running the CA eHealth SystemEDGE agent as a user other than the root user.

If you do not correct this error, the CA eHealth SystemEDGE agent will continue to operate but will not be able to support the Host Resources Installed Software table.

Action:

Check that the /var/sadm/pkg directory exists and that the CA eHealth SystemEDGE agent has permission to read it.

failed to open config file *filename***Reason:**

The CA eHealth SystemEDGE agent failed to open the specified configuration file *filename*. The CA eHealth SystemEDGE agent will not operate until you fix this problem.

Action:

Verify that the *filename* file exists and that it is readable by the CA eHealth SystemEDGE agent.

failed to open ioconfig**Reason:**

Upon startup, the CA eHealth SystemEDGE agent attempts to discover the devices in the underlying system using available operating system support..

This error message indicates that the CA eHealth SystemEDGE agent could not properly read the HP-UX file /etc/ioconfig and will, therefore, attempt to discover devices without the operating system's help.

Action:

None. Information only.

failed to open ip for mib2

Valid on Solaris 2.x

Reason:

The CA eHealth SystemEDGE agent needs access to the /dev/ip file to properly support MIB-II. This problem can occur if you are running the CA eHealth SystemEDGE agent as a user other than the root user.

Action:

Check that the /dev/ip file exists and that it is readable by the CA eHealth SystemEDGE agent.

failed to open kmem

Valid on UNIX

Reason:

The CA eHealth SystemEDGE agent reads the kmem file to retrieve certain kernel parameters and statistics. This error message indicates that the agent could not open the /dev/kmem file, perhaps because of permissions problems.

Action:

Verify that the agent has read access to /dev/kmem.

failed to open mnttab file

Valid on UNIX

Reason:

The CA eHealth SystemEDGE agent could not open the system file that indicates which file systems were mounted and are accessible to users. Although the CA eHealth SystemEDGE agent will continue to operate, it may be unable to answer SNMP queries regarding file systems, disks, and partitions.

Action:

Verify that the system-specific mounted file system file exists and that it is readable by the CA eHealth SystemEDGE agent.

failed to open/create mon file**Reason:**

The CA eHealth SystemEDGE agent could not open or create a monitor configuration file. When the CA eHealth SystemEDGE agent's in-memory monitor table changes, it is written out, in ASCII format, to the monitor table configuration file.

This message indicates that the operation failed. The CA eHealth SystemEDGE agent will continue to operate and will continue to monitor MIB objects based on its in-memory monitor table. However, that contents of the in-memory monitor table may be lost if the agent is restarted before it can properly save the data.

Action:

None. Information only.

failed to open openprom device**Valid on Sun****Reason:**

The CA eHealth SystemEDGE agent obtains some configuration information from the openprom facility. This error message indicates that the agent could not open that file; consequently, some configuration information may not be supported.

Action:

None. Information only.

failed to parse config file**Reason:**

The CA eHealth SystemEDGE agent could not parse the configuration file. The CA eHealth SystemEDGE agent will not operate if it cannot find a valid configuration file.

Action:

Verify that the configuration file exists, is in either the default location or the location that you specified on the command line, and is readable.

failed to push ARP for mib2

Valid on Solaris 2.x

Reason:

The CA eHealth SystemEDGE agent needs access to the /dev/ip file and the arp Streams module to properly support MIB-II. This problem can occur if you are running the CA eHealth SystemEDGE agent as a user other than the root user.

Action:

Verify that the /dev/ip file exists and that it is readable by the CA eHealth SystemEDGE agent.

failed to push TCP for mib2

Valid on Solaris 2.x

Reason:

The CA eHealth SystemEDGE agent needs access to the /dev/ip file and the tcp Streams module to properly support MIB-II. This problem can occur if you are running the CA eHealth SystemEDGE agent as a user other than the root user.

Action:

Verify that the /dev/ip file exists and that it is readable by the CA eHealth SystemEDGE agent.

failed to push UDP for mib2

Valid on Solaris 2.x

Reason:

The CA eHealth SystemEDGE agent needs access to the /dev/ip file and the udp Streams module to properly support MIB-II. This problem can occur if you are running the CA eHealth SystemEDGE agent as a user other than the root user.

Action:

Verify that the /dev/ip file exists and that it is readable by the CA eHealth SystemEDGE agent.

failed to read ioconfig magic

Reason:

Upon startup, the CA eHealth SystemEDGE agent attempts to discover the devices in the underlying system using whatever operating system support is available.

This error message indicates the CA eHealth SystemEDGE agent could not properly read the HP-UX /etc/ioconfig file and will, therefore, attempt to discover devices without the operating system's help.

Action:

None. Information only.

failed to read monitor file

Reason:

The CA eHealth SystemEDGE agent failed to open either the default monitor file or the file that you specified on the UNIX command line.

The default monitor file for UNIX is /etc/sysedge.mon; for Windows, it is %SystemRoot%\system32\sysedge.mon.

Action:

None. Information only.

failed to reload utmp cache

Reason:

The CA eHealth SystemEDGE agent failed to reload its internal table of users who are currently logged in to the system. The CA eHealth SystemEDGE agent will continue function normally, but may be unable to answer SNMP queries of Who Table objects.

Action:

None. Information only.

failed to rename mon file

Reason:

The CA eHealth SystemEDGE agent periodically writes its in-memory monitor table to the sysedge.mon file. Before doing so, the CA eHealth SystemEDGE agent renames the current sysedge.mon to sysedge.old (in the same directory as the current file).

This error message indicates that the rename operation failed. The CA eHealth SystemEDGE agent will continue to function normally, but the contents of the old monitor table will not be recoverable.

Action:

None. Information only.

failed to send COLDSTART trap

Reason:

The CA eHealth SystemEDGE agent failed to send a MIB-II defined Cold Start trap message to its SNMP management systems. This error usually results from underlying operating system problems.

The CA eHealth SystemEDGE agent will continue to operate normally and will attempt to continue to send SNMP trap messages as necessary.

Action:

None. Information only.

fork failed for extension command

Reason:

The CA eHealth SystemEDGE agent failed to fork itself to run the extension command. Fork failures occur if the system has insufficient resources to create new processes. This error indicates that no extension command ran.

Action:

None. Information only.

fork failed for logmonitor action

Reason:

The CA eHealth SystemEDGE agent failed to fork itself to run the Log Monitor action. Fork failures occur if the system has insufficient resources to create new processes.

This error indicates that no Log Monitor action ran, but the Log Monitor table row is still active and will continue to attempt action commands when the table row evaluates to True.

Action:

None. Information only.

fork failed for monitor action

Reason:

The CA eHealth SystemEDGE agent failed to fork itself to run the monitor action. Fork failures occur if the system has insufficient resources to create new processes.

This error indicates that no monitor action ran, but the monitor table row is still active and will continue to attempt action commands when the table row evaluates to True.

Action:

None. Information only.

FPE signal caught

Reason:

The CA eHealth SystemEDGE agent caught a floating-point exception error, probably because of a divide-by-zero error.

Action:

For assistance, contact Technical Support at <http://ca.com/support>.

ID is processname,threadname

Valid on Windows systems only

Reason:

The CA eHealth SystemEDGE agent reports its process *processname* and thread *threadname* identifiers in its log file. This message is for debugging purposes only and can be ignored.

Action:

None. Information only.

identical threads IDs

Valid on Windows systems only

Reason:

The CA eHealth SystemEDGE agent has discovered two separate, distinct threads with the same thread identifier. Although this condition is technically impossible, it can still occur.

This message indicates that the CA eHealth SystemEDGE agent has discovered this situation and has properly accommodated it.

Action:

None. Information only.

invalid extension variable access mode

Reason:

The extension statement in the CA eHealth SystemEDGE agent configuration file contains an invalid access mode. Valid access modes are read-only or read-write, indicating whether an extension variable can be only read or read or written.

The agent will ignore the offending extension statement and will parse the remainder of the configuration file.

Action:

None. Information only.

For more information about extending the CA eHealth SystemEDGE agent, see the chapter "Adding Custom MIB Objects."

invalid extension variable type *typename***Reason:**

The extension statement in the CA eHealth SystemEDGE agent configuration file contained an invalid SNMP type. Valid extension variable SNMP types include the following: integer, counter, gauge, octetstring, timeticks, objectid, and ipaddress.

The agent will ignore the offending extension statement and will parse the remainder of the configuration file.

Action:

None. Information only.

For more information about extending the CA eHealth SystemEDGE agent, see the chapter "Adding Custom MIB Objects."

invalid history description**Reason:**

An emphistory configuration file statement contained an invalid description field. The agent will not create a history table entry and will ignore the statement. The agent, will, however, continue to parse the remainder of the configuration file.

Action:

Verify that the description is delineated by single quotation marks.

invalid history object type**Reason:**

An emphistory configuration file statement contained an object identifier whose base SNMP type is not an integer, counter, or gauge.

The CA eHealth SystemEDGE agent will not create a history table entry and will ignore the statement. The agent will, however, continue to parse the remainder of the configuration file.

Action:

None. Information only.

Invalid monitor table index

Reason:

An invalid Monitor table index was specified either in a configuration file command or through SNMP row creation to the monitor table. Rows 1 through 10 are reserved by the CA eHealth SystemEDGE agent for internal use.

Action:

Verify that the monitor table indexes are greater than 10.

invalid monprocess regular expression

Reason:

A monprocess configuration file statement contained an invalid regular expression. The agent will not create the Monitor table entry and will ignore the statement. The CA eHealth SystemEDGE agent, however, will continue to parse the remainder of the configuration file.

Action:

None. Information only.

invalid NT event log name

Reason:

The Windows event-log name that was supplied through the configuration file or through the command-line utility was incorrect.

Action:

None. Information only.

invalid NT event type

Reason:

The Windows event type that was supplied through the configuration file or through the command-line utility was incorrect.

Action:

None. Information only.

invalid number history buckets**Reason:**

An emphistory configuration file statement contained an invalid number of history buckets. The CA eHealth SystemEDGE agent will not create a history table entry and will ignore the statement. The agent will, however, continue to parse the remainder of the configuration file.

Action:

None. Information only.

invalid SNMP variable type *typename***Reason:**

The SNMP variable type *typename*, as specified in the agent's configuration file for registry extension objects, was incorrect.

Action:

None. Information only.

For more information about supported SNMP types, see the chapter "Adding Custom MIB Objects."

license file */etc/sysedge.lic* not found**Reason:**

The CA eHealth SystemEDGE agent could not find the default UNIX license file, */etc/sysedge.lic*. Without a proper license, the CA eHealth SystemEDGE agent will not continue to operate.

Action:

None. Information only.

license file not found**Reason:**

The CA eHealth SystemEDGE agent could not find a license file that was provided on the command line. Without a proper license, the CA eHealth SystemEDGE agent will not continue to operate.

Action:

None. Information only.

lock of mnttab lock failed

Reason:

The CA eHealth SystemEDGE agent failed to lock the UNIX mounted-device file. Consequently, information about mounted file systems may not be supported.

Action:

None. Information only.

log file is not regular

Reason:

The CA eHealth SystemEDGE agent was instructed, either through configuration file statements or remotely through SNMP, to monitor an irregular ASCII file. Consequently, the CA eHealth SystemEDGE agent will not monitor the log file for the corresponding regular expression.

Action:

None. Information only.

log filename too long

Reason:

A log-file name that was specified in the CA eHealth SystemEDGE agent configuration file exceeds the maximum file name length of 256 characters. Consequently, the CA eHealth SystemEDGE agent will not monitor the log file for the corresponding regular expression.

Action:

None. Information only.

logmon entry *entryname* re-initialized

Reason:

The Log Monitor entry *entryname* was automatically reinitialized by the CA eHealth SystemEDGE agent.

Action:

None. Information only.

logmon regcomp failed, *entryname***Reason:**

An invalid regular expression was configured for log file monitoring, either remotely through SNMP or through logmon configuration file statements. The agent will not add this Log Monitor table entry, but it will continue to monitor other log files for their corresponding regular expressions.

Action:

None. Information only.

logmon trap entry not ready Index:*entryname***Reason:**

The CA eHealth SystemEDGE agent has sent a Log Monitor trap message that indicates that entry *entryname* is notReady. The agent will no longer perform log monitoring on behalf of Log Monitor table entry *entryname*, but will continue to perform log monitoring on all other entries. A Log Monitor table entry can become notReady when an error occurs while reading its log file.

Action:

None. Information only.

logmon trap Index:*entryname***Reason:**

The CA eHealth SystemEDGE agent has sent a Log Monitor trap message that indicates that it has detected a log file match for entry *entryname*. The agent will continue to monitor this log file and will send additional trap messages when it finds new matches.

Action:

None. Information only.

logmonitor action execution failed**Reason:**

The CA eHealth SystemEDGE agent failed to run a Log Monitor action command. The agent will continue to monitor log files and will attempt to perform action commands when it finds matches.

Action:

None. Information only.

malloc of trap contents failed

Reason:

The CA eHealth SystemEDGE agent failed to send a trap message because it could not acquire the necessary memory. This error is most likely caused by a lack of memory on the underlying system. The CA eHealth SystemEDGE agent will continue to attempt to send trap messages.

Action:

None. Information only.

monitor action execution failed

Reason:

The CA eHealth SystemEDGE agent failed to run a Monitor table action command. The agent will continue to monitor entries and will attempt to perform action commands when monitor table expressions evaluate to True.

Action:

None. Information only.

monitor entry *entryname* not ready

Reason:

The CA eHealth SystemEDGE agent has sent a Monitor table trap message that indicates that entry *entryname* is notReady. The agent will no longer evaluate Monitor table entry *entryname*, but will continue to perform monitoring on all other entries. A Monitor table entry can become notReady when an error occurs while accessing a MIB variable.

Action:

None. Information only.

monitor trap Index:*entryname*

Reason:

The CA eHealth SystemEDGE agent has sent a Monitor trap message that indicates that Monitor table row *entryname* has evaluated to True. If the entry contained an action command, the agent ran it.

The agent will continue to monitor this entry and will send additional trap messages when the entry reevaluates to True.

Action:

None. Information only.

monprocess requires regular expression**Reason:**

A monprocess statement in the CA eHealth SystemEDGE agent configuration file did not contain a regular expression. A monprocess statement must contain a Monitor table index number that identifies which monitor table entry to use and a regular expression that identifies which process to monitor. The CA eHealth SystemEDGE agent will ignore the error and will continue to parse the configuration file.

Action:

None. Information only.

nlist of /unix failed**Reason:**

The CA eHealth SystemEDGE agent could not obtain the kernel name list (or symbol table) for the UNIX operating system running on the underlying system.

The agent will continue to operate but will be unable to report many kernel performance statistics and configuration parameters. This error occurs when an alternative kernel is booted.

Action:

Reboot with the /unix kernel file.

Example: kernel name list

/unix.boot for testing purposes.

no extension variable found for *indexname***Reason:**

The CA eHealth SystemEDGE agent could not find an extension variable that corresponds to index *indexname*. Extension variables are numbered from 1 through 32. This error message may occur if you attempt to add (through the configuration file) an extension variable whose number falls outside this range.

The CA eHealth SystemEDGE agent will ignore the configuration file statement and will continue to parse the configuration file.

Action:

None. Information only.

For more information about extending the CA eHealth SystemEDGE agent, see the chapter "Adding Custom MIB Objects."

non-existent object to track history of

Reason:

An emphistory configuration file statement contains an invalid object identifier; the object identifier does not exist within the CA eHealth SystemEDGE agent's MIB. The agent will not create a History table entry and will ignore the statement. The agent will, however, continue to parse the remainder of the configuration file.

Action:

None. Information only.

no process matching expression

Reason:

The CA eHealth SystemEDGE agent could not match a configuration file monprocess regular expression to a corresponding process name. Consequently, it will ignore the monprocess statement in the configuration file.

Action:

None. Information only.

not querying serial port status

Reason:

The CA eHealth SystemEDGE agent will not query the status of serial ports in response to queries of the variable hrDeviceStatus. On some older UNIX systems, serial port status queries can interfere with serial-port based applications. The CA eHealth SystemEDGE agent configuration file parameter, no_serial_status, enables this option.

Action:

None. Information only.

not sending authen failure traps

Reason:

The CA eHealth SystemEDGE agent will not send MIB-II authenFailure trap messages in response to SNMP queries using invalid community strings. The CA eHealth SystemEDGE agent configuration file parameter, no_authen_traps, enables this option.

Action:

None. Information only.

not stat'ing disks devices**Reason:**

The CA eHealth SystemEDGE agent will not check the status of disk devices according to the configuration file directive.

Action:

None. Information only.

not stat'ng floppy devices**Reason:**

The CA eHealth SystemEDGE agent will not check the status of floppy disk devices according to the configuration file directive.

Action:

None. Information only.

not stating NFS filesystems**Valid on UNIX****Reason:**

The CA eHealth SystemEDGE agent will not monitor or report statistics for NFS-mounted filesystems. Attempts to ascertain the status of NFS-mounted filesystems whose file servers are unavailable or down can indefinitely block the CA eHealth SystemEDGE agent.

Unfortunately, programmatic options to prevent this are not possible.

Action:

Enable this option using the CA eHealth SystemEDGE agent configuration file parameter, `no_stat_nfs_filesystems`.

not supporting actions**Reason:**

The CA eHealth SystemEDGE agent is not supporting actions according to the configuration file directive.

Action:

None. Information only.

not supporting remoteShell group

Reason:

The CA eHealth SystemEDGE agent will not support SNMP queries (Gets and Sets) to the remoteShell group because local system security policies prohibit this functionality.

Action:

Enable this option using the CA eHealth SystemEDGE agent configuration file parameter, `no_remoteshell_group`.

not supporting user/group tables

Reason:

The CA eHealth SystemEDGE agent will not support SNMP queries to the User and Group tables because local system security policies prohibit the dissemination of valid user and group information.

Action:

Enable this option using the CA eHealth SystemEDGE agent configuration file parameter, `no_usergroup_table`.

not supporting who table

Reason:

The CA eHealth SystemEDGE agent will not support SNMP queries to the Who table because local system security policies prohibit the dissemination of currently logged in users.

Action:

Enable this option using the CA eHealth SystemEDGE agent configuration file parameter, `no_who_table`.

nteventmon entry *entryname* not ready**Reason:**

The CA eHealth SystemEDGE agent has sent an NT Event Monitor trap message that indicates that entry *entryname* is notReady. The CA eHealth SystemEDGE agent will no longer perform Windows event monitoring on behalf of the NT Event Monitor table entry *entryname*, but will continue to perform Windows event monitoring on all other entries.

An NT Event Monitor table entry can become notReady when the CA eHealth SystemEDGE agent cannot read the corresponding Windows event log file. This error occurs only when there are errors on the underlying Windows system.

Action:

None. Information only.

odm_initialize failed**Valid on AIX****Reason:**

The CA eHealth SystemEDGE agent uses an odm library for obtaining hardware and device information. This message indicates that initialization of that library failed. Consequently, it may not support device information.

Action:

None. Information only.

openProcess failed on pid**Valid on Windows systems only****Reason:**

The CA eHealth SystemEDGE agent failed to open a process for statistics retrieval because the process may not be in existence anymore. The agent will continue to operate normally and will continue to support the process table.

Action:

None. Information only.

openProcessToken failed on pid

Valid on Windows systems only

Reason:

The CA eHealth SystemEDGE agent failed to open a process for statistics retrieval because the process may not be in existence anymore. The agent will continue to operate normally and will continue to support the process table.

Action:

None. Information only.

openprom device not supported

Valid on UNIX

Reason:

Upon startup the CA eHealth SystemEDGE agent attempts to discover the devices in the underlying system using whatever operating system support is available.

This error messages indicates that the underlying system does not support the openprom device, which is used to determine system configuration and hardware information. Consequently, the CA eHealth SystemEDGE agent will attempt to determine the system's configuration without the operating system's help.

Action:

None. Information only.

perfDiskObjects *objectname* != Num_Disks

Reason:

The CA eHealth SystemEDGE agent found a number of disk objects that is not equivalent to the number of disks it found through other mechanisms.

Action:

Ignore this message.

realloc of mnt cache failed!**Reason:**

The CA eHealth SystemEDGE agent failed to reallocate space for its internal cache of mounted file systems. This error usually occurs when the system is extremely low on memory. The agent will continue to operate but may be unable to report file system statistics until more memory is available.

Action:

None. Information only.

recvfrom failed**Reason:**

The CA eHealth SystemEDGE agent encountered an error when reading an SNMP request from the underlying UDP transport.

Action:

None. Information only.

reload_process_table: open /proc failed**Valid on UNIX****Reason:**

The CA eHealth SystemEDGE agent failed to open the /proc directory on; consequently, it cannot support the Process Monitor table.

Action:

None. Information only.

reload_process_table: proc ioctl failed**Reason:**

The CA eHealth SystemEDGE agent failed to perform an I/O control operation (ioctl) on a particular process located in the /proc directory. The agent, therefore, cannot support process information for this particular process, but will perform nominally for other processes.

Action:

None. Information only.

root device ptr failed, no openprom

Valid on UNIX

Reason:

The CA eHealth SystemEDGE agent attempts to discover the devices in the underlying system using whatever operating system support is available.

This error messages indicates that the underlying system does not support the openprom device, which is used to determine system configuration and hardware information. Consequently, the agent will attempt to determine the system's configuration without the operating system's help.

Action:

None. Information only.

sent SIGKILL to process *processname*

Valid on UNIX

Reason:

The CA eHealth SystemEDGE agent sent a KILL signal to a process whose PID is *processname*. This function is accomplished through SNMP Sets to the processKill variable in the Process Monitor table or to the hrRunStatus variable in the Host Resources hrSWRunTable.

Action:

None. Information only.

sent signal *signalname* to process *processname*

Valid on UNIX

Reason:

The CA eHealth SystemEDGE agent sent a signal to a process whose PID is *processname*. This function is accomplished through SNMP Sets to the processKill variable in the Process Monitor table or to the hrRunStatus variable within the Host Resources hrSWRunTable. Any valid UNIX signal can be sent to a process.

Action:

None. Information only.

setLogmonEntry: invalid set (logfile), row of status**Reason:**

To set Log Monitor table rows that have a status of notInService, you must perform SNMP Set operations made to those rows. This error indicates that the Set operation it references failed.

Action:

None. Information only.

setMonEntry: bad size for OID val**Reason:**

An SNMP Set operation to the Log Monitor table contained an incorrect length for a particular object-identifier value. The Set operation referenced by this message failed.

Action:

None. Information only.

setMonEntry: invalid oper**Reason:**

An SNMP Set operation to the Log Monitor table contained an improper operator type. The Set operation referenced by this message failed.

Action:

None. Information only.

setMonEntry: invalid oper type**Reason:**

An SNMP Set operation to the Log Monitor table contained an improper operator type. The Set operation referenced by this message failed.

Action:

None. Information only.

setMonEntry: invalid stype

Reason:

An SNMP Set operation to the Log Monitor table contained an improper sample type. The Set operation referenced by this message failed.

Action:

None. Information only.

setMonEntry: invalid type for OID

Reason:

An SNMP Set operation to the Log Monitor table contained an improper object identifier type. The Set operation referenced by this message failed.

Action:

None. Information only.

setMonEntry: invalid type for val

Reason:

An SNMP Set operation to the Log Monitor table contained an improper value type. The Set operation referenced by this message failed.

Action:

None. Information only.

setMonEntry: oid type invalid

Reason:

An SNMP Set operation to the Log Monitor table contained an improper object-identifier type. The Set operation referenced by this message failed.

Action:

None. Information only.

setMonEntry: stype type invalid**Reason:**

An SNMP Set operation to the Log Monitor table contained an improper sample type. The Set operation referenced by this message failed.

Action:

None. Information only.

stat logfilename failed**Reason:**

The CA eHealth SystemEDGE agent could not determine file information for a particular log file that it is monitoring for a regular expression. Consequently, it sets the status of the corresponding Log Monitor table row to notReady.

Action:

None. Information only.

stat of extension command file *variablename* failed**Reason:**

The CA eHealth SystemEDGE agent could not determine file information for the extension command that corresponds to extension variable *variablename*. Consequently, the agent will not support extension variable *variablename*. The agent will support all other valid extension variables.

Action:

None. Information only.

For more information about extending the CA eHealth SystemEDGE agent, see the chapter "Adding Custom MIB Objects."

stat of logmon action *actionname* failed**Reason:**

The CA eHealth SystemEDGE agent could not determine file information about an action command file used with a Log Monitor configuration file statement. Consequently, it sets the status of the corresponding Log Monitor table row to notReady.

Action:

None. Information only.

stat of monfilesys action *actionname* failed

Reason:

The CA eHealth SystemEDGE agent could not determine file information about an action command file used with a monfilesys configuration file statement. Consequently, it sets the status of the corresponding Monitor table row to notReady.

Action:

None. Information only.

stat of monprocess action *actionname* failed

Reason:

The CA eHealth SystemEDGE agent could not determine file information about an action command file used with a monprocess configuration file statement. Consequently, it sets the status of the corresponding Process Monitor table row to notReady.

Action:

None. Information only.

stat of nteventmon action *actionname* failed

Reason:

The CA eHealth SystemEDGE agent could not determine file information about an action command file used with a NT Event Monitor table configuration file statement. Consequently, it sets the status of the corresponding NT Event Monitor table row to notReady.

Action:

None. Information only.

sysedge using port *portname*, config file *filename*

Valid on UNIX

Reason:

Upon startup the CA eHealth SystemEDGE agent report which UDP port and configuration file they are using. This message is informational only and does not represent an error condition.

Action:

None. Information only.

system call ret error *errornumber***Reason:**

The CA eHealth SystemEDGE agent could not run a command that was specified as part of the remoteShell group functionality. This error can occur when the remoteShell function is invalid, or if the underlying system cannot create a subprocess.

Action:

None. Information only.

This agent binary is compiled for *operatingsystemname*, not *operatingsystemname***Reason:**

The CA eHealth SystemEDGE agent is specific to the version of the operating system on which it runs; therefore it must often be compiled specifically for each operating system. This error message indicates that a version of the CA eHealth SystemEDGE agent for one version of the operating system was run on a version that is not compatible with the one for which it was compiled.

Action:

None. Information only.

Example: Solaris 2.x

You must use separate CA eHealth SystemEDGE agent binaries for versions 2.5.x, and 2.6, 2.7 (32-bit) and 2.7 (64-bit).

For more information about binaries for each operating system, see the chapter "Troubleshooting and Usage Suggestions."

timeGetDevCaps failed, exiting**Valid on Windows systems only****Reason:**

The CA eHealth SystemEDGE agent could not get the system's timer resolution capabilities necessary for internal operation and self-monitoring of MIB objects. Consequently, the agent will not operate. This problem is most likely due to an error on the underlying Windows system.

Action:

Reboot the Windows system.

timeKillEvent failed

Valid on Windows systems only

Reason:

The CA eHealth SystemEDGE agent delays its initialization to avoid potential race conditions that can be created by the order in which the registry and services are initialized.

This error message indicates that the CA eHealth SystemEDGE agent could not stop its internal timer event from firing. The problem most likely is due to an error on the underlying Windows system.

Action:

Reboot the Windows system to resolve this problem.

trap ipaddress/hostname *ipaddress/hostname* invalid

Valid on UNIX

Reason:

The sysedge.cf configuration file indicates to which hosts the CA eHealth SystemEDGE agent should send SNMP trap messages. This error indicates that one of the trap statements in the sysedge.cf configuration file specifies an incorrect hostname or IP address.

The agent will ignore the offending trap statement, but will parse the rest of the configuration file.

Action:

None. Information only.

turning off process table support

Reason:

The CA eHealth SystemEDGE agent is disabling support of the process table according to the configuration file directive.

Action:

None. Information only.

turning off sets to Empire process table

Reason:

The CA eHealth SystemEDGE agent is disabling support of SNMP Sets to the process table according to the configuration file directive.

Action:

None. Information only.

two processes with PID *processname*

Reason:

The CA eHealth SystemEDGE agent has discovered two separate, distinct processes with the same process identifier. Although this condition is technically impossible, the CA eHealth SystemEDGE agent still guards against it. This message indicates that the agent has discovered this situation and has properly accommodated it.

Action:

None. Information only.

two software packages with same index

Reason:

The CA eHealth SystemEDGE agent has discovered two software packages with the same index value. This condition can occur when local users have changed files in the system's software installation area, or if those files have been damaged.

This message indicates that the CA eHealth SystemEDGE agent has discovered the situation and has properly accommodated it.

Action:

For assistance, contact Technical Support at <http://ca.com/support>.

unable to open monitor file

Reason:

The CA eHealth SystemEDGE agent could not open the monitor file that was specified as a command-line argument or could not open the default monitor file.

Action:

Verify that the monitor file that is specified as part of the UNIX command line specifies a valid monitor file or that the default monitor file is in the proper location. The default monitor file, sysedge.mon, is in the UNIX /dir directory or in the Windows %SystemRoot%\system32\ directory.

unable to process acl for community *communityname*

Reason:

This message indicates that the CA eHealth SystemEDGE agent could not parse an access control list specification as part of a community declaration in the sysedge.cf configuration file. The agent will ignore the offending access control list but will continue to support the corresponding community string declaration.

Action:

None. Information only.

unknown HP CPU type

Reason:

This message indicates that the underlying HP-UX system contains a processor type unknown to the CA eHealth SystemEDGE agent.

Action:

For assistance, contact Technical Support at <http://ca.com/support>.

unknown NT event log name

Reason:

This message indicates that an invalid Windows event log name was specified, either through SNMP or through the nteventmon command-line tool.

Valid Windows event log names are application, security, or system. The agent will not create the NT Event Monitor table entry.

Action:

None. Information only.

unknown NT event type**Reason:**

This message indicates that an invalid Windows event type was specified, either through SNMP or through the command-line tool nteventmon.

Valid Windows event types are error, warning, information, success, fail, or all. The NT event monitor entry will not be created.

Action:

None. Information only.

unknown service start type**Reason:**

This message indicates that the CA eHealth SystemEDGE agent discovered a Windows service start type that it did not understand.

Action:

For assistance, contact Technical Support at <http://ca.com/support>.

unknown system type**Reason:**

This message indicates that the CA eHealth SystemEDGE agent could not determine if the underlying Windows system is configured as a Windows Server or a Windows Workstation.

Action:

For assistance, contact Technical Support at <http://ca.com/support>.

username *username* not found, all subprograms will be disabled**Reason:**

This message indicates that the specified username for running subprograms does not exist in the sysedge.cf configuration file. Consequently, all subprogram execution by CA eHealth SystemEDGE agent will be disabled.

Action:

None. Information only.

Using config file

Reason:

This message indicates that the CA eHealth SystemEDGE agent for Windows is using the specified configuration file. This message is informational only and does not indicate that an error has occurred.

Action:

None. Information only.

Using monitor file

Reason:

This message indicates that the CA eHealth SystemEDGE agent for Windows is using the specified monitor configuration file. This message is informational only and does not indicate that an error has occurred.

Action:

None. Information only.

using old config file

Reason:

This message indicates that the CA eHealth SystemEDGE agent is using a configuration file from a release earlier than Release 3.0. This message is informational in nature.

Action:

None. Information only.

using old monitor file *filename*; updates will be placed in *filename*

Reason:

This message indicates that the CA eHealth SystemEDGE agent is reading a monitor configuration file *filename* for a release that was earlier than Release 3.0 and that it will write updated, Release 4.0 monitor configuration files to the file *filename*.

Action:

None. Information only.

Command-line Utility Error Messages

This section lists the error messages that can occur when you are using the CA eHealth SystemEDGE command-line utilities. These error messages are generally printed to standard error or standard output. Messages are sorted alphabetically.

Common Error Messages

This section describes the common error messages that are applicable to all the following SNMP utilities. These error messages should be interpreted in conjunction with the `sysedge_utility.log` and `xtrapmon.log` (for `xtrapmon`) files in the `bin` subdirectory of the agent's installation.

The error messages described in this section are applicable to the following utilities:

- `edgemon`
- `edgewatch`
- `emphistory`
- `nteventmon`
- `sendtrap`
- `snmpget`
- `sysvariable`
- `walktree`
- `xtrapmon`

Additional error messages specific to a utility are described in the following section.

ERROR: Argument *oid* is not an oid

ERROR: Varbind could not be created

Reason:

An invalid Object Identifier (OID) is specified with the `-o` option.

Action:

Verify that the OID is a valid MIB OID. These error messages are applicable for `walktree`, `snmpget`, and `snmpset`.

ERROR: Cannot create a SNMP socket

Reason:

An error occurred while trying to create a socket to communicate with the agent.

Action:

Verify that a correct port number is specified.

ERROR: Cannot resolve address for *hostname*

Reason:

The SNMP utility could not resolve the machine name (provided with the -h option) to a valid IP address and therefore did not perform the requested action.

Action:

Verify that the machine name was provided has a valid IP address through tools like host or nslookup.

Error code set in packet - Bad variable value. Index: *index*

Reason:

The value of an OID or an index that is requested to be set is incorrect.

Action:

Verify that the value is specified correctly and that it is the correct type (integer, string, and so forth).

Error code set in packet - No such variable name. Index: *index*

Reason:

The OID that is specified does not exist in the agent.

Action:

Specify the correct OID that exists in the agent.

ERROR: Couldn't send the SNMP request**Reason:**

An error occurred while trying to communicate with the agent using the options that were specified.

Action:

Verify that the specified options are correct.

ERROR: Incorrect status**Reason:**

An invalid status is specified to the command.

Action:

Verify that the specified status is valid. This error message is applicable to edgemon, edgewatch, and emphistory utilities.

ERROR: Invalid number of retries specified: *retries*. Number of retries should be at least 1**Reason:**

The number of retries specified is invalid.

Action:

Specify an integer value greater than or equal to 1.

ERROR: Invalid port number *port***Reason:**

The port number that was provided using the -p option is invalid.

Action:

Verify that the port number is a positive integer value. A valid port number is 1 through 65535.

ERROR: Invalid security level: *secLevel*. Accepted values are: 1, 2, 3 or noAuthNoPriv. AuthNoPriv, AuthPriv

Reason:

The SNMPv3 security level version that was provided using the -s option is invalid.

Action:

Verify that the SNMPv3 security level is one of the following values: 1, 2, 3, noAuthNoPriv, AuthNoPriv, or AuthPriv.

ERROR: Invalid timeout value: *timeout*

Reason:

An invalid timeout value is specified using the -t option.

Action:

Verify that the timeout value is an integer number greater than zero.

ERROR: missing status

Reason:

The returned status value is not specified on the command line.

Action:

Specify a valid status and try again. This error message is applicable to edgemon, edgewatch, and emphistory utilities.

ERROR: OID is not specified

Reason:

The Object Identifier (OID) is not specified using the -o option.

Action:

Specify a valid OID. This error message is applicable for walktree, snmpget, and snmpset.

Error sending *SNMPversion* Get request

Error while waiting for *SNMPversion* Get response

Error while receiving *SNMPversion* Get response

Reason:

The command failed while trying to send or receive one of the request variables because of an error in the SNMP operation.

Action:

SNMPv1 and SNMPv2: Verify that the community string and target IP address (or system name) are correct.

SNMPv3: Verify that the SNMPv3 security parameters and the target IP address (or system name) are correct.

ERROR: SNMPv3 security user name is required

Reason:

This error occurs when the SNMP version is 3 and the SNMPv3 security name is not specified using the -u option.

Action:

Specify a valid SNMPv3 user name.

ERROR: unknown authentication protocol *authProtocol*. Accepted values are: MD5, SHA

Reason:

The authentication protocol that was specified using the -A option is invalid.

Action:

Verify that the authentication protocol is MD5 or SHA.

ERROR: unknown FIPS mode: *FIPS mode*. Acceptable values are: 0, 1, 2

Reason:

The FIPS mode that was provided using the -m option is invalid.

Action:

Verify that the FIPS mode is 0, 1, or 2.

ERROR: unknown privacy protocol *privacy protocol*. Accepted values are: DES, 3DES, AES

Reason:

The privacy (encryption) protocol that was provided using the -X option is invalid.

Action:

Verify that the privacy (encryption) protocol is DES, 3DES, or AES.

ERROR: unknown SNMP version: *version*. Accepted values are 1, 2c, 3

Reason:

The SNMP version that was provided using the -v option is invalid.

Action:

Verify that the SNMP version is 1, 2c, or 3.

get response - timeout

Reason:

The utility cannot get a response from the agent with the command line options that are specified.

Action:

Verify that the community name (for SNMPv1 or SNMPv2c), port number, and SNMPv3 user information matches the querying agent's configuration.

invalid index *index number*

Reason:

An invalid index value is specified to the command.

Action:

Specify a valid index value greater than or equal to zero. This error message is applicable to edgemon, edgewatch, and emphistory utilities.

SNMP error (-2) - could not retrieve SNMP packet from input

Reason:

An error occurred while retrieving the SNMP packet information from the agent.

Action:

Verify that the options that are specified on the command line are correct.

Timeout receiving SNMPversion Get response**Reason:**

The command timed out while trying to receive one of the request variables because of an error in the SNMP operation.

Action:

SNMPv1 or SNMPv2c: Verify that the community string and target IP address (or system name) are correct and increase the timeout value using the -t option.

SNMPv3: Verify that the SNMPV3 security parameters and the target IP address (or system name) are correct and increase the timeout value using the -t option.

WARNING: authentication set to MD5**Reason:**

If the SNMP version is 3 (-v 3), the security level is AuthPriv or AuthNoPriv, and if the authentication protocol is not specified using the -A option, the utility defaults to using the MD5 authentication protocol.

Action:

None. Information only.

WARNING: encryption set to DES**Reason:**

If the SNMP version is 3 (-v 3), the security level is AuthPriv, and if the encryption protocol is not specified using the -X option, the utility defaults to using the DES encryption protocol.

Action:

None. Information only.

WSAStartup failed**Valid on Windows systems only****Reason:**

The command line utility could not initialize the WinSock library and could not perform the requested action.

Action:

None. Information only.

edgemon Error Messages

This section describes error messages that may be generated by the edgemon utility.

Couldn't find filesystem *name* on *target host*

Reason:

The file system name that was provided to edgemon was not present or mounted on the remote system. Consequently, the monitor table entry was not created.

Action:

None. Information only.

edgewatch Error Messages

This section describes error messages that may be generated by the edgewatch utility.

Operating System **does not support procMonAttribute**

Reason:

The Process Monitor attribute that was passed to edgewatch is not supported by the underlying AIX operating system (on which the CA eHealth SystemEDGE agent is running). The agent therefore does not create a Process Monitor table.

Action:

None. Information only.

For more information about the Process Monitor attributes not supported by a particular platform, see the *CA eHealth SystemEDGE Release Notes*.

invalid/unknown system type

Reason:

The system type of the target CA eHealth SytemEDGE agent is unknown to edgwatch, so the utility did not perform the requested operation. This error can occur when an older version of the relevant command-line utility is used with a newer version of the agent ported to a system not originally supported at the time this utility was written.

Action:

For assistance, contact Technical Support at <http://ca.com/support>.

procAlive is the only attribute which can be applied to a process group

Reason:

The CA eHealth SystemEDGE agent can support process monitoring of groups of processes only if the particular attribute is procAlive. If another attribute was passed to edgwatch, no operation will be performed.

Action:

None. Information only.

unknown NT event log name

Reason:

An invalid Windows event log name was passed to edgwatch for NT Event Monitor table manipulation, so no operation was performed.

Action:

Verify that the event log name is valid on the target system.

unknown NT event type

Reason:

An invalid Windows event type was passed to edgwatch for NT Event Monitor table manipulation, so no operation was performed.

Action:

Verify that the event type name is valid on the target system.

sendtrap Error Messages

This section describes specific error messages that may be generated by the sendtrap utility. These error messages are in addition to the error messages described in the section Common Error Messages.

error: bad enterprise-specific trap subtype

Reason:

The enterprise-specific trap subtype that was specified to sendtrap was invalid, so an SNMP Trap PDU was not sent. Specific trap subtypes must be greater than or equal to zero.

Action:

None. Information only.

error: bad trap type

Reason:

The trap type that was specified to sendtrap was invalid, so an SNMP Trap PDU was not sent. Trap types must be greater than or equal to zero.

Action:

None. Information only.

error initializing FIPS mode

Reason:

sendtrap was specified to send a trap using the FIPS 140-2 mode of operation; however, the underlying FIPS (B-Safe Crypto) libraries failed to load or they are unavailable.

Action:

Verify that the FIPS libraries are available. FIPS libraries can be obtained by installing the CA eHealth Advanced Encryption package.

Incorrect input: unsupported Hex-Type

WARNING: Unknown attribute type

Reason:

Variable bindings can be passed to sendtrap and then sent in an SNMP Trap PDU. This message indicates that the variable type that was specified for one of the variable bindings was invalid.

Action:

None. Information only.

Input error: incomplete varbind

Reason:

The sendtrap utility encountered an error parsing a variable binding. The utility ignored the offending line and sent the SNMP Trap PDU.

Action:

None. Information only.

Send inform request failed

Reason:

sendtrap failed to send an inform Trap request using the -i option.

Action:

Verify the community strings, SNMPv3 user information, and port number, then try again.

Send trap failed

Reason:

The underlying library routine for sending the SNMP Trap PDU failed and no trap PDU was sent.

Action:

None. Information only.

sendtrap: incorrect arguments

Reason:

An incorrect number or invalid arguments were passed to the sendtrap utility.

Action:

Check the sendtrap utility usage using by entering sendtrap -? and then rerun the command.

sendtrap: The winsock.dll is not available

Valid on Windows systems only

Reason:

sendtrap could not find the WinSock library and could not perform the requested action.

Action:

None. Information only.

sendtrap: This requires at least version 1.1 of the winsock.dll

Valid on Windows systems only

Reason:

sendtrap could not find the WinSock library version 1.1 or newer.

Action:

None. Information only.

sendtrap: You must specify a subtrap type for trap type 6!

Reason:

An enterprise-specific trap type (6) was specified but a mandatory subtrap type is not specified.

Action:

Specify a subtrap type (an integer value greater than zero) for the enterprise-specific trap type.

warning: Only a maximum of 100 varbinds are supported**Reason:**

The sendtrap command using the older, deprecated usage (that is, versions older than CA eHealth SystemEDGE r4.3) can only send 100 varbinds (variable bindings) in a single trap PDU. The rest of the varbinds are ignored

Action:

None. Information only.

walktree Error Messages

This section describes specific error messages that may be generated by the walktree utility. These error messages are in addition to the error messages described in the section Common Error Messages.

ERROR: could not create output file**Reason:**

walktree could not open the specified output file due to an error opening or creating it. No walk operation was performed.

Action:

Make sure that the directory exists when the file will be created and is writable.

walktree: retries should be at least 1**Reason:**

walktree attempts to retry failed SNMP requests because SNMP packets may be dropped or lost. This error message indicates that the retry value that was specified on the command line was invalid.

Action:

None. Information only.

xtrapmon Error Messages

This section describes error messages that may be generated by the xtrapmon utility.

xtrapmon: cannot initialize. Unicode converting XLT files are missing in AWSCOMM DIR=directory

Reason:

xtrapmon could not open the specified SNMPv3 configuration file.

Action:

Verify that the SNMPv3 configuration file exists. If the SNMPv3 configuration file is specified using the -e option, make sure that it is specified with an absolute pathname.

xtrapmon: Could not open SNMPv3 configuration file: file

Reason:

xtrapmon could not open the specified SNMPv3 configuration file.

Action:

Verify that the SNMPv3 configuration file exists. If the SNMPv3 configuration file is specified using the -e option, make sure that it is specified with an absolute pathname.

xtrapmon: failed to alloc space for SNMPv3 config file

Reason:

xtrapmon could not allocate internal space for storing the file name of the SNMPv3 configuration file.

Action:

Verify that the system has enough memory resources and try again.

xtrapmon: failed to create log file

Reason:

xtrapmon could not open the specified output file because of an error opening or creating it.

Action:

Make sure that the directory exists where the file will be created and that it is writable.

xtrapmon: failed to get trap socket handle. Check the port number.

Reason:

xtrapmon failed to bind to port UDP/162 (the standard SNMP trap port) or a user-specified port. This problem usually occurs if another process is already bound to that port.

Action:

Terminate the other trap-receiving process or start xtrapmon on a different port.

xtrapmon: Only a maximum of 100 varbinds are supported.

Reason:

xtrapmon can only display 100 varbinds if the received trap message contains more than 100 varbinds.

Action:

None. Information only.

Appendix B: Using the syslog Facility

The CA eHealth SystemEDGE agent uses the UNIX syslog facility to log informational messages and error conditions that it may encounter during its operation. For more information about the syslog facility, refer to the following man pages for more information about the syslog facility:

- `syslog(3)`
- `syslog.conf(5)`
- `syslogd(8)`

You can also edit the `sysedge.cf` (default in `/etc`) file to instruct the CA eHealth SystemEDGE agent to log these messages in a different facility. For instructions, see [Configuring Alternative Syslog Facilities \(UNIX Only\)](#) in the chapter “Configuring the CA eHealth SystemEDGE Agent.”

This section contains the following topics:

[Logging syslog Messages](#) (see page 487)

[Creating a Log File for Daemon Messages](#) (see page 489)

Logging syslog Messages

syslog messages are typically logged to `/var/adm/messages` on Solaris, `/usr/adm/syslog/syslog.log` on HP-UX, and `/var/log/messages` on Linux. syslog file locations may differ depending on your system and how the syslog daemon is configured on your system. By default, the CA eHealth SystemEDGE agent daemon uses syslog to log messages with priority levels of *informational* through *emergency*. If you are running the agent in debug mode with the runtime command line option `-d`, syslog will also log messages of the debug priority level.

This guide does not provide a complete tutorial on the syslog utility. Instead, it describes how to configure the syslog daemon on your system to log messages from daemon processes like the CA eHealth SystemEDGE agent to a text file (`/var/log/daemon-log` for Sun SPARC systems, and `/usr/adm/daemon-log` for HP-UX systems).

Every message sent using syslog includes a facility code and a priority code that tells the message source and its severity, respectively.

The following describes the available syslog facility codes:

LOG_KERN

Kernel messages

LOG_USER

Random user-level messages

LOG_MAIL

Mail system

LOG_DAEMON

System daemons

LOG_AUTH

Security/authorization messages

LOG_SYSLOG

Messages generated internally by syslog

LOG_LPR

Line printer subsystem

LOG_NEWS

Network News subsystem

LOG_UUCP

UUCP subsystem

LOG_CRON

Cron/at subsystem

The following describes the syslog priority codes:

LOG_EMERG

System is unusable.

LOG_ALERT

Immediate action is required.

LOG_CRIT

Critical condition.

LOG_ERR

Error condition.

LOG_WARNING

Warning condition.

LOG_NOTICE

Normal but signification condition.

LOG_INFO

Informational.

LOG_DEBUG

Debug-level messages.

The typical syslog configuration logs messages to the text file `/var/adm/messages` on Solaris systems and to `/usr/adm/syslog/syslog.log` on HP-UX 11.x systems. If the message has a priority of `LOG_ERR` or higher, syslog also displays the message to the console.

Creating a Log File for Daemon Messages

Because the typical log file contains messages from many facilities in addition to daemon processes like the CA eHealth SystemEDGE agent, you may want to configure all messages from daemon processes to be logged to a separate daemon-log file. If your `/etc/syslog.conf` file does not already contain an entry for logging daemon processes separately, you can add an entry to cause syslog to log those messages to a separate daemon log file.

Create a Daemon Log File for Solaris SPARC Systems

To create a separate file for logging daemon messages for a Sun SPARC system, edit the `/etc/syslog.conf` file by entering following:

```
daemon /var/log/daemon-log
```

Create a Daemon Log File for HP-UX Systems

To create a separate file for logging daemon messages for HP-UX (version 11.x) systems, edit the `/etc/syslog.conf` file by entering following:

```
daemon.info /usr/adm/daemon-log
```

These changes take effect when the system is rebooted or when the syslog daemon reads its configuration file again. You can set it to do so by sending it an HUP signal.

Create a Daemon Log File for AIX Systems

By default, an AIX system will not do any logging. syslog is not configured on a default AIX operating system install.

The typical configuration file for syslog is /etc/syslog.conf. Configure syslog if you want to perform system messages logging. For example, /etc/syslog.conf may look like the following:

```
mail.debug                /var/adm/maillog
mail.none                 /var/adm/maillog
auth.notice               /var/adm/authlog
lpr.debug                 /var/adm/lpd-errs
kern.debug                /var/adm/messages
*.emerg;*.alert;*.crit;*.warning;*.err;*.notice;*.info /var/adm/messages
```

Restart syslogd after all of the updates are made to /etc/syslog.conf as follows:

```
refresh -s syslogd
```

Create a Daemon Log File for Linux Systems

By default, Linux's /etc/syslog.conf file is configured to log most of the messages in the file /var/log/messages. Here is an example:

```
*.info;mail.none;authpriv.none;cron.none /var/log/messages
```

To create a separate file for logging daemon messages for Linux systems, edit the /etc/syslog.conf file by entering following:

```
daemon.* /var/log/daemonlog
```

Changes to /etc/syslog.conf will not take effect until you restart syslog. Issue this command to do so:

```
/etc/init.d/syslog restart
```

Appendix C: Adding Self-Monitoring Entries to the sysedge.mon File

This appendix describes the format for the sysedge.mon file, which provides stable storage for the self-monitoring and history tables supported by CA eHealth SystemEDGE agent. In most cases, you do not need to edit the sysedge.mon file. Instead, you can configure these features of the CA eHealth SystemEDGE agent through the sysedge.cf configuration file or through one of the command-line utilities. If you must edit sysedge.mon, use this appendix to verify that you are using the correct file format.

This section contains the following topics:

- [CA eHealth SystemEDGE Table Backing Store](#) (see page 491)
- [Adding Monitor Table Entries to the sysedge.mon File](#) (see page 492)
- [Sample Monitor Table Entries in sysedge.mon](#) (see page 495)
- [Adding Process Monitor Table Entries to the sysedge.mon File](#) (see page 496)
- [Sample Process Monitor Entries in sysedge.mon](#) (see page 498)
- [Adding Process Group Monitor Table Entries to the sysedge.mon File](#) (see page 499)
- [Sample Process Group Monitor Entry in sysedge.mon](#) (see page 501)
- [Adding Log Monitor Table Entries to the sysedge.mon File](#) (see page 501)
- [Sample Log Monitor Entry in sysedge.mon](#) (see page 503)
- [Adding NT Event Monitor Table Entries to the sysedge.mon File](#) (see page 503)
- [Sample NT Event Monitor Entries in sysedge.mon](#) (see page 505)
- [Adding History Control Table Entries to the sysedge.mon File](#) (see page 506)
- [Sample History Control Table Entries in sysedge.mon](#) (see page 507)

CA eHealth SystemEDGE Table Backing Store

On startup, the CA eHealth SystemEDGE agent's sysedge.mon file reports to the agent the state of the self-monitoring tables while the agent was previously running. The agent looks for sysedge.mon in the /etc or %SystemRoot%\system32\ directories by default when it is started, unless you have specified an alternate directory and file name. This monitor configuration file consists of a series of entries, each describing a row in one of the self-monitoring tables.

Note: For more information about valid values and more examples, see the chapter "Configuring Threshold Monitoring."

Edit the sysedge.mon file only when CA eHealth SystemEDGE agent is not running. The CA eHealth SystemEDGE agent overwrites this file every time the stored tables (Monitor, Process Monitor, Process Group Monitor, Log Monitor, NT Event Monitor, or History Control) are modified.

Before you edit this file, copy it from the CA eHealth SystemEDGE agent distribution to the /etc directory or the system root directory, as follows:

- To copy sysedge.mon to the /etc directory on a UNIX system, enter the following command:

```
cp config/sysedge.mon /etc
```

- To copy sysedge.mon to the system root directory on a Windows system, enter the following command at the command prompt:

```
copy config\sysedge.mon %SystemRoot%\system32\
```

The sysedge.mon file consists of comments and table entries. The empty lines and the lines that begin with a pound sign (#) are treated as comments and are ignored. Comments conclude at the end of the each line.

Adding Monitor Table Entries to the sysedge.mon File

Monitor table entries begin with the keyword monitor. They are delineated by open and closed brackets and include ten fields.

Following are the fields of the Monitor table entries for the sysedge.mon file:

monitor {

Specifies the beginning of the entry, which is marked by an open bracket ({}).

monIndex

Specifies an integer (1 to MAXINT) that indicates the row index for this entry. Rows 1 through 10 are reserved for the agent's internal use; the index for additional rows must fall in the range of 11 to MAXINT.

The index is particularly important because SNMP does not directly support creation and deletion of MIB objects; instead, it creates and deletes them as side effects of SNMP Set operations. This limitation means that the person creating or modifying Monitor table entries through the monitor command or by editing sysedge.mon must know the exact MIB table index to use for row creation.

monDescr

Specifies a quoted string, 0 to 128 characters in length, that typically contains a description of the monitored object and a severity level for this event.

monInterval

Specifies an integer value (30 to MAXINT) that indicates how often (in seconds) the monitoring should be performed. For example, the value 30 instructs the agent to monitor this entry every 30 seconds.

Note: This value must be a multiple of 30 seconds.

monSampleType

Indicates whether this entry should sample the object's absolute value (absoluteValue(1)) or whether the agent should take the difference between successive samples (deltaValue(2)).

For example, when monitoring counter variables, use deltaValue because it describes the rate of change. When monitoring gauges, use absoluteValue because it describes the object's exact value.

monOID

Specifies the complete object-instance identifier that represents the value to be monitored.

Note: The instance portion of the object-identifier (for example, .0 for scalars) is also required. The object-instance must exist and must be contained within the CA eHealth SystemEDGE agent's supported MIBs. Any supported (integer-based) object that exists in MIB-II, the Host Resources MIB, or the Systems Management MIB is valid. Objects should be of integer type, including counter, gauge, integer, or enumerated integer.

monOperator

Specifies the operator type, which is a Boolean operators used for evaluating the expression:

current operator value

The operator can be one of the following:

- nop (no operation; monitor the object's value, but do not evaluate the Boolean expression)
- gt (greater than)
- lt (less than)
- ge (greater than or equal to)
- le (less than or equal to)
- eq (equal)
- ne (not equal)

monValue

Specifies an integer value to which the current value of the monitored MIB variable is compared during each monitoring cycle. If the comparison evaluates to true, (where the operator-type tells how to compare them), the agent sends a trap.

For example, if you wanted to be notified if the value of some gauge goes more than 100, you would set 100 as the monValue against which the current value of the gauge is compared.

monRowStatus

Specifies the row status, which can be one of the following:

- active
- notInService
- notReady
- createAndGo
- createAndWait

Typically, a row is either active or notInService. These values are identical in meaning to the SNMPv2 SMI RowStatus textual convention.

monAction

Specifies a quoted string, 0 to 256 characters in length, that specifies the full path of the command (with any parameters) which runs, when the expression evaluates to true and a trap is sent. If the string is empty, no action will be performed for this entry.

monFlags

Specifies an unsigned integer flags value that indicates additional behavioral semantics that this row should follow during the course of its operation. By default, this field is assigned the hexadecimal value 0x00.

Note: For more information about this field, see the chapter “Configuring Threshold Monitoring.”

}

Indicates the end of the entry.

Sample Monitor Table Entries in sysedge.mon

This section includes examples for adding entries to the Monitor table through the sysedge.mon file.

Example: Monitor 1-Minute Load Average

The following entry in the sysedge.mon file instructs the CA eHealth SystemEDGE agent to monitor the 1 minute load average every 60 seconds. If the sampled value is greater than 300 (3.00 in this example because SNMP does not support real numbers), the agent sends an SNMP trap message.

```
monentry {  
  11  
  "Monitor 1 minute load average"  
  60  
  absoluteValue  
  1.3.6.1.4.1.546.1.1.7.8.26.0  
  gt  
  300  
  active  
  ""  
  0x0  
}
```

Example: Monitor File Systems

The following entry in the sysedge.mon file instructs the CA eHealth SystemEDGE agent to monitor how full the / file system is every 2 minutes (120 seconds). If the file system becomes 95% full, the agent sends an SNMP trap message.

```
monentry {  
  19  
  "Monitor / filesystem"  
  120  
  absoluteValue  
  1.3.6.1.4.1.546.1.1.1.7.1.14.1  
  ge  
  95  
  active  
  ""  
  0x0  
}
```

Note: The object-instance identifier for the CA eHealth SystemEDGE agent may not be the same across all instantiations of the agent.

Adding Process Monitor Table Entries to the sysedge.mon File

Process Monitor table entries begin with the keyword `processmon`. They are delineated by open and closed brackets and include eleven fields.

The following list describes the fields for Process Monitor table entries in the `sysedge.mon` file:

processmon {

Indicates the beginning of the entry.

pmonIndex

Specifies an integer (1 to MAXINT) that indicates the row index for this entry.

The index is particularly important because SNMP does not directly support creation and deletion of MIB objects; instead, it creates and deletes them as side effects of SNMP Set operations. This limitation means that the person creating or modifying Process Monitor table entries through the `processmon` command or by editing `sysedge.mon` must know the exact MIB table index to use for row creation.

pmonDescr

Specifies a quoted string, 0 to 128 characters in length, that typically contains a description of the process and attribute that the agent is monitoring and a severity level for this event.

pmonInterval

Specifies an integer value (30 to MAXINT) that indicates how often (in seconds) the agent should perform this monitoring. For example, the value 30 instructs the agent to monitor this entry every 30 seconds.

Note: This value must be a multiple of 30 seconds.

pmonSampleType

Indicates whether this entry should sample the object's absolute value (`absoluteValue(1)`) or whether the agent should take the difference between successive samples (`deltaValue(2)`).

For example, when monitoring counter attributes, use `deltaValue` because it describes the rate of change. When monitoring gauges, use `absoluteValue` because it describes the object's exact value.

pmonAttribute

Specifies the process attribute being monitored.

Note: For more information about supported attributes, see the chapter "Configuring Process and Service Monitoring."

For example, to monitor a process to verify that it is alive, specify the attribute `procAlive`. To track the number of packets received by the particular application or process, specify `procMsgsSent`.

pmonOperator

Specifies the operator type, which is a Boolean operator used for evaluating the following expression:

`current operator value`

The operator can be one of the following:

- `nop` (no operation; monitor the object's value, but do not evaluate the Boolean expression)
- `gt` (greater than)
- `lt` (less than)
- `ge` (greater than or equal to)
- `le` (less than or equal to)
- `eq` (equal)
- `ne` (not equal)

pmonValue

Specifies an integer value to which the current value of the monitored process attributes is compared during each monitoring cycle. If the comparison evaluates to True, (where the operator type tells how to compare them), the agent sends a trap.

For example, if you want to be notified if the value of a gauge goes above 100, set 100 as the `pmonValue` against which the current value of the gauge is compared.

pmonFlags

Specifies an integer flags value that indicates additional behavioral semantics this row should follow during the course of its operation. By default, this field is assigned the hexadecimal value 0x00.

Note: For more information about this field, see the chapter "Configuring Process and Service Monitoring."

pmonAction

Specifies a quoted string, 0 to 256 characters in length, that specifies the full path of the command (with any parameters) which runs when the expression evaluates to True and a trap is sent. If the string is empty, no action will be performed for this entry.

pmonRowStatus

Specifies the row status, which can be one of the following:

- `active`

- notInService
- notReady
- createAndGo
- createAndWait

Typically, a row is either active or notInService. These values are identical in meaning to the SNMPv2 SMI RowStatus textual convention.

Sample Process Monitor Entries in sysedge.mon

This section includes examples for adding entries to the Process Monitor table through the sysedge.mon file.

Example: Monitor the Netscape Process Run Status

The following entry in the sysedge.mon file instructs the CA eHealth SystemEDGE agent to monitor the netscape process every 60 seconds. If the process is down, the agent sends an SNMP trap message.

```
processmon {
1
"Monitor netscape alive"
60
absoluteValue
procAlive
eq
4
0x0
""
"netscape"
active
}
```

Example: Monitor the Netscape Process Size

The following entry in the sysedge.mon file instructs the CA eHealth SystemEDGE agent to monitor the memory utilization of the netscape process every 2 minutes (120 seconds). If the process RSS value exceeds 50000, a SNMP Trap message is sent.

```
processmon {  
  2  
  "Monitor netscape RSS"  
  60  
  absoluteValue  
  procRSS  
  gt  
  50000  
  0x0  
  ""  
  "netscape"  
  active  
}
```

Adding Process Group Monitor Table Entries to the sysedge.mon File

Process Group Monitor table entries begin with the keyword **procgroupmon**. They are delineated by open and closed brackets.

The following list describes the fields for Process Group Monitor table entries in the sysedge.mon file:

procgroupmon {

Specifies the beginning of the entry.

pgmonIndex

Specifies an integer (1 to MAXINT) that indicates the row index for this entry.

The index is particularly important because SNMP does not directly support creation and deletion of MIB objects; instead, it creates and deletes them as side effects of SNMP Set operations.

This limitation means that the person creating or modifying Process Monitor table entries through the processmon command or by editing sysedge.mon must know the exact MIB table index to use for row creation.

pgmonDescr

Specifies a quoted string, 0 to 128 characters in length, that typically contains a description of the process and attribute that the agent is monitoring and a severity level for this event.

pgmonInterval

Specifies an integer value (30 to MAXINT) that indicates how often (in seconds) the agent should perform this monitoring. For example, the value 30 instructs the agent to monitor this entry every 30 seconds.

Note: This value must be a multiple of 30 seconds.

pgmonProcRegExpr

Specifies the regular expression to apply when the agent is attempting to match processes by name.

pgmonFlags

Specifies the integer flags value that indicates additional behavioral semantics this row should follow during the course of its operation. By default, this field is assigned the hexadecimal value 0x00.

Note: For more information about this field, see the chapter "Configuring Process Group Monitoring."

pgmonAction

Specifies a quoted string, 0 to 256 characters in length, that specifies the full path of the command (with any parameters) which runs when the expression evaluates to True and a trap is sent. If the string is empty, no action will be performed for this entry.

pgmonRowStatus

Specifies the row status, which can be one of the following:

- active
- notInService
- notReady
- createandGo
- createAndWait

Typically, a row is either active or NotInService. These values are identical in meaning to the SNMPv2 SMI RowStatus textual convention.

}

Specifies the end of the entry.

Sample Process Group Monitor Entry in sysedge.mon

This section includes an example for adding entries to the Process Group Monitor table through the sysedge.mon file.

Example: Monitor the httpd Process Group

The following entry in the sysedge.mon file instructs the CA eHealth SystemEDGE agent to monitor the httpd process group every 60 seconds:

```
progroupmon {  
  1  
  "Monitor Web process group"  
  60  
  'httpd'  
  0x0  
  ""  
  active  
}
```

Adding Log Monitor Table Entries to the sysedge.mon File

Log Monitor table entries begin with the keyword logmon. They are delineated by open and closed brackets, and they include seven fields.

The following list describes the field for Log Monitor table entries in the sysedge.mon file:

logmon {

Specifies the beginning of the entry.

logMonitorIndex

Identifies the row of the table.

The index is particularly important because SNMP does not directly support creation and deletion of MIB objects; instead, it creates and deletes them as side effects of SNMP Set operations.

This limitation means that the person creating or modifying Log Monitor table entries through the watch logfile command or by editing sysedge.mon must know the exact MIB table index to use for row creation.

logMonitorLogFile

Specifies the complete path and file name of the log file to be monitored.

logMonitorRegularExpression

Specifies the regular expressions to apply when scanning the log file for matches.

Note: For information about the rules for specifying regular expressions, see the UNIX main page on `egrep(1)`.

logMonitorAction

Specifies a quoted string, 0 to 256 characters in length, that specifies the full path of the command (with any parameters) that runs when the regular expression is matched and a trap is sent. If the string is empty, no action will be performed for this entry.

logMonitorFlags

Specifies the unsigned integer flags value indicating additional behavioral semantics this row should follow during the course of its operation. By default, this field is assigned the hexadecimal value 0x00.

Note: For more information about this field, see the chapter “Configuring Windows Event Monitoring.”

logMonitorRowStatus

Specifies the row status, which can be one of the following:

- active
- notInService
- notReady
- createAndGo
- createAndWait

Typically, a row is either active or notInService. These values are identical in meaning to the SNMPv2 SMI RowStatus textual convention.

}

Specifies the end of the entry.

Sample Log Monitor Entry in sysedge.mon

This section includes an example for adding an entry to the Log Monitor table through the sysedge.mon file.

Example: Monitor for Failed su Attempts

The following entry instructs the CA eHealth SystemEDGE agent to monitor the /var/adm/messages log file for the expression su.*fail. If a match is found, the agent sends an SNMP trap message.

```
logmon {  
  2  
  "SU - WARNING"  
  "/var/adm/messages"  
  "su.*fail"  
  ""  
  0x0  
  active  
}
```

Adding NT Event Monitor Table Entries to the sysedge.mon File

NT Event Monitor table entries begin with the keyword nteventmon. They are delineated by open and closed brackets, and they include seven fields.

The following list describes the field for NT Event Monitor table entries in the sysedge.mon file:

nteventmon {

Specifies the beginning of the entry.

ntEventMonIndex

Identifies the row of the table.

The index is particularly important because SNMP does not directly support creation and deletion of MIB objects; instead, it creates and deletes them as side effects of SNMP Set operations. This limitation means that the person creating or modifying NT Event Monitor table entries through the watch ntevent command or by editing sysedge.mon must know the exact MIB table index to use for row creation.

ntEventMonDescription

Specifies a quoted text string that contains a description of the purpose, function, and (optionally) creator of the entry.

ntEventMonLog

Specifies an integer that designates which Windows Event Log to monitor. The following are possible values:

- Application(1)
- Security(2)
- System(3)

ntEventMonTypeFilter

Identifies the event type to match for this entry. Types 1 through 5 are defined by Windows as follows:

- error(1)
- warning(2)
- information(3)
- success(4)
- failure(5)

Type all(6) indicates that all event types should match.

ntEventMonSrcFilter

Specifies the regular expression to apply to the Event Source when scanning the events for matches.

Note: For more information about specifying regular expressions, see the UNIX man page on `egrep(1)`.

ntEventMonDescFilter

Specifies the regular expression to apply to the Event Description when scanning the events for matches.

Note: For more information about specifying regular expressions, see the UNIX man page on `egrep(1)`.

ntEventMonStatus

Specifies the SNMPv2RowStatus, which can be one of the following:

- active(1)
- notInService(2)
- notReady(3)

ntEventMonAction

Specifies a quoted string, 0 to 256 characters in length, that specifies the full path of the command (with any parameters) which runs when a match is found and a trap is sent. If the string is empty, no action will be performed for this entry.

ntEventMonFlags

Specifies the unsigned integer flags value indicating additional behavioral semantics this row should follow during the course of its operation. By default, this field is assigned the hexadecimal value 0x00.

Note: For more information about this field, see the chapter “Configuring History Collection.”

}

Specifies the end of the entry.

Sample NT Event Monitor Entries in sysedge.mon

This section includes an example for adding an entry to the NT Event Monitor table through the sysedge.mon file.

Example: Monitor for Application Errors

The following entry instructs the CA eHealth SystemEDGE agent to monitor the Application NT Event Log for events of type Error. If a match is found, the agent sends an SNMP trap message.

```
ntheventmon {  
5  
"Application - ERROR"  
Application  
Error  
".*"  
".*"  
active  
""  
0x0  
}
```

Adding History Control Table Entries to the sysedge.mon File

History Control Table entries begin with the keyword `history`. They are delineated by open and closed brackets, and they include six fields.

The following list describes the field for NT Event Monitor table entries in the `sysedge.mon` file:

history {

Specifies the beginning of the entry.

empireHistoryControlIndex

Specifies an integer (1 to MAXINT) that uniquely identifies the entry in the table.

The index is particularly important because SNMP does not directly support creation and deletion of MIB objects; instead, it creates and deletes them as side effects of SNMP Set operations.

This limitation means that the person creating or modifying History Control table entries through the `emphistory` command or by editing `sysedge.mon` must know the exact MIB table index to use for row creation.

empireHistoryControlDescr

Describes the data-collection function defined by this entry, and that may include who created it.

empireHistoryControlBuckets

Specifies the total number of samples to be stored for this variable.

emphistoryControlObjID

Specifies the complete object-instance identifier of the MIB variable to be sample. You must include the instance portion of the object-identifier (for example, .0 for scalars). The object-instance must exist and must be contained within the Systems Management MIB.

For example, any supported (integer-based) object in MIB-II, the Host Resources MIB, or the Systems Management MIB is valid. Objects should be of integer type including counter, gauge, integer, or enumerated integer.

empHistoryControlInterval

Specifies an integer value indicating how often (in seconds) to sample the MIB variable. The interval must be a multiple of 30 seconds.

empHistoryControlStatus

Specifies the control status, which can be one of the following:

- active
- notInService

- notReady
- createAndGo
- createAndWait

These values are identical in meaning to the SNMPv2 SMI RowStatus textual convention.

Setting the status to destroy(6) causes the agent to discontinue history sampling for this entry, and to delete both this row and the corresponding data sample rows in the empireHistoryTable.

}

Specifies the end of the entry.

Sample History Control Table Entries in sysedge.mon

This section includes an example for adding an entry to the History Control table through the sysedge.mon file.

Example: Disk Transfer History

The following entry instructs the CA eHealth SystemEDGE agent to collect the disk transfer statistics for the first physical disk. This is entry index 10. It will keep 100 samples and take a new sample every 60 seconds.

```
history {
10
"Disk 1 Transfers"
100
1.3.6.1.4.1.546.12.1.1.6.1
60
active
}
```


Appendix D: Textual Conventions for Row Status

This appendix provides information about the SNMPv2 textual conventions for row status. This appendix contains material from RFC 1443, Textual Conventions for SNMPv2 (Case et al., 1993).

This section contains the following topics:

[RFC 1443: Textual Conventions for SNMPv2](#) (see page 509)

[Conceptual Row Creation](#) (see page 513)

[Conceptual Row Suspension](#) (see page 517)

[Conceptual Row Deletion](#) (see page 518)

RFC 1443: Textual Conventions for SNMPv2

The RowStatus textual convention manages the creation and deletion of conceptual rows and is the value of the SYNTAX clause for the status column of a conceptual row.

RowStatus ::= TEXTUAL-CONVENTION

STATUS current

The status column has six defined values:

active

Indicates that the conceptual row is available for use by the managed device.

notInService

Indicates that the conceptual row exists in the agent, but is unavailable for use by the managed device.

notReady

Indicates that the conceptual row exists in the agent, but is missing the necessary information for use by the managed device.

createAndGo

Supplied by a management station wanting to create an instance of a conceptual row and to have it available for use by the managed device.

createAndWait

Supplied by a management station wanting to create an instance of a conceptual row but not to have it available for use by the managed device.

destroy

Supplied by a management station wanting to delete all of the instances associated with an existing conceptual row.

Whereas five of the six values (all except notReady) may be specified in a management protocol set operation, only three values will be returned in response to a management protocol retrieval operation. When queried, an existing conceptual row can have one of the following states:

active

Indicates it is available for use by the managed device.

notInService

Indicates it is not available for use by the managed device, though the agent has sufficient information to make it available.

notReady

Indicates it is not available for use by the managed device because the agent lacks sufficient information.

Note: This textual convention may be used for a MIB table, regardless of whether the values of that table's conceptual rows can be modified while it is active, or whether its conceptual rows must be taken out of service to be modified. That is, it is the responsibility of the DESCRIPTION clause of the status column to specify whether the status column must be notInService for the value of some other column of the same conceptual row to be modified.

The following table describes the effect of the conceptual row with status column, where the SYNTAX clause value of the status column is RowStatus:

| Action | State | | | |
|----------------------------------|--|---------------------------|-------------------------------|-------------------------|
| | A | B | C | D |
| | status column does not exist | status column is notReady | status column is notInService | status column is active |
| set status column to createAndGo | noError → D or inconsistentValue | inconsistentValue | inconsistentValue | inconsistentValue |

| Action | State | | | |
|------------------------------------|--|--|-------------------------------|-------------------------|
| | A | B | C | D |
| | status column does not exist | status column is notReady | status column is notInService | status column is active |
| set status column to createAndWait | noError (Note: Go to column B or C, depending on information available to the agent.) or wrongValue | inconsistentValue | inconsistentValue | inconsistentValue |
| set status column to active | inconsistentValue | inconsistentValue Note: If other variable bindings included in the same PDU, provide values for all columns which are missing but required, then return noError and go to column D. → D | noError → D | noError → D |

| Action | State | | | |
|------------------------------------|--|--|-------------------------------|---------------------------------|
| | A | B | C | D |
| | status column does not exist | status column is notReady | status column is notInService | status column is active |
| set status column to notInService | inconsistentValue | inconsistentValue Note: If other variable bindings included in the same PDU, provide values for all columns which are missing but required, then return noError and go to column C. → C | noError | noError → C or wrongValue |
| set status column to destroy | noError → A | noError → A | noError → A | noError → A |
| set any other column to some value | Note: At the discretion of the agent, either noError or inconsistentValue may be returned. → A | noError Note: Go to column B or C, depending on information available to the agent. | noError → C | noError → D |

Note: Other processing of the set request may result in a response other than noError being returned, for example, wrongValue, noCreation, and so on.

Conceptual Row Creation

There are four potential interactions when creating a conceptual row:

- Selecting an instance-identifier which is not in use
- Creating the conceptual row
- Initializing any objects for which the agent does not supply a default
- Making the conceptual row available for use by the managed device

Interaction 1: Selecting an Instance-Identifier

The algorithm used to select an instance- identifier varies for each conceptual row. In some cases, the instance-identifier is semantically significant, for example, the destination address of a route, and a management station selects the instance-identifier according to the semantics.

In other cases, the instance-identifier is used solely to distinguish conceptual rows, and a management station without specific knowledge of the conceptual row might examine the instances present to determine an unused instance-identifier. (This approach may be used, but it is often highly suboptimal; however, it is also a questionable practice for a naive management station to attempt conceptual row creation.)

Alternately, the MIB module which defines the conceptual row might provide one or more objects which provide assistance in determining an unused instance-identifier. For example, if the conceptual row is indexed by an integer-value, then an object having an integer-valued SYNTAX clause might be defined for such a purpose, enabling a management station to issue a management protocol retrieval operation. To avoid unnecessary collisions between competing management stations, adjacent retrievals of this object should be different.

Finally, the management station could select a pseudo-random number to use as the index. In the event that this index was already in use and an inconsistentValue was returned in response to the management protocol set operation, the management station should simply select a new pseudo-random number and retry the operation.

A MIB designer should select between the two latter algorithms based on the size of the table (and therefore the efficiency of each algorithm). For tables in which a large number of entries are expected, define a MIB object that returns an acceptable index for creation. For tables with small numbers of entries, use the latter pseudo-random index mechanism.

Interaction 2: Creating the Conceptual Row

After selecting an unused instance-identifier, the management station determines if it wants to create and activate the conceptual row in one transaction or in a negotiated set of interactions.

Interaction 2a: Creating and Activating the Conceptual Row

The management station must first determine the column for which it has to provide values. Depending on the complexity of the table and the management station's knowledge of the agent's capabilities, this determination can be made locally by the management station. Alternately, the management station issues a management protocol get operation to examine all columns in the conceptual row that it wants to create.

In response, for each column, there are three possible outcomes:

- A value is returned, indicating that some other management station has already created this conceptual row. We return to interaction 1.
- The exception `noSuchInstance` is returned, indicating that the agent implements the object-type associated with this column, and that this column in at least one conceptual row would be accessible in the MIB view used by the retrieval were it to exist. For those columns to which the agent provides read- create access, the `noSuchInstance` exception tells the management station that it should supply a value for this column when the conceptual row is to be created.
- The exception `noSuchObject` is returned; indicating that the agent does not implement the object-type associated with this column or that there is no conceptual row for which this column would be accessible in the MIB view used by the retrieval. As such, the management station can not issue any management protocol set operations to create an instance of this column.

A management protocol set operation is issued after determining the column requirements. This operation also sets the new instance of the status column to `createAndGo`.

When the agent processes the set operation, it verifies that it has sufficient information to make the conceptual row available for use by the managed device. The information available to the agent is provided by two sources: the management protocol set operation which creates the conceptual row, and, implementation-specific defaults supplied by the agent.

Note: The agent must provide implementation-specific defaults for at least those objects which it implements as read-only.

If there is sufficient information available, then the conceptual row is created, a `noError` response is returned, the status column is set to active, and no further interactions are necessary; interactions 3 and 4 can be skipped.

If there is insufficient information, then the conceptual row is not created, and the set operation fails with an error of `inconsistentValue`. On this error, the management station can issue a management protocol retrieval operation to determine if this was because it failed to specify a value for a required column, or, because the selected instance of the status column already existed. In the latter case, we return to interaction 1. In the former case, the management station can re-issue the set operation with the additional information, or begin interaction 2 again using `createAndWait` to negotiate creation of the conceptual row.

Note: Regardless of the method used to determine the column requirements, it is possible that the management station might deem a column necessary when, in fact, the agent will not let that particular columnar instance to be created or written. In this case, the management protocol set operation will fail with an error such as `noCreation` or `notWriteable`. In this case, the management station decides whether it needs to be able to set a value for that particular columnar instance. If not, the management station re-issues the management protocol set operation, but without setting a value for that particular columnar instance; otherwise, the management station aborts the row creation algorithm.

Interaction 2b: Negotiating the Creation of the Conceptual Row

The management station issues a management protocol set operation which sets the requested instance of the status column to `createAndWait`. If the agent cannot process a request of this sort, the set operation fails with a `wrongValue` error. (Consequently, such an agent must be prepared to accept a single management protocol set operation, containing all of the columns indicated by its column requirements. For more information, see Interaction 2a: Creating and Activating the Conceptual Row.)

Otherwise, the conceptual row is created, a `noError` response is returned, and the status column is immediately set to either `notInService` or `notReady`, depending on whether it has sufficient information to make the conceptual row available for use by the managed device. If there is sufficient information available, then the status column is set to `notInService`; otherwise, if there is insufficient information, then the status column is set to `notReady`. Regardless, we proceed to interaction 3.

Interaction 3: Initializing Non-defaulted Objects

The management station must now determine the column requirements. It issues a management protocol get operation to examine all columns in the created conceptual row.

In the response, for each column, there are three possible outcomes:

- A value is returned; indicating that the agent implements the object-type associated with this column and had sufficient information to provide a value. For those columns to which the agent provides read-create access, a value return tells the management station that it may issue additional management protocol set operations, if required, to change the value associated with this column.
- The exception `noSuchInstance` is returned, indicating that the agent implements the object-type associated with this column, and that this column in at least one conceptual row would be accessible in the MIB view used by the retrieval were it to exist. However, the agent does not have sufficient information to provide a value, and until a value is provided, the conceptual row may not be made available for use by the managed device. For those columns to which the agent provides read-create access, the `noSuchInstance` exception tells the management station that it must issue additional management protocol set operations to provide a value associated with this column.
- The exception `noSuchObject` is returned; indicating that the agent does not implement the object-type associated with this column or that there is no conceptual row for which this column would be accessible in the MIB view used by the retrieval. As such, the management station can not issue any management protocol set operations to create an instance of this column.

If the value associated with the status column is `notReady`, then the management station must first deal with all `noSuchInstance` columns, if any. Having done so, the value of the status column becomes `notInService`, and we proceed to interaction 4.

Interaction 4: Making the Conceptual Row Available

After the management station is satisfied with the values associated with the columns of the conceptual row, it issues a management protocol set operation to set the status column to active. If the agent has sufficient information to make the conceptual row available for use by the managed device, the management protocol set operation succeeds (a `noError` response is returned). Otherwise, the management protocol set operation fails with an error of `inconsistentValue`.

Note: A conceptual row having a status column with value `notInService` or `notReady` is unavailable to the managed device. As such, it is possible for the managed device to create its own instances during the time between the management protocol set operation which sets the status column to `createAndWait` and the management protocol set operation which sets the status column to active. In this case, when the management protocol set operation is issued to set the status column to active, the values held in the agent supersede those used by the managed device.

If the management station is prevented from setting the status column to active (for example, due to management station or network failure) the conceptual row will be left in the '`notInService`' or `notReady` state, consuming resources indefinitely. The agent must detect conceptual rows that have been in either state for an abnormally long period of time and remove them. This period of time should be long enough for human response (including think time) between the creation of the conceptual row and the setting of the status to active. It is suggested that this period be approximately 5 minutes in length.

Conceptual Row Suspension

When a conceptual row is active, the management station may issue a management protocol set operation which sets the instance of the status column to `notInService`. If the agent is unwilling to do so, the set operation fails with an error of `wrongValue`. Otherwise, the conceptual row is taken out of service, and a `noError` response is returned. It is the responsibility of the `DESCRIPTION` clause of the status column to indicate under what circumstances the status column should be taken out of service (for example, to modify the value of some other column of the same conceptual row).

Conceptual Row Deletion

For deletion of conceptual rows, a management protocol set operation is issued which sets the instance of the status column to destroy. This request may be made regardless of the current value of the status column (for example, it is possible to delete conceptual rows which are notReady, notInService or active.) If the operation succeeds, then all instances associated with the conceptual row are immediately removed.

SYNTAX

INTEGER { }

The following table describes the management protocol set operation for deletion of conceptual rows:

| Value | Type of Value | Permissions |
|------------------|---------------|-------------|
| active(1) | State | read/write |
| notInService(2) | State | read/write |
| notReady(3) | State | read-only |
| createAndGo(4) | Action | write-only |
| createAndWait(5) | Action | write-only |
| destroy(6) | Action | write-only |

Appendix E: SNMPv3 in CA eHealth SystemEDGE

SNMP provides an enhanced level of security with SNMPv3. SNMPv3 has the following levels of communication:

- **noAuthNoPriv** is similar to SNMPv1 and SNMPv2. Messages are accompanied by a username, which must be consistent between the sender and the receiver.
- **AuthNoPriv** uses a consistent username and a password.
- **AuthPriv** uses a username, a password, and an encryption key. This key encrypts the body of the message.

To maintain the consistent information, both the sender and receiver must be synchronized and the information made secure; an encryption tool is provided.

CA eHealth SystemEDGE takes advantage of the benefits of SNMPv3; the agent is based on User-based Security Model (USM) defined for SNMPv3. The security information is provided in a configuration file. The agent determines whether a particular request has read and write access, or only read access. The agent can also determine access based on the source of the request.

This section contains the following topics:

[SNMPv3 Configuration](#) (see page 519)

[Configuring FIPS 140-2 Mode](#) (see page 528)

[Encrypt the SNMPv3 Configuration File](#) (see page 528)

[Disable SNMPv1/SNMPv2c](#) (see page 528)

[Command Line Utilities using SNMPv3](#) (see page 529)

SNMPv3 Configuration

You can configure CA eHealth SystemEDGE to use SNMPv3 based communication. CA eHealth SystemEDGE provides a configuration file, `sysedgeV3.cf`, for configuring SNMPv3 user and key information.

Modifying the SNMPv3 Configuration File

The SNMP administrator is the gatekeeper for communication with the CA eHealth SystemEDGE agent on a network node. The sysedgeV3.cf file contains policy defining how the SNMP administrator handles responsibilities and specifies the level of security expected when accessing a host. The sysedgeV3.cf file is located in the config subdirectory of the agent's installation directory.

The sysedgeV3.cf configuration file lets you specify the following:

- SNMP engine ID prefix
- SNMPv3 security users
- The level of security
- Authentication protocol and its associated password
- Encryption (privacy) protocol and its associated password

SNMPv3 User Configuration

The user configuration section of the SNMPv3 configuration file expects two types of keywords, SNMP_V3_ENGINE_ID or SNMP_V3_USER_INFO.

SNMP_V3_ENGINE_ID Keyword

The SNMP_V3_ENGINE_ID keyword specifies a textual SNMP engine ID prefix, which will be concatenated with a process ID and IP address by the agent's SNMP library. The default value is SystemEDGEAdmin. Do not use spaces in the configured string.

For example, the following line in the SNMPv3 configuration file specifies the string CompanySNMPV3ADMIN as a prefix for SNMP engine ID:

```
SNMP_V3_ENGINE_ID      CompanySNMPV3ADMIN
```

Note: This value should not contain any spaces.

SNMP_V3_USER_INFO Keyword

The SNMP_V3_USER_INFO keyword specifies the SNMPv3 USM user's information and security information. The SNMP_V3_USER_INFO keyword has the following syntax. All of the configuration fields for a SNMPv3 user must be on one line and in the specified order separated by blank spaces:

```
SNMP_V3_USER_INFO contextName userName securityModel securityLevel authProtocol  
authPassword privProtocol privPassword
```

contextName

Specifies the context name used by the agent in the following format (no blank spaces are allowed; blank spaces are provided in this usage for reading clarity only):

```
mibName<:InstanceName><|access|ip_filter>
```

mibName<:InstanceName>

Specifies access to a mibName and an instance name. * (asterisk) is the only supported value in this field.

access

Specifies read or write access. Value "read" or "write" is mandatory.

***|read**

Specifies that the user will have read-only access to the agent.

***|write**

Specifies that the user will have read and write access to the agent.

ip-filter

Specifies an IP filter to filter the requests originated from a specified IP address or a subnet. This field is not mandatory. If this is not specified, agent information is accessible to all of the hosts. See Address Filtering for SNMPv3 Users for more information.

userName

Specifies the name of the SNMPv3 secure user.

securityModel

Specifies the SNMPv3 security model in use. The CA Health SystemEDGE agent currently only supports the User-based Security Model (USM). Only a value of **3** is supported.

securityLevel

The following values are supported for the supported levels of security:

noAuthNoPriv

Indicates that no authentication and no privacy (encryption) protocols are configured for use for this SNMPv3 user.

AuthNoPriv

Indicates that an authentication protocol is configured and no privacy protocol is configured for this SNMPv3 user.

AuthPriv

Indicates that an authentication and a privacy protocol is configured for use with this SNMPv3 user.

authProtocol

Specifies the authentication protocol to be used. Currently MD5 and SHA protocols are only used. Specify *MD5* or *SHA* to indicate the type of authentication protocol to use.

You should only specify this option if AuthPriv or AuthNoPriv security level is set.

authPassword

Specifies the SNMPv3 user's authentication password (key) used by the authentication protocol. Specifying authPassword is required if authProtocol (MD5 or SHA) is set.

You should only specify this option if AuthPriv or AuthNoPriv security level is set.

privProtocol

Specifies the encryption (privacy) protocol used by the SNMPv3 user. DES, 3DES, and AES-128 are the only protocols supported. Specify the value *DES*, *3DES*, or *AES* (for AES-128).

If you specify an encryption protocol, you must specify authProtocol and authPassword also. If you specify privProtocol, AuthPriv is the only supported securityLevel.

privPassword

Specifies the SNMPv3 user's encryption password (key) used by the encryption protocol. privPassword is required if you set privProtocol.

You can assign read or write access to different security levels. For example, the security levels of No Authentication and No Privacy (noAuthNoPriv) can be equivalent to the public community string, while Authentication and Privacy (AuthPriv) can be equivalent to the admin community string.

Examples

Examples of valid SNMPv3 user definitions follow:

```
SNMP_V3_USER_INFO * joedoe1 3 AuthPriv MD5 apass AES ppass
```

```
SNMP_V3_USER_INFO *|read joedoe2 3 AuthPriv SHA apass DES ppass
```

```
SNMP_V3_USER_INFO *|write|138.42.29.0 joedoe3 3 AuthPriv SHA apass 3DES ppass
```

```
SNMP_V3_USER_INFO *|write|130.10-255.100.101,e000-  
efff:f0ff:bef0:*,130.10.120.0,*:1 joedoe4 3 AuthNoPriv SHA evansar  
  
SNMP_V3_USER_INFO *|read joedoe5 3 noAuthNoPriv
```

Address Filtering for SNMPv3 Users

The SNMPv3 configuration file lets you specify a source address filter to restrict access to the information only to those source addresses which match the address filter.

Following are supported syntax and usage guidelines for the address filtering field:

- Full IPv4 addresses are supported.

Example:

```
130.10.100.101
```

This line specifies that the requests from the source address 130.10.100.101 will be allowed access to the agent.

- Full IPv6 addresses are supported. Specify hexadecimal notation for IPv6 addresses. You can specify characters in lowercase or uppercase. Following is an example IPv6 address:

```
Ea2f:fe90:abcd:0000:230:a2f:200:ad01
```

This line specifies that the requests from the source address ea2f:fe90:abcd:0000:230:a2f:200:ad01 will be allowed access to the agent.

- Host names are supported. However, we highly recommend using IP addresses as filters instead of host names, because a machine might have multiple IP addresses associated with it, and not all of the IP addresses might be associated with the host-name in the DNS.

Example:

```
box1.domain.com
```

This line specifies that the requests from box1.domain.com will be allowed access to the agent.

Note: If the host name is specified, agent must be able to resolve the IP address that is received by the agent to the specified source host-name

- Wild card character asterisks (*) are supported. Use wild card characters to specify a range of addresses that can access the agent.

Example:

```
130.10.*.101
```

This line specifies that the requests from the source IPv4 addresses starting from 130.10 and ending with 101 (i.e. 130.10.0.101 to 130.10.255.101) will be allowed access to the agent.

Example:

```
Ea2f:fe90*:ad01
```

This line specifies that the requests from the source IPv6 addresses starting with ea2f:fe90 and ending with ad01 will be allowed access to the agent.

- If a wild card character (*) is specified, it should be the only character in that IP field. IP fields are the characters between dot "." (for IPv4 addresses) and colon ":" (for IPv6 addresses).

For example, an entry of 130.10.3*.200 is invalid because the 3 cannot be within the same dot as the asterisk. An entry of Ea2f:fe*:abcd:0000:230:a2f:200:ad01 is also invalid because the 'fe' cannot be within the same colon as the asterisk.

- In the case of multiple wild cards, only the leftmost wild card is stretched to fill the missing fields. For example, entering 2002*:12f4*:1012 is the same as entering 2002*:*:*:12f4*:1012.
- 0 (zero) is treated as a wild card character.

Example:

```
130.10.120.0
```

This line specifies that the requests from the IPv4 subnet 130.10.120 will be allowed access to the agent. Entering the address this way is the same as specifying 130.10.120.*.

Example:

```
Ea2f:fe90:0:ad01
```

This line specifies that the request from IPv6 addresses starting with ea2f:fe90 and ending with ad01 will be allowed access to the agent. This is the same as entering Ea2f:fe90*:ad01.

- Use two or more zeros to remove the wild card behavior of a single zero and to treat it as a literal zero.

Example:

```
130.000.120.00
```

This line specifies that the requests from the IPv4 address 130.0.120.0 will be allowed access to the agent.

Example:

```
Ea2f:*:0000:000:ad01
```

This line specifies that requests from the IPv6 addresses starting with ea2f and ending with 0:0:ad01 will be allowed access to the agent.

- Partial IP addresses are supported. Partial addresses are treated as partial addresses followed by a wild card (*). Any requests coming from the source address starting with the partial address are allowed.

For example, 130.10 would be same as specifying 130.10.*. Also, Ea2f:fe90 would be same as specifying Ea2f:fe90:.*.

- For IPv6 addresses, two consecutive colons (::) are treated as a wildcard between two colons (:*:.).
- A range of addresses using a dash (-) is supported for IPv4 and/or IPv6 addresses. Any or all the fields can have the range defined.

Examples:

```
130.10-255.100.101
```

This line specifies that the requests from the source IPv4 address range from 130.10 to 130.255 will be allowed access to the agent

```
e000-ffff:f0ff:bef0:*
```

This line specifies that the requests from the source IPv6 address range from e000:* to efff:* will be allowed access to the agent.

- Multiple filters delimited by a comma (,) are supported.

Example:

```
130.10-255.100.101, e000-ffff:f0ff:bef0:*, 130.10.120.0, box2.domain.com
```

- No spaces are allowed.

Notes:

- We highly recommend using IP addresses as filters instead of host names, because a machine might have multiple IP addresses associated with it, and not all of the IP addresses might be associated with the host-name in the DNS.
- When a host has multiple interfaces or IP addresses, you must supply all of its interfaces and IP addresses to access the agent, because the request (such as snmpget, walktree, snmpset, etc) could be sent through any of the interfaces or IP addresses. You must define all IP addresses assigned to the host in the agent's IP filter, or the host will not have access to the agent. As an alternative to defining all IP addresses and interfaces, you can simply specify the host name, as long as you are aware of the limitations of using host names (see previous note).

Configuring SNMPv2c/SNMPv3 Traps

The SNMPv3 configuration file lets you specify the types of traps that the agent should send. The following trap types are supported:

- SNMPv2c traps
- SNMPv2c notifications (also referred to as INFORM requests and confirmed traps)
- SNMPv3 traps
- SNMPv3 notifications (also referred to as INFORM requests and confirmed traps)

SNMPv2c traps and SNMPv2c notifications are sent using a SNMP community string. This community string should be defined in `sysedge.cf` (for example, `community global read-only`).

SNMPv3 traps and SNMPv3 notifications are sent using the SNMPv3 user's credentials. The SNMPv3 user must be defined in the `sysedgeV3.cf` configuration file prior to the trap definition record.

The following key words let you specify types of SNMPv2c and SNMPv3 traps and trap destinations in the `sysedgeV3.cf`:

SNMP_V2_TRAP_INFO

Sends a SNMPv2c trap to a specified destination host.

SNMP_V2_TRAP_INFO has the following syntax:

```
SNMP_V2_TRAP_INFO <destination_host>|<port> <trap_context> <community>
```

destination_host

Specifies the host you want the trap sent to. You can specify a host name or an IP address.

port

Specifies the port number on the destination host that you want to send the trap.

trap_context

* (asterisk) is the only supported value for this field. This value is mandatory.

The following examples define SNMP_V2_TRAP_INFO:

```
SNMP_V2_TRAP_INFO localhost|162 * public
```

```
SNMP_V2_TRAP_INFO sysmanager|1692 * private
```

SNMP_V2_NOTIFICATION_INFO

Sends a notification trap, also known as a confirmed trap or INFORM request. The syntax is similar to SNMP_V2_TRAP_INFO, with two additional optional arguments, timeout and number of retries:

```
SNMP_V2_TRAP_INFO <destination_host>|<port> <trap_context> <community>  
<timeout> <num_of_retries>
```

The following examples define SNMP_V2_NOTIFICATION_INFO:

```
SNMP_V2_NOTIFICATION_INFO localhost|162 * public  
SNMP_V2_NOTIFICATION_INFO sysmanager|1692 * private  
SNMP_V2_NOTIFICATION_INFO localhost|162 * public 30 3  
SNMP_V2_NOTIFICATION_INFO sysmanager|1692 * private 10 1
```

SNMP_V3_TRAP_INFO

Sends a SNMPv3 trap to a specified destination host. The syntax for SNMP_V3_TRAP_INFO is the same as SNMP_V2_TRAP_INFO, except that community string is replaced with a SNMPv3 security user defined in the sysedgeV3.cf configuration file.

```
SNMP_V3_TRAP_INFO <destination_host>|<port> <trap_context> <SNMPv3_user>
```

SNMP_V3_NOTIFICATION_INFO

Sends a SNMPv3 notification trap, also known as a confirmed trap and INFORM request. The syntax is similar to SNMP_V3_TRAP_INFO, with two additional optional arguments, timeout and number of retries:

```
SNMP_V3_NOTIFICATION_INFO <destination_host>|<port> <trap_context>  
<SNMPv3_user> <timeout> <num_of_retries>
```

Note: SNMPv1 trap destinations are configured in sysedge.cf rather than the SNMPv3 configuration file, sysedgeV3.cf.

Agent Addresses of Traps from SystemEDGE

The source addresses of the traps sent from the CA eHealth SystemEDGE agent will be the address that the agent is bound to. By default, the CA eHealth SystemEDGE agent binds to all of the network interfaces, so the traps sent from the agent will use its first successful IP address.

If SystemEDGE is configured to send the traps to a trap receiver (such as xtrapmon) running on the same local server as the agent, the source address will most likely be a loop back address (127.0.0.1 (for IPv4) or ::1 (for IPv6)).

Configuring FIPS 140-2 Mode

You can operate CA eHealth SystemEDGE in FIPS 140-2 mode. For more information, see Configuring FIPS 140-2 Mode in the appendix "Using FIPS 140-2 Encryption".

Encrypt the SNMPv3 Configuration File

Encryption is provided using the encryption (privacy) protocol, DES and the authentication protocol, and SHA using a default CA defined key. The encrypted configuration files can be generated in a central location and shared among several hosts. The CA eHealth SystemEDGE distribution includes a clear text SNMPv3 configuration file, named sysedgeV3.cf.nosec, which you can use as a prototype to create your own SNMPv3 user and security information. The CA eHealth SystemEDGE installation copies the sysedgeV3.cf.nosec file to sysedgeV3.cf (if it does not already exist) in the agent's installation config subdirectory.

For additional security, you can encrypt the SNMPv3 configuration file, sysedgeV3.cf. You must create a clear text version of the file, encrypt it, and then install the encrypted file in the appropriate directory.

To encrypt the SNMPv3 configuration file

1. Test and validate the syntax sysedgeV3.cf.
2. Encrypt sysedgeV3.cf with the following command:

```
se_enc -i sysedgeV3.cf -o sysedgeV3.cf.crypt
```
3. Move the clear text sysedgeV3.cf to an archive area.
4. Rename sysedgeV3.cf.crypt to sysedgeV3.cf.
5. Copy sysedgeV3.cf to the config subdirectory in the CA eHealth SystemEDGE installation directory.

Note: SNMPv3 configuration files can be shared among several hosts.

Disable SNMPv1/SNMPv2c

To turn off the agent's SNMPv1/SNMPv2c communication, you must edit sysedge.cf on each host and comment out or delete the lines containing "community <community-name> <permissions> <access-list>".

Command Line Utilities using SNMPv3

For information about command line utilities using SNMPv3, see the chapter “Command Line Utilities”.

Appendix F: Using the Monitored Windows AIM

This appendix provides information about using the Monitored Windows AIM.

This section contains the following topics:

[Monitored Windows AIM](#) (see page 531)

[Operation of the Monitored Windows AIM](#) (see page 531)

[Limitations of the Monitored Windows AIM](#) (see page 533)

Monitored Windows AIM

Use the Monitored Windows AIM for time-based control of the activity of CA eHealth SystemEDGE monitoring. This AIM applies to self-monitoring, process monitoring, process group monitoring, log file monitoring, history, and NT event monitoring (for Windows only). Using the Monitored Windows AIM stops the production of traps (alarms) from such monitors during periods of the day when they would not be significant without stopping the monitor itself.

Operation of the Monitored Windows AIM

To start the AIM, you must specify the AIM on a line in the SystemEDGE configuration file, `sysedge.cf`, and restart SystemEDGE. The AIM is started automatically by CA eHealth SystemEDGE afterwards.

Examples:

Windows:

```
sysedge_plugin C:\sysedge\plugins\monwin\monwinmod.dll
```

Solaris Sparc 64-bit:

```
sysedge_plugin /opt/EMPSysedge/plugins/monwin/monwinmod-sol64bit.so
```

Note: The name of the AIM varies from platform to platform.

The AIM is controlled by a configuration file (monwin.cf) that is located in the monwin subdirectory under the plugins subdirectory of the CA eHealth SystemEDGE install directory. Entries are made with address monitor type, specific monitor index, the 24-hour based time at which the monitor is to go inactive, the 24-hour based time at which the monitor is to become active, and a flag which indicates whether the timing control is to be active or inactive upon startup of SystemEDGE.

Following is the syntax of a row in monwin.cf:

```
monitor_type index time_off time_on {1 (active) | 2 (inactive)}
```

monitor_type

Specifies the type of monitoring you want to edit. You can specify monentry, processmon, procgroupmonex, logmon, history, or nteventmon (Windows, only).

index

Specifies the index value of a monitor entry in the monitor table that you want to edit.

time_off

Specifies the 24-hour based time when you want the monitor to stop.

time_on

Specifies the 24-hour based time when you want the monitor to start.

{active | inactive}

Indicates whether the timing control is currently exercised. Specify 1 to make the control active or 2 to make it inactive. Note that the value of this field has absolutely no effect on the state of a monitor upon SystemEDGE startup. The field only exists to facilitate activation and deactivation of the timing control for your convenience.

All fields are read/write and are controllable by snmpsets from a management console.

Note: For more information about snmpsets, see the chapter "Command Line Utilities".

The Monitored Windows AIM is capable of re-reading its configuration file while running and either returning to a starting configuration or changing to a new scheme after an edit of the configuration file. Refer to the monwin.cf file and the monwinmod.asn1 file under the monwin subdirectory to identify the read/write OID which triggers this function.

Example

```
processmon 99 0200 0300 1
```

This row addresses a process monitor, with an index of 99. The monitor would temporarily cease at 2 A.M., and it would restart at 3 A.M. The final value of 1 indicates that the timing control is currently active.

Limitations of the Monitored Windows AIM

The AIM functions only on a daily basis. There is no support for day of the week, date of the month, etc.

Any monitor specified in the AIM configuration file that does not actually exist in SystemEDGE's monitor table is ignored until the time when it might exist. Any monitor upon which the AIM executes timing control is ignored when its index is removed from CA eHealth SystemEDGE's monitor table.

The AIM will make best attempts to execute monitor activations and deactivations at the specified times. However, on systems with a significant load, the AIM may not experience a cycle at the exact moment. Ideally then, the time off and the time on for any row should be greater than one minute apart.

Any single monitor may be the target of multiple rows in the AIM's configuration file. That is, a monitor may be switched off and on several times during the day. It is up to you to avoid the logical problems that might result from overlap of off and on times in this case.

Appendix G: FIPS 140-2 Encryption

This appendix describes how to install and enable FIPS mode for CA eHealth SystemEDGE.

This section contains the following topics:

[FIPS 140-2 Mode](#) (see page 535)

[Installing FIPS Libraries](#) (see page 535)

[Platform Support](#) (see page 536)

[Supported Encryption Protocols](#) (see page 536)

[Supported Authentication Protocols](#) (see page 536)

[Configuring FIPS 140-2 Mode](#) (see page 537)

[FIPS Mode Considerations](#) (see page 538)

[Protecting Keys in SystemEDGE](#) (see page 538)

FIPS 140-2 Mode

US Federal regulations require that all new software product sales to the US Federal government use Federal Information Processing Standard (FIPS 140-2) validated encryption algorithms if that product contains encryption.

You can operate CA eHealth SystemEDGE in FIPS mode using a version of the cryptographic library that has been certified according to the rules of the FIPS 140-2 standard.

Installing FIPS Libraries

In order to use FIPS mode of operation, you must install the CA eHealth Advanced Encryption package in addition to the CA eHealth SystemEDGE package. The CA eHealth Advanced Encryption package installs FIPS libraries along with libraries providing more advanced encryption than what the base SystemEDGE package provides.

The following FIPS certified files are installed by the CA eHealth Advanced Encryption package:

Windows

- cryptocme2.dll [FIPS library]
- cryptocme2.sig [FIPS library signature file]

UNIX

- libcryptocme2.so [FIPS library for UNIX except for HP-UX PA-RISC]
- libcryptocme2.sl [FIPS library for HP-UX PA-RISC]
- libcryptocme2.sig [FIPS library signature file]

For more information about how to install CA eHealth Advanced Encryption see the appendix "CA eHealth Advanced Encryption".

Platform Support

RSA B-safe Crypto FIPS compliant libraries are currently only available for the following platforms:

- Windows 32-bit
- Solaris SPARC 32 bit
- Solaris SPARC 64 bit
- HP-UX PA-RISC 32 bit
- AIX 32 bit
- Linux x86 32 bit
- Linux x86_64 32 bit mode

Supported Encryption Protocols

The following encryption protocols are supported by CA eHealth SystemEDGE in the FIPS mode of operation. Anything not listed below will not be supported.

- Data Encryption Standard (DES)
- Triple Data Encryption Standard (3DES)
- Advanced Encryption Standard (AES) using cryptographic keys of 128 bits

Supported Authentication Protocols

Only the following authentication protocol is supported by CA eHealth SystemEDGE in the FIPS only mode of operation. Anything not listed below will not be supported in FIPS only mode.

- Secure Hash Algorithm-1 (SHA)

Configuring FIPS 140-2 Mode

CA eHealth SystemEDGE gives you the ability to configure how the agent should treat its encryption using the `sysedge_fips_mode` option in the SystemEDGE configuration file, `sysedge.cf`.

Following are the options for `sysedge_fips_mode`:

0 (zero)

Indicates non-FIPS mode. The agent will use basic built-in encryption and authentication protocols that are based on the non-certified FIPS code. This is the default mode if `sysedge_fips_mode` is not configured.

1

Indicates FIPS co-existence mode. The agent will try to use FIPS certified encryption and authentication protocols using FIPS certified libraries. However, if FIPS certified libraries could not be located or if the agent fails to load these libraries, the agent's functionality will fall back to using built-in non-certified FIPS encryption and authentication code.

2

Indicates FIPS only mode. Only FIPS certified encryption and authentication code and protocols are supported, and all of the non-certified FIPS code and protocols are disallowed.

Example

Add the following line to the `sysedge.cf` file to run the agent in FIPS only mode:

```
sysedge_fips_mode 2
```

FIPS Mode Considerations

Note the following considerations when using FIPS mode:

- On Windows 32 bit operating systems, the R_SHLIB_LD_LIBRARY_PATH environment variable is set during installation of CA eHealth SystemEDGE. This variable is required for FIPS mode of operation. A reboot is required to enable CA eHealth SystemEDGE in FIPS mode.
- When the CA eHealth SystemEDGE agent is enabled in FIPS mode (sysedge_fips_mode 1 or sysedge_fips_mode 2) on a platform that does not support FIPS mode, its behavior is the same as a platform that supports SystemEDGE FIPS mode.

For example, if you enable FIPS only mode (sysedge_fips_mode 2), and if authorization protocol MD5 is used on a platform that does not have support for FIPS mode, access using MD5 will fail (which is the same behavior as if FIPS only mode was supported on the platform).

Protecting Keys in SystemEDGE

CA eHealth SystemEDGE provides SNMPv3 user configuration using a configuration file, sysedgeV3.cf. This file can be encrypted to protect keys and passwords. For more information, see [Encrypt the SNMPv3 Configuration File](#) in the appendix "SNMPv3 in CA eHealth SystemEDGE".

Appendix H: CA eHealth Advanced Encryption

This chapter describes how to install the CA eHealth Advanced Encryption package for providing additional encryption capability and FIPS 140-2 mode of operation for CA eHealth SystemEDGE.

This section contains the following topics:

[CA eHealth Advanced Encryption](#) (see page 539)

[Supported Platforms](#) (see page 539)

[Supported Encryption Protocols](#) (see page 540)

[Supported Authentication Protocols](#) (see page 540)

[FIPS Compatibility](#) (see page 540)

[Prerequisites for Installation](#) (see page 540)

[Installing CA eHealth Advanced Encryption](#) (see page 540)

[Installed Files](#) (see page 542)

[CA eHealth SystemEDGE Considerations](#) (see page 543)

CA eHealth Advanced Encryption

CA eHealth Advanced Encryption provides additional encryption to what CA eHealth SystemEDGE or CA eHealth TrapEXPLODER provides. This package contains CA ETrust Public Key Infrastructure (ETPKI) libraries that give additional encryption capability and include RSA BSAFE Crypto libraries for FIPS mode of operation.

Supported Platforms

CA eHealth Advanced Encryption is supported and can be installed on any platform that CA eHealth SystemEDGE and CA eHealth TrapEXPLODER supports.

Supported Encryption Protocols

The following encryption protocols are supported by CA eHealth SystemEDGE with the CA eHealth Advanced Encryption package installed:

- Data Encryption Standard (DES)
- Triple Data Encryption Standard (3DES)
- Advanced Encryption Standard (AES) using cryptographic keys of 128 bits

Supported Authentication Protocols

The following encryption protocols are supported by CA eHealth SystemEDGE with the CA eHealth Advanced Encryption package installed:

- Message-Digest algorithm 5 (MD5)
- Secure Hash Algorithm-1 (SHA)

FIPS Compatibility

CA eHealth Advanced Encryption package installs RSA B-safe Crypto FIPS compliant libraries automatically. For detailed information about platforms supported by FIPS mode and any specific considerations, see the appendix "Using FIPS 140-2 Encryption".

Prerequisites for Installation

Having CA eHealth SystemEDGE or CA eHealth TRAPEXPLODER installed is a prerequisite for installing CA eHealth Advanced Encryption.

Installing CA eHealth Advanced Encryption

The following section describes how to install and uninstall CA eHealth Advanced Encryption on Windows and UNIX.

Install CA eHealth Advanced Encryption on Windows

The following CA eHealth Advanced Encryption packages are available:

- Windows x86
- Windows x64 (Intel EM64T and AMD64)
- Windows IA64 (Itanium)

Download the appropriate Windows version of the package and double click to run.

You are prompted whether to install the CA eHealth Advanced Encryption package for CA eHealth SystemEDGE, CA eHealth TrapEXPLODER, or both. Click OK to continue or click Cancel on the installation dialog to quit the installation.

Uninstall CA eHealth Advanced Encryption on Windows

To uninstall CA eHealth Advanced Encryption on Windows, select Change/Remove from the Add/Remove Programs dialog to remove CA eHealth Advanced Encryption.

Note: If you uninstall CA eHealth SystemEDGE, CA eHealth Advanced Encryption is automatically uninstalled.

Install CA eHealth Advanced Encryption on UNIX

Installation packages are available for all UNIX platforms and architectures that SystemEDGE supports. Install the appropriate package suitable to your server and architecture.

To install CA eHealth Advanced Encryption on UNIX

1. Create a temporary directory. You can remove this directory after installation.

```
mkdir <some-temp-dir>
```

2. Change to the temporary directory.

```
cd <some-temp-dir>
```

3. Extract the files.

```
uncompress -c CA_AdvancedEncryption_3.2.0*<plat>.tar.Z | tar xvf -
```

Note: For Linux, the extension of the package name is .gz. Use gunzip instead of uncompress.

4. Install using the script `setup_CA_AdvancedEncryption` with the “install” argument.

```
./setup_CA_AdvancedEncryption install
```

You are prompted whether to install CA eHealth Advanced Encryption for CA eHealth SystemEDGE, CA eHealth TrapEXPLODER, or both. Select the appropriate option.

5. Enter the directory where CA eHealth SystemEDGE or CA eHealth TrapEXPLODER is located.

The install stops CA eHealth SystemEDGE or CA eHealth TrapEXPLODER if you installed the CA eHealth Advanced Encryption package for these products.

Uninstall CA eHealth Advanced Encryption on UNIX

You can uninstall CA eHealth Advanced Encryption on UNIX at any time. Note that if you uninstall CA eHealth SystemEDGE, CA eHealth Advanced Encryption is automatically uninstalled.

To uninstall CA eHealth Advanced Encryption on UNIX

1. Change the current working directory into the SystemEDGE or TrapEXPLODER installation directory.
2. Uninstall using the script `setup_CA_AdvancedEncryption` with the “remove” argument.

```
./setup_CA_AdvancedEncryption remove
```

Note: You cannot run this script from the temporary extracted directory.

Installed Files

When you install CA eHealth Advanced Encryption, the following files are installed on Windows systems:

- The following installed files which provide authentication and encryption (non-FIPS mode):
 - `libetpki2.dll`
 - `libetpki2_thread.dll`
 - `libetpki_openssl_crypto.dll`
 - `ipthread.dll`

- The following are installed RSA BSAFE Crypto FIPS files for FIPS mode of operation. These are currently only installed on Windows x86.

- cryptocme2.dll
- cryptocme2.sig

For more information, see the appendix "Using FIPS 140-2 Encryption".

When you install CA eHealth Advanced Encryption, the following files are installed on UNIX:

- The following installed files provide authentication and encryption (non-FIPS mode):

- libetpki2.so
- libetpki2_thread_posix.so
- libetpki_openssl_crypto.so
- libetpki_openssl_ssl.so

Note: The file extensions for HP-UX PA-RISC is .sl instead of .so.

- The following are installed RSA BSAFE Crypto FIPS files for FIPS mode of operation:

- libcryptocme2.so
- libcryptocme2.sig

For more information, see the appendix "Using FIPS 140-2 Encryption".

Note: The file extension for HP-UX PA-RISC is .sl instead of .so.

CA eHealth SystemEDGE Considerations

CA eHealth Advanced Encryption package contains CA ETrust Public Key Infrastructure (ETPKI) libraries. ETPKI is widely used by many CA products.

Take note of the following considerations when installing CA eHealth Advanced Encryption with CA eHealth SystemEDGE:

- CA eHealth SystemEDGE uses ETPKI libraries from its installed location only if CA eHealth Advanced Encryption package is installed.
- CA eHealth SystemEDGE uses ETPKI libraries from the operating system library path if a CA product installs ETPKI libraries and includes the ETPKI library's location in the operating system's library path environment variables (varies by platform).

- CA eHealth SystemEDGE agent and any of its utilities will not function if CA eHealth Advanced Encryption package is not installed and the operating system bitness does not match the bitness of ETPKI libraries installed with other CA products.
- For example, if CA eHealth SystemEDGE is installed on HP-UX 11.23 PA-RISC 64-bit, and ETPKI libraries are installed in 32 bit mode, CA eHealth SystemEDGE cannot function.

Following are the operating system library path environment variables:

Linux, Solaris, HP-UX PA-RISC 64 bit, HP-UX Itanium 64, and DEC-TRU64:

LD_LIBRARY_PATH

HP-UX PA-RISC 32 bit and 64 bit:

SHLIB_PATH

AIX:

LIBPATH

Windows:

PATH

Index

A

access

communities

- configuring • 66
- default • 67
- setting • 38

lists, specifying • 67

action utilities • 403

- bounce.exe • 404
- checkfile.exe • 405
- email.exe • 406
- getver.exe • 407
- nt4bigmem.exe • 408
- restartproc.exe • 408
- restartproc.sh • 411
- restartsvc.exe • 410

actions

- configuring support • 74
- default parameters
 - Log Monitor table • 270
 - Monitor table • 181, 182
 - NT Event Monitor table • 297
 - Process Group Monitor table • 257
 - Process Monitor table • 226
- display file information • 407
- display file size • 405
- display memory information • 408
- error messages • 423
- Monitor table • 181, 182
- NT Event Monitor table • 297
- overview • 26
- Process Group Monitor table • 257
- Process Monitor table • 226
- reboot system • 404
- restart a process • 408, 411
- restart a service • 410
- send an email • 406
- utilities • 403

adding

- custom MIB objects • 26, 327
- entries

- History Control table • 324, 506
- Log Monitor table • 280, 501
- Monitor table • 492
- NT Event Monitor table • 503
- Process Group Monitor table • 259
- Process Monitor table • 496, 499
- performance registry variables • 337

addrChangeTrap • 155

AdvantEDGE View

configuring

- History Control table • 316
- Log Monitor table • 272
- Monitor table • 184
- NT Event Monitor table • 298
- Process Monitor table • 227, 257

using with SystemEDGE • 28

agent multiplexing • 94

AIMs

- See also eHealth application insight modules. • 30
- configuring support • 80
- enabling in sysedge.cf • 80
- overview • 30

AIX

- installing SystemEDGE • 54
- removing SystemEDGE • 61
- SNMP agent • 97

asset tracking • 25

assigning entry rows

- Monitor table • 184
- Process Group Monitor table • 258
- Process Monitor table • 228

audience • 19

authentication failure traps

- UNIX • 69

automating

- deployment of SystemEDGE agent
 - configuring software • 350
 - methods • 348
 - security issues • 351

B

- Boot Configuration group, Systems Management MIB • 104

bounce.exe utility • 404

C

- CA eHealth Advanced Encryption
 - about • 539
 - and ETPKI • 543
 - CA eHealth Advanced Encryption, installing
 - 541
 - files installed with • 542
 - FIPS • 540
 - platform support • 539
 - supported protocols • 540
 - uninstalling • 541, 542
- changing process nice value • 109
- checkfile.exe utility • 405
- CIM SNMP agent • 98
- collecting history
 - History Control table • 137
 - history sampling • 311
 - History table • 136
 - overview • 25
- configuration files
 - overview • 57
 - sysedge.mon • 186
- configuring
 - access communities • 66
 - actions • 74
 - agent debugging support • 72
 - AIM support • 80
 - alternative syslog facilities
 - UNIX • 71
 - Windows • 72
 - authentication failure traps
 - UNIX • 69
 - disk probing • 74
 - during installation • 64
 - extension variables • 329
 - history collection support • 78
 - History Control table
 - AdvantEDGE View • 316
 - dynamically • 318
 - emphistory directive • 317
 - methods • 317
 - IP family for SNMP UDP communications • 79
 - Linux free memory support • 81
 - log file monitoring • 77
 - Log Monitor table
 - AdvantEDGE View • 272
 - dynamically • 275
 - edgewidth utility • 275
 - methods • 272
 - Monitor table
 - AdvantEDGE View • 184
 - dynamically • 187
 - methods • 185
 - sysedge.mon file • 186
 - NT Event Monitor table
 - AdvantEDGE View • 298
 - dynamically • 299
 - methods • 299
 - watch ntevent directive • 299
 - overview • 63
 - permissions for subprograms • 78
 - Process Group Monitor table
 - dynamically • 258
 - methods • 258
 - overview • 77
 - watch procgroup directive • 259
 - Process Monitor table
 - AdvantEDGE View • 227, 257
 - dynamically • 228
 - methods • 228
 - support • 75
 - watch process directive • 229
 - Remote Shell group support • 71
 - security • 82
 - SNMP bind address • 79
 - SNMPv1 trap communities • 68
 - status checking
 - floppy devices • 73
 - serial ports • 73
 - sysedge.cf file • 65
 - system information
 - UNIX • 65
 - threshold monitoring • 171
 - Top Processes AIM • 80
 - User and Group support • 70
 - Who Table support • 70
 - Windows
 - event monitoring • 289
 - registry and performance variables • 339
- copying sysedge.cf • 65
- CPU Statistics group • 141
- creating log file for daemon messages • 489

D

- daemon messages
 - logging • 489
- debugext.sh • 332
- debugging SystemEDGE agent • 72
- deleting
 - entries
 - History Control table • 324
 - Log Monitor table • 280
 - Process Group Monitor table • 262
 - Process Monitor table • 237, 363, 367
 - interprocess communication • 114
- deploying SystemEDGE agent • 345
- diagsysedge.exe • 354, 413, 415
- disabling
 - entries
 - History Control table • 324
 - Log Monitor table • 280
 - NT Event Monitor table • 308
 - support for remote file system checking • 75
- disk
 - performance statistics, enabling on Windows • 139
 - probing, configuring support • 74
- Disk Statistics group • 139
- Disk Storage Table • 164
- distributing software through protocols • 349
- dynamic configuration
 - History Control table • 318
 - Log Monitor table • 275
 - Monitor table • 187
 - NT Event Monitor table • 299
 - Process Group Monitor table • 258
 - Process Monitor table • 228

E

- edgemon utility
 - error messages • 478
 - overview • 202, 359
 - removing entries
 - Monitor table • 209
 - NT Event Monitor table • 310
 - Process Group Monitor table • 262
 - Process Monitor table • 246
- edgemon utility
 - arguments • 240
 - command arguments • 279
 - error messages • 478

- examples • 242, 280
- introduction • 229
- removing entries
 - Log Monitor table • 282
- syntax • 237, 275, 363
- using • 275, 363
- eHealth • 31
- eHealth application insight modules • 30
- eHealth for Cisco CallManager • 30
- eHealth Service Availability • 29
- eHealth Voice Quality Monitor • 30
- email.exe utility • 406
- emphistory directive
 - example • 318
 - syntax • 317
- emphistory utility
 - overview • 319, 367
- enable disk performance statistics on Windows • 139
- encryption, using FIPS 140-2 • 535
- enterprise system object identifier • 143
- error messages
 - actions • 423
 - authentication failure • 423
 - command-line utilities • 471
 - edgemon utility • 478
 - edgemon utility • 478
 - invalid PID • 424
 - sendtrap utility • 480
 - xtrapmon utility • 483
- ETPKI • 543
- event logs
 - criteria for searching • 290
- examples
 - edgemon utility • 242, 280
 - extending Systems Management MIB • 330
 - Extension group • 330
 - history monitoring • 324
 - log file monitoring • 280
 - ntRegPerf group • 341
 - process group monitoring • 260
 - process monitoring • 234
 - threshold monitoring • 190
 - watch logfile • 274
 - watch ntevent • 301
 - Windows event monitoring • 301, 308
- executing remote commands • 111
- extending Systems Management MIB
 - adding custom MIB objects • 327

- Extension group • 142
- ntRegPerf group • 335
- extension directive
 - specifying additional parameters • 330
 - syntax • 329
- Extension group
 - examples
 - pinging remote system • 330
 - returning NIS domain name • 330
 - features • 328
 - sample MIB branch • 327
 - Systems Management MIB • 142, 327
 - writing extension scripts • 332
- extension scripts, writing • 332
- extension variables
 - configuring • 329
 - editing empire.asn1 • 333
 - editing separate MIB specification • 334
 - examples • 328
 - pinging remote system • 330
 - returning NIS domain name • 330
 - using with management software • 333

F

- Fault Manager • 32
- file system space, monitoring • 102
- filtering addresses • 523
- FIPS 140-2 encryption
 - about • 535
 - configuring • 537
 - files • 542
 - installing libraries • 535
 - installing with advanced encryption • 539, 540
 - platform support • 536
 - supported protocols • 536
- flags
 - Log Monitor table • 267
 - Monitor table • 178
 - NT Event Monitor table • 295
 - Process Group Monitor table • 255
 - Process Monitor table • 220
- floppy devices, configuring status checking • 73

G

- getver.exe utility • 407
- Group table • 107

H

- history collection
 - collecting disk transfer history • 507
 - History Control table • 137
 - History table • 136
 - introduction • 25
 - overview • 311
- History Control table
 - adding entries • 506
 - AdvantEDGE View display • 316
 - columns • 312
 - configuring • 317
 - configuring support • 78
 - configuring with emphistory directive • 317, 367
 - description • 312
 - emphistory utility • 319, 367
 - examples
 - adding entries • 324
 - collecting disk transfer history • 507
 - deleting entries • 324
 - listing entries • 324
 - retrieving stored data samples • 324
 - setting status of entries • 324
 - overview • 312
 - sample sysedge.mon entries • 507
 - Systems Management MIB • 136
- history sampling • 311
- History table
 - columns • 314
 - description • 314
 - Systems Management MIB • 136
- Host Resources MIB
 - Device group • 162
 - Disk Storage table • 164
 - File System table • 166
 - introduction • 21
 - overview • 159
 - Partition table • 165
 - Processor table • 163
 - Running Software group • 167
 - Storage group • 161
 - System group • 160
- Host System group • 101
- HP-UX
 - I/O Buffer Cache group, Systems Management MIB • 117
 - installing SystemEDGE • 48
 - removing SystemEDGE • 61

SNMP agent • 97

I

identifying processes that consume the most CPU • 25

implementing trap severity levels • 422

indicating

- change in IP address • 155

- change in process group • 156

- Log Monitor entry is notReady • 147

- match in log file • 146

- match in Windows event log file • 148

- Monitor event • 144

- monRowsStatus is notReady • 145

- previous condition no longer exists • 150

- process attribute is no longer at threshold • 154

- process attributed has reached threshold • 153

- process has restarted • 152

- process has stopped running • 151

- status of NT Event Monitor entry is notReady • 149

installing SystemEDGE

- AIX • 54

- HP-UX • 48, 52

- Linux • 52, 54

- Solaris • 45

- Tru64 UNIX • 56

- Windows • 41

interprocess communication

- deleting • 114

- tracking • 113

IP family for SNMP UDP communications, configuring • 79

IPv6

- MIB tables • 21

K

Kernel

- Configuration group, Systems Management MIB • 103

- Performance group, Systems Management MIB • 111

keys, protecting • 538

L

licenseTrap • 155

Linux

- installing SystemEDGE • 52

- removing SystemEDGE • 61

Linux free memory

- configuring support • 81

- enabling in sysedge • 81

listing entries

- History Control table • 324

- Log Monitor table • 280

Live Health • 32

log files, monitoring • 77, 263

Log Monitor table

- adding entries, sysedge.mon • 501

- AdvantEDGE View display • 272

- columns • 265

- configuring • 272

- default action parameters • 270

- description • 265

- examples

 - adding an entry • 280

 - deleting entries • 280

 - disabling entries • 280

 - listing entries • 280

 - monitoring log files • 503

 - searching for pop connection attempt • 274

 - searching for su attempts • 274

- flags • 267

- monitoring log files • 77

- overview • 136, 263

- removing entries • 282

- sample sysedge.mon entries • 503

- Systems Management MIB • 136

- watch logfile directive • 273

logging

- action commands • 188

- agent-operating messages • 91

- daemon messages • 489

- syslog messages • 487

logmon keyword • 501

logMonMatch trap • 146

logMonNotReady trap • 147

M

Message Buffer Allocation table, Systems Management MIB • 115

message buffers, monitoring • 115

MIB

- Host Resources • 21

- IPv6 tables • 21

- MIB II • 20
- supported by SystemEDGE agent • 20
- Systems Management • 22
- monitor directive
 - monitor keyword • 492
 - parameters • 188
 - usage • 188
- Monitor table
 - actions • 181, 182
 - adding entries through sysedge.mon • 492
 - AdvantEDGE View display • 184
 - assigning entry rows • 184
 - columns • 173
 - configuring • 185
 - default action parameters • 181, 182
 - description • 173
 - edgemon utility • 202
 - examples
 - monitoring /usr file system • 190
 - monitoring 15-minute load average • 190
 - monitoring 1-minute load average • 190, 495
 - monitoring 5-minute load average • 190
 - monitoring input packet rate • 190
 - monitoring interrupt rate • 190
 - monitoring number of processes • 190
 - monitoring output packet rate • 190
 - monitoring page-fault rate • 190
 - monitoring root file system • 190
 - monitoring SNMP packets received • 190
 - flags
 - definitions • 178
 - overview • 178
 - monitor directive
 - sysedge.cf • 186
 - overview • 134
 - removing entries • 208
 - sample sysedge.mon entries • 495
 - threshold monitoring • 171
- monitor trap • 144
- monitorClear trap • 150
- Monitored windows AIM
 - about • 531
 - limitations • 533
 - using • 531
- monitorEntryNotReady trap • 145
- monitoring
 - /usr file system • 190
 - 15-minute load average • 190
 - 1-minute load average • 190
 - 5-minute load average • 190
 - boot configuration • 104
 - call quality • 30
 - CPU statistics • 141
 - devices • 162
 - disk I/O statistics • 139
 - disk-storage devices • 164
 - DT processes • 260
 - edgemon utility • 275
 - file systems • 102, 166
 - groups of processes • 135, 249
 - host system • 160
 - httpd process, monitoring example in sysedge.mon • 501
 - I/O buffers • 117
 - input packet rate • 190
 - interrupt rate • 190
 - kernel configuration • 103
 - kernel performance • 111
 - log files
 - overview • 263
 - sample Log Monitor table • 136
 - sysedge.mon example • 503
 - message buffers • 115
 - mounted devices • 101
 - netscape process
 - edgemon utility • 242
 - sysedge.mon • 498
 - NFS facilities • 118
 - number of processes • 190
 - output packet rate • 190
 - page-fault rate • 190
 - partitions for disk-storage devices • 165
 - process
 - groups with watch procgroup • 259
 - size • 234
 - status • 230
 - with edgemon utility • 363
 - process attributes • 217
 - processes • 135, 211
 - processors • 163
 - queues • 113
 - root file system • 190
 - RPC facilities • 118
 - running

- processes • 108
- software • 167
- semaphores • 113
- sendmail • 234
- services • 211
- shared memory • 113
- Simple TCP/IP Services service • 234
- size of netscape process • 498
- SNMP packets received • 190
- storage areas • 161
- streams • 116
- Streams subsystem • 104
- system information • 101
- TCPVCS process • 234
- thresholds • 134
- user
 - account information • 106
 - groups • 107
 - logged on to system • 110
- voice quality • 30
- Windows
 - cache performance • 126
 - event logs • 290, 505
 - events • 130
 - memory performance • 128
 - page file performance • 130
 - performance extensions • 131
 - registry • 123, 337
 - services • 124
 - system • 120
 - system performance • 125
 - threads • 122
- Xterm processes • 260
- ypbind process
 - edgewatch • 242
 - watch procAlive • 234
- monolithic agents • 94
- Mounted Devices table • 101
- multiple SNMP agents • 93

N

- NFS group • 118
- nhAddSysEdgeMonEntries command • 31
- nice value, changing • 109
- NT Cache Performance group • 126
- NT Event Monitor group • 130
- NT Event Monitor table
 - actions • 297
 - adding entries • 503

- AdvantEDGE View display • 298
- columns • 292
- configuring • 299
- default action parameters • 297
- description • 292
- event monitoring • 289
- examples
 - monitoring log files • 505
 - searching application log for specific events • 301
 - searching application log for Web server messages • 301
 - searching security log for failure events • 301
- flags • 295
- overview • 78
- removing entries • 309
- sample sysedge.mon entries • 505
- watch ntevent directive • 299
- NT Memory Performance group • 128
- NT Page File Performance group • 130
- NT Registry and Performance Extension group • 131
- NT Registry group • 123
- NT Service group • 124
- NT System group • 120
- NT System Performance group • 125
- NT Thread group • 122
- nt4bigmem.exe utility • 408
- nteventmon utility
 - nteventmon keyword • 503
- ntEventMonMatch trap • 148
- ntEventMonNotReady trap • 149
- ntRegPerf directive
 - syntax • 340
- ntRegPerf group
 - examples
 - returning number of transmitted TCP segments • 341
 - returning path to dump file • 341
 - returning total number of current threads • 341
 - overview • 335
 - sample MIB group • 335
 - unsupported Windows performance data types • 339
 - valid variable types • 336
- ntRegPerf variables
 - editing empire.asn1 • 342

- editing separate MIB specification • 343
- using with management software • 342

O

- obtaining
 - diagnostic information about the agent • 413
 - troubleshooting data • 354, 413
- operating system
 - patches • 422
- optimizing row creation
 - History table • 315
 - Log Monitor table • 267
 - Monitor table • 177
 - NT Event Monitor table • 294
 - Process Group Monitor table • 255
 - Process Monitor table • 219
 - scalar objects • 177

P

- patches required • 422
- perfmon extensions • 26
- private-enterprise traps • 143
- procAlive attribute • 230
- process attributes • 217
- Process Group Monitor table
 - actions • 257
 - assigning entry rows • 258
 - columns • 250
 - configuring support • 77
 - default action parameters • 257
 - description • 250
 - examples
 - monitoring DT process group • 260
 - monitoring emacs process group • 260
 - monitoring httpd process • 501
 - monitoring xterm process group • 260
 - flags • 255
 - overview • 135, 249
 - removing entries • 261
 - sample sysedge.mon entries • 501
 - watch procgroup directive • 259
- process group monitoring
 - configuring support • 77
 - introduction • 24
- Process Monitor table
 - actions

- default parameters • 226
- overview • 226
- adding entries through sysedge.mon • 496, 499
- AdvantEDGE View display • 227, 257
- assigning entry rows • 228
- columns • 214
- configuration examples • 234
- configuring support • 75
- description • 214
- examples
 - monitoring netscape process • 498
 - monitoring process size • 234
 - monitoring sendmail • 234
 - monitoring Simple TCP/IP Service • 234
 - monitoring size of netscape process • 498
 - monitoring TCPSVCS process • 234
 - monitoring ypbind • 234
- flags
 - description • 220
 - overview • 220
- overview • 213
- process attributes • 217
- processClear trap • 220
- removing entries • 245
- sample sysedge.mon entries • 498
- sample table entry • 212
- Systems Management MIB • 135
- using edgewatch utility • 237, 363
- process monitoring
 - introduction • 24
 - sample entry • 212
 - sending signal to a process • 109
 - using edgewatch • 363
- Process table • 108
- processClear trap • 154
- processClear trap, example • 220
- processes
 - automatically restarting • 421
- processmon keyword • 496
- processStart trap • 152
- processStop trap • 151
- procGroupChangeTrap • 156
- procgroupmon keyword • 499
- procThreshold trap • 153

R

- remote file system, status checking • 75

- Remote Shell group
 - configuring support • 71
 - Systems Management MIB • 111
- removing
 - entries
 - History Control table • 324
 - Log Monitor table • 282
 - Monitor table • 208
 - NT Event Monitor table • 309
 - Process Group Monitor table • 261
 - Process Monitor table • 245
 - SystemEDGE
 - AIX • 61
 - HP-UX • 61
 - Linux • 61
 - Solaris • 61
 - Tru64 UNIX • 61
 - Windows • 60
- reserving rows in self-monitoring tables • 185
- restarting processes • 421
- restartproc.exe utility • 408
- restartproc.sh • 411
- restartsvc.exe utility • 410
- retrieving stored data samples • 324
- row status
 - active • 509
 - createAndGo • 509
 - createAndWait • 509
 - destroy • 509
 - notInService • 509
 - notReady • 509
- RPC group • 118

S

- searching
 - application log
 - specific events • 301
 - Web server messages • 301
 - event logs
 - criteria • 290
 - logs
 - pop connection attempts • 274
 - su attempts • 274
 - security log for failure events • 301
 - security issues
 - deploying the agent • 351

- action commands • 352
- extension variables • 352
- MIBs • 352
- self-monitoring • 23
- sending signal to a process • 109
- sendtrap utility
 - error messages • 480
 - using • 372
- serial ports, configuring status checking • 73
- service monitoring • 24
- service startup script • 88
- setting
 - policy for monitoring tables • 184
 - status of History Control table entries • 324
- signal, sending to a process • 109
- SNMP
 - access communities • 38
 - communities • 37
 - deleting rows • 518
 - disabling SNMPv1 and SNMPv2c • 528
 - message types • 36
 - overview • 36
 - requests
 - troubleshooting • 415
 - row creation • 512
 - row suspension • 517
 - supporting multiple agents • 93
 - trap
 - communities • 39
 - sending with sendtrap • 372
 - severity levels • 422
 - troubleshooting • 418
 - utilities • 353
- SNMP bind address, configuring • 79
- SNMP UDP communications, IP family for, configuring • 79
- SNMP_V3_ENGINE_ID • 520
- SNMP_V3_USER_INFO • 521
- snmpget utility, using • 380
- snmpset utility, using • 384
- SNMPv1 traps
 - format • 156
- SNMPv3
 - about • 519
 - address filtering for users • 523
 - configuration file • 519, 520
 - configuring • 519
 - configuring traps • 526
 - configuring users • 520, 521

-
- Solaris
 - installing SystemEDGE • 45
 - removing SystemEDGE • 61
 - starting SystemEDGE • 89
 - Solstice Enterprise Agent • 95
 - specifying
 - access list • 67
 - MIB objects to monitor • 186
 - starting SystemEDGE
 - automatically • 89
 - AIX • 91
 - HP-UX • 90
 - Linux • 90
 - Solaris • 89
 - Tru64 UNIX • 91
 - command line • 87
 - overview • 85
 - service startup script for UNIX • 88
 - Windows command line options • 86
 - starting Windows Master agent • 90
 - stopping Windows Master agent • 85
 - Streams
 - Buffer Allocation table, Systems Management MIB • 116
 - group, Systems Management MIB • 104
 - subprogram support, configuring • 78
 - supporting
 - custom MIB objects • 26
 - multiple SNMP agents
 - agent multiplexing • 94
 - AIX SNMP agent • 97
 - CIM SNMP agent • 98
 - HP-UX SNMP agent • 97
 - Microsoft Windows Extensible • 96
 - monolithic agents • 94
 - overview • 93
 - Solstice Enterprise Agent • 95
 - Tru64 UNIX SNMP agent • 98
 - sysedge.cf configuration file • 63
 - sysedge.mon file
 - adding entries
 - History Control table • 506
 - Log Monitor table • 501
 - Monitor table • 492
 - NT Event Monitor table • 503
 - Process Monitor table • 496, 499
 - backing store • 491
 - examples
 - collecting disk transfer history • 507
 - monitoring 1-minute load average • 495
 - monitoring httpd process • 501
 - monitoring log files • 503
 - monitoring netscape process • 498
 - monitoring size of netscape process • 498
 - monitoring Windows event log files • 505
 - format • 491
 - history keyword • 506
 - location • 57
 - logmon keyword • 501
 - monitor keyword • 492
 - nventmon keyword • 503
 - processmon keyword • 496
 - procgrouppmon keyword • 499
 - sample entries
 - History Control table • 507
 - Log Monitor table • 503
 - Monitor table • 495
 - NT Event Monitor table • 505
 - Process Group Monitor table • 501
 - Process Monitor table • 498
 - updating • 420
 - sysedgev3.cf
 - about • 519, 520
 - address filtering • 523
 - configuring SNMP traps • 526
 - encrypting • 528
 - keywords • 520, 521
 - modifying • 520
 - syslog facility
 - configuring alternative
 - UNIX • 71
 - Windows • 72
 - facility codes • 487
 - logging
 - agent-operation messages • 91
 - daemon messages • 489
 - messages • 487
 - overview • 487
 - priority codes • 487
 - SystemEDGE agent
 - configuration
 - files • 57
 - overview • 63, 353
 - configuring support for debugging • 72
 - default settings • 67
 - guidelines • 33
-

- logging messages • 91
- overview • 19
- removing • 59
- self-monitoring features • 23
- specifying actions • 26
- starting • 85
- supporting custom MIB objects • 26
- traps • 143
- troubleshooting • 413
- uninstalling • 59
- using with
 - eHealth • 31
 - Fault Manager • 32
- using with SNMPv3 • 519
- Windows
 - event monitoring • 24
 - extensions • 26
- Systems Management MIB
 - adding support for Windows registry and performance counters • 335
 - boot configuration parameters • 104
 - configuring support
 - Group information • 70
 - history collection • 78
 - Log Monitor table • 77
 - NT Event Monitor table • 78
 - Process Group Monitor table • 77
 - Process Monitor table • 75
 - Remote Shell group • 71
 - User information • 70
 - Who Table • 70
 - CPU Statistics group • 141
 - Disk Statistics group • 139
 - extending monitoring capability • 327
 - Extension group
 - overview • 327
 - using • 142
 - History table • 136
 - Host System group • 101
 - I/O buffer cache • 117
 - interprocess communications • 113
 - kernel
 - configuration parameters • 103
 - performance • 111
 - Log Monitor table • 136
 - message buffers • 115
 - Monitor table • 134
 - mounted devices • 101
 - NFS group • 118

- NT
 - Cache Performance group • 126
 - Event Monitor group • 130
 - Memory Performance group • 128
 - Page File Performance group • 130
 - Registry and Performance Extension • 131
 - Registry group • 123
 - Service group • 124
 - System group • 120
 - System Performance group • 125
 - Thread group • 122
- ntRegPerf group • 335
- overview • 22, 100
- Process Monitor table • 135
- process table • 108
- remote command execution • 111
- RPC group • 118
- self-monitoring tables • 22
- streams
 - buffers • 116
 - group • 104
- system information • 101
- unsupported MIB objects on Windows • 132
- user
 - accounts • 106
 - groups • 107
 - logged on to system • 110
- Who Table • 110
- Windows-specific groups • 120
- sysvariable utility
 - using • 390, 416

T

- threshold monitoring
 - examples • 190
 - introduction • 23
 - sample entry • 172
 - using edgemon • 359
- Top Processes AIM
 - configuring support • 80
 - overview • 25
- tracking assets • 25
- trap communities
 - overview • 39
- trap communities, SNMPv1
 - configuring • 68
- Trap PDU format • 143
- traps

- address changing • 155
- agent addresses of • 69
- authentication failure • 69
- capturing • 399
- changing process group • 156
- configuring SNMPv3 traps • 526
- enabling monitorClear traps • 178
- format • 156
- implementing severity levels • 422
- license • 155
- LogMonMatch • 146
- LogMonNotReady • 147
- monitorClear
 - description • 150
 - enabling • 186
 - setting in Monitor table • 178
- monitorEntryNotReady • 145, 178
- monitorEvent • 144
- ntEventMonMatch • 148
- ntEventMonNotReady • 149
- private enterprise • 143
- processClear
 - description • 154
 - enabling in Process Monitor table • 220
- processStart
 - description • 152
 - disabling in Process Monitor table • 220
- processStop
 - description • 151
 - setting in Process Monitor table • 220
- processThreshold • 153
- sendtrap utility • 372
- supported by SystemEDGE agent • 143
- xtrapmon utility • 399
- troubleshooting
 - agent not responding to SNMP requests • 415
 - agent not running • 419
 - diagsysedge.exe • 354, 413
 - failed bind call • 419
 - management system not receiving SNMP traps • 418
 - obtaining diagnostic information about the agent • 354, 413
 - required system patches • 422
 - restarting processes • 421
 - SNMP traps • 422
 - SystemEDGE agent • 413
 - tool • 354, 413

- trap messages • 422
- updating sysedge.mon • 420
- verifying that the agent is running • 354, 413, 415
- Tru64 UNIX
 - installing SystemEDGE • 56
 - removing SystemEDGE • 61
 - SNMP agent • 98

U

- uninstalling SystemEDGE
 - UNIX • 61
 - Windows • 59
- unmounting devices • 102
- unsupported Windows performance data types • 339
- updating sysedge.mon file • 420
- User table, Systems Management MIB • 106
- using
 - edgemon utility • 202, 359
 - edgewatch utility • 275
 - eHealth application insight modules • 30
 - eHealth Service Availability • 29
 - emphistory utility • 319
 - extension variables with management software • 333
 - Live Health • 32
 - ntRegPerf variables with management software • 342
 - performance registry variables • 337
 - sysvariable utility • 416
 - watch procgroup directive • 259
- utilities
 - bounce • 404
 - checkfile • 405
 - diagsysedge • 354
 - edgemon
 - using • 202, 359
 - edgewatch
 - monitoring log files • 275
 - monitoring processes • 229
 - process monitoring • 237, 363
 - email • 406
 - emphistory • 367
 - getver • 407
 - nt4bigmem • 408
 - restartproc • 408
 - restartproc.sh • 411
 - restartsvc • 410

- se_enc • 371
- SNMP utilities • 353
- snmpget • 380
- snmpset • 384
- sysvariable • 390
- walktree • 395
- walktree error messages • 483
- xtrapmon • 399

V

- verifying
 - agent
 - response to queries • 416
 - starting at system initialization • 416
 - status • 415
 - that the agent is running • 354, 413, 415

W

- walktree utility, error messages • 483
- walktree utility, using • 395
- watch logfile directive
 - examples • 274
 - syntax • 273
- watch ntevent directive
 - examples • 301
 - syntax • 300
- watch procgroup directive
 - examples • 260
 - syntax • 259
- Who Table
 - configuring support • 70
 - information • 110
- Windows
 - adding registry variables • 335
 - configuring
 - registry and performance variables • 339
 - event monitoring
 - examples • 301
 - introduction • 24
 - log-searching criteria • 290
 - overview • 289
 - Extensible agent • 96
 - installing SystemEDGE • 41
 - monitoring
 - event logs • 290
 - registry and performance • 337
 - registry extensions • 26
 - removing SystemEDGE • 60
 - starting Master agent • 90

- stopping Master agent • 85
- supported registry data types • 337
- unsupported MIB objects • 132
- writing extension scripts • 332

X

- xtrapmon utility
 - authentication • 402
 - data displayed • 402
 - error messages • 483
 - platform information • 401
 - using • 399