

CA SystemEDGE

用户指南

版本 5.7.1



本文档包括内嵌帮助系统和以电子形式分发的材料（以下简称“文档”），其仅供参考，CA 随时可对其进行更改或撤销。

未经 CA 事先书面同意，不得擅自复制、转让、翻印、透露、修改或转录本文档的全部或部分内容。本文档属于 CA 的机密和专有信息，不得擅自透露，或除以下协议中所允许的用途，不得用于其他任何用途：(i) 您与 CA 之间关于使用与本文档相关的 CA 软件的单独协议；或者 (ii) 您与 CA 之间单独的保密协议。

尽管有上述规定，但如果您为本文档中所指的软件产品的授权用户，则您可打印或提供合理数量的本文档副本，供您及您的雇员内部用于与该软件相关的用途，前提是所有 CA 版权声明和标识必须附在每一份副本上。

打印或提供本文档副本的权利仅限于此类软件所适用的许可协议的有效期内。如果该许可因任何原因而终止，您应负责向 CA 书面证明已将本文档的所有副本和部分副本已退还给 CA 或被销毁。

在所适用的法律允许的范围内，CA 按照“现状”提供本文档，不附带任何保证，包括但不限于商品适销性、适用于特定目的或不侵权的默示保证。CA 在任何情况下对您或其他第三方由于使用本文档所造成的直接或间接的损失或损害都不负任何责任，包括但不限于利润损失、投资受损、业务中断、信誉损失或数据丢失，即使 CA 已经被提前明确告知这种损失或损害的可能性。

本文档中涉及的任何软件产品的使用均应遵照有关许可协议的规定且根据本声明中的条款不得以任何方式修改此许可协议。

本文档由 CA 制作。

仅提供“有限权利”。美国政府使用、复制或透露本系统受 FAR Sections 12.212、52.227-14 和 52.227-19(c)(1) - (2) 以及 DFARS Section 252.227-7014(b)(3) 的相关条款或其后续条款的限制。

版权所有 © 2013 CA。保留所有权利。此处涉及的所有商标、商品名称、服务标识和徽标均归其各自公司所有。

CA Technologies 产品引用

本文档引用以下 CA Technologies 产品：

- CA eHealth®
- CA Embedded Entitlements Manager (CA EEM)
- CA IT Asset Manager (CA ITAM)
- CA IT Client Manager (CA ITCM)
- CA Network and Systems Management (CA NSM)
- CA Patch Manager
- CA Server Automation
- CA Service Desk Manager (CA SDM)
- CA Spectrum®
- CA SystemEDGE
- CA Systems Performance for Infrastructure Managers
- CA Virtual Assurance for Infrastructure Managers
- CA Software Delivery

联系技术支持

要获取在线技术帮助以及办公地址、主要服务时间和电话号码的完整列表，请联系技术支持：<http://www.ca.com/worldwide>。

目录

第 1 章：简介	17
CA SystemEDGE	17
读者	17
相关出版物.....	17
约定	18
术语	19
 第 2 章：功能和集成概述	 21
功能概述.....	21
监控功能.....	22
状态管理.....	25
支持的 MIB	25
MIB 扩展	30
纠正性操作.....	30
插件和集成.....	31
Application Insight Module (AIM)	31
CA Virtual Assurance	32
CA Server Automation.....	33
CA Spectrum.....	34
CA eHealth	35
CA NSM	35
 第 3 章：概念	 37
体系结构概述.....	37
SNMP 版本和访问	38
SNMP 陷阱.....	39
SNMP 表格	40
按实例创建监视器.....	41
配置文件.....	41
监控体系结构.....	43
SNMPv2 行状态	44
有状态监控的工作方式.....	45
无状态监控的工作方式.....	46
自主监控.....	48
进程和服务监控.....	53
进程组监控.....	55
日志文件监控.....	56

Windows 事件监控.....	57
历史记录收集.....	58
状态管理模型.....	59
对象状态.....	60
对象合计.....	61
超级聚合.....	62
状态管理和聚合的工作原理.....	63
状态管理陷阱.....	65
状态管理配置选项.....	65
传统模式操作.....	66
传统监视器.....	66
监视器示例.....	67
与 CA Virtual Assurance 集成	67
平台管理模块.....	68
远程配置和部署.....	68
事件引擎.....	69
CA Virtual Assurance 安全选项	70
AIM 体系结构	71
按需模块.....	71
AIM 与 CA Virtual Assurance 的集成.....	72
使用 SystemEDGE 代理的准则.....	72

第 4 章： 安装和部署 75

安装说明.....	75
安装期间的代理配置	76
使用 CA Virtual Assurance 安装	76
在 Windows 系统上安装.....	76
在 Windows 上安装代理.....	77
在 Windows 上从命令行安装代理.....	82
在 UNIX 和 Linux 系统上进行安装.....	88
在 UNIX 和 Linux 系统上安装代理	88
在 UNIX 上从命令行安装代理.....	93
\$CASYSEDGE 变量的传统支持	99
配置和使用响应文件	100
以传统模式安装代理.....	100
代理部署.....	102
使用 CA Virtual Assurance 部署代理	103
卸载 SystemEDGE	105
在 Windows 上卸载 SystemEDGE 和 AIM.....	105
在 UNIX 系统上卸载 SystemEDGE 和 AIM.....	107
重新安装.....	108
升级受管节点和 AIM 服务器	109

代理和 AIM 升级	109
将 SystemEDGE 监视器导入策略	112
 第 5 章：代理配置	 115
配置基础	115
sysedge.cf 和 sysedge.mon 之间的交互	117
使用 CA Virtual Assurance 的配置	117
File-Based 配置	117
配置系统信息	118
配置 SNMP 信息	119
配置身份验证失败陷阱	126
配置 SNMP Set 通知	126
配置 SNMP Set 限制	127
配置 GetBulk 响应消息大小	128
配置联邦信息处理标准模式	129
配置诊断日志记录机制	130
配置维护模式	132
为操作配置支持	132
为子程序配置用户权限和组权限（仅 UNIX）	133
配置 MIB 表限制	133
配置设备状态检查限制	134
配置热启动发现	136
配置 Linux 可用内存计算	136
在多层次结构中配置监视器合计	137
监控配置	139
加载 SystemEDGE AIM	140
有关配置安全的建议	142
监视器和自动监测器的正则表达式示例	142
Perl 兼容的正则表达式 (PCRE) 支持	145
多行正则表达式	146
配置文本模式排除	147
正则表达式示例	147
SNMPv3 配置	150
配置 SNMPv3 引擎 ID	151
配置 SNMPv3 用户信息	151
配置 SNMPv2c 和 SNMPv3 陷阱目标	156
加密 SNMPv3 配置文件	158
禁用 SNMPv1 和 SNMPv2c	158
SystemEDGE Control Panel for Windows	159
 第 6 章：启动代理	 161
在 Windows 系统上启动/停止代理	161

在 UNIX 系统上启动/停止代理	162
UNIX 系统的服务启动脚本	162
命令行启动选项	163
在系统启动时自动启动代理	164
在 Windows 系统上自动启动代理	164
在 UNIX 系统上自动启动代理	164
代理热启动	165
热启动例外	165
维护模式	165

第 7 章： Systems Management Empire MIB 167

Systems Management Empire MIB 概述	168
系统组	169
设备表	169
内核配置组	170
启动配置组	170
数据流组	170
用户表	170
组表	171
进程表	171
更改进程的 nice 值（仅 UNIX）	171
将信号发送到进程	171
用户表	172
远程外壳组	172
运行远程命令	172
性能组	173
进程间通信组	173
删除 IPC 机制	174
缓冲区组	174
消息缓冲区组	174
数据流缓冲区组	174
I/O 缓冲区缓存组	175
目录名称查找缓存组	175
AIX 逻辑分区组	175
工作负荷管理程序 (WLM) 类的表	176
陷阱团体表	176
NT 系统组	176
NT 线程表	177
NT 注册表组	177
NT 服务表	177
NT 性能组	177
NT 事件监视器组	179

NT 注册表和性能扩展组	179
自主监视器表	180
分布式系统组	180
RPC 统计信息组	180
NFS 统计信息组	181
历史记录表	181
日志监视器表	182
磁盘统计信息表	182
为 AIX 系统配置磁盘性能统计信息收集	183
CPU 统计信息表	183
扩展组	184
进程监视器表	184
进程组监视器表	185
聚合状态表	185
LPAR 组	185
镜像表	185
自动监测器表	186
不受支持的 Systems Management Empire MIB 对象	186

第 8 章： 自主监控 187

自主监控概述	187
自主监视器的状态管理	188
受管对象创建	188
聚合状态表	190
自主监视器表	190
自主监视器表列	191
示例自主监视器表条目	197
行创建	198
自主监视器表标志	199
自主监视器表操作参数	201
查看自主监视器表	203
自主监控配置	204
monitor 指令—将条目添加到自主监视器表	205
反向 OID 查找	208
自主监控示例	209
用于自主监控的 edgemon 命令	223
edgemon 示例	226
删除自主监控条目	227

第 9 章： 进程和服务监控 229

进程和服务监控概述	229
Windows 服务监控	230

进程监视器的状态管理.....	231
受管对象创建.....	232
聚合状态表.....	233
进程监视器表.....	233
进程监视器表列.....	234
抽样进程监视器表条目.....	240
进程属性.....	241
行创建对象.....	243
进程监视器表标志.....	245
进程监视器表操作参数.....	250
查看进程监视器表.....	252
进程监控配置.....	252
watch process 指令—将条目添加到进程监视器表.....	254
watch ntservice 指令—将服务监控条目添加到进程监视器表.....	257
进程监控示例.....	259
用于进程监控的 edgwatch 命令.....	265
edgwatch 示例.....	268
删除进程监控条目.....	271
Solaris Zone 进程监控.....	272
有关进程和服务监控的建议.....	273

第 10 章：自动监测器 275

自动监测器的工作方式.....	275
常规自动监测器.....	277
进程和服务自动监测器.....	277
edgwatch 实用工具--使用自动监测器.....	278
用于自动监测器的 edgwatch 命令.....	281
edgwatch 自动监测器示例.....	285
指定 sysedge.cf 中的自动监测器.....	286
常规自动监测器示例.....	289
服务自动监测器示例.....	290
进程自动监测器示例.....	291

第 11 章：进程组监控 293

进程组监控概述.....	293
进程组监视器表.....	293
“进程组监视器”表的各列.....	294
行创建对象.....	298
进程组监视器表标志.....	299
进程组监视器表操作参数.....	301
查看进程组监视器表.....	302
进程组监控配置.....	302

watch procgroup 指令--将条目添加到“进程组监视器”表中	303
watch procgroupex 指令--将条目添加到“进程组监视器”表中（仅 UNIX）	305
进程组监控示例	307
edgwatch 实用工具--监控进程组	308
用于进程组监控的 edgwatch 命令	312
edgwatch 示例	313
删除进程组监控条目	313
Solaris Zone 进程组监控	314

第 12 章：日志文件监控 315

日志文件监控概述	315
目录监控	316
日志监视器表	316
日志监视器表列	316
行创建对象	320
日志监视器表标志	321
日志监视器表操作参数	323
查看日志监视器表	324
日志监控配置	325
watch logfile 指令--将条目添加到日志监视器表	326
日志文件监控示例	327
edgwatch 实用工具--监控日志文件	329
用于日志文件监控的 edgwatch 命令	332
edgwatch 示例	333
删除日志监控条目	335
日志文件监控建议	336
轮转日志文件	338

第 13 章：Windows 事件监控 339

Windows 事件监控概述	339
Windows 事件搜索条件	340
NT 事件监视器表	341
NT 事件监视器表列	341
行创建对象	344
NT 事件监视器表标志	346
NT 事件监视器表操作参数	348
查看 NT 事件监视器表	349
Windows 事件监控配置	350
watch ntevent 指令--将条目添加到 NT 事件监视器表中	351
Windows 事件监控示例	353
edgwatch 实用工具--监控 Windows 事件	354
用于 Windows 事件监控的 edgwatch 命令	357

edgwatch 示例	359
删除 Windows 事件监控条目	360

第 14 章：历史记录收集 363

历史记录收集概述	363
历史记录抽样	363
历史记录控制表和历史记录数据表	364
历史记录控制表的列	364
历史记录表的列	366
行创建对象	366
历史记录采样示例	367
查看历史记录控制表	369
历史记录控制表配置	369
启动期间的初始配置	369
操作期间的动态配置	369
emphistory 指令--将条目添加到历史记录控制表中	370
历史记录收集示例	371
用于管理历史记录控制表中的条目的 emphistory 命令	371

第 15 章：自定义 MIB 对象 375

Systems Management Empire MIB 扩展组	375
扩展组功能	376
扩展变量配置	376
extension 关键字--向扩展组中添加条目	377
其他参数	378
扩展示例	378
扩展脚本	380
示例脚本测试	381
在管理软件中使用扩展变量	382
如何为扩展变量编辑 empire.asn1	383
如何为扩展变量编辑单独的 MIB 规格	383
针对使用扩展的建议	384

第 16 章：Windows 注册表和性能 MIB 对象 385

Windows 注册表和性能功能	385
Systems Management Empire MIB ntRegPerf 组	386
注册表数据	386
性能数据	387
Windows 注册表和性能变量的配置	389
ntregperf 关键字--添加 Windows 注册表和性能 MIB 对象	390
Windows 注册表和性能示例	391

在管理软件中使用 Windows 注册表和性能变量	392
如何为 ntRegPerf 变量编辑 empire.asn1	392
如何为 ntRegPerf 变量添加单独的 MIB 规格	392

第 17 章： SystemEDGE AIM 395

核心 AIM	395
顶级进程 AIM	395
监控维护 Windows AIM	395
启用监视器维护窗口 AIM	396
监视器维护窗口 AIM 的限制	397
性能多维数据集 AIM	397
启用性能多维数据集 AIM	397

第 18 章： 命令行实用工具 399

SystemEDGE 命令行实用工具	399
SystemEDGE 其他命令行实用工具	400
配置 SystemEDGE 命令行实用工具	400

第 19 章： 使用状态管理的示例 401

监控 Windows 任务排定程序	401
监控隔离文件是否存在	403
监控文件系统	403
通过 SNMP 重新配置扩展脚本	405
监控日志文件并通过电子邮件发送通知	409
监控进程	410
创建受监控的对象	410

第 20 章： 故障排除和用法建议 411

使用 diagsysedge.exe	411
使用 diagsysedge 确定代理是否正在运行	411
获取故障排除报告	412
验证 AIM 插件	413
常见问题	413
故障排除的诊断日志记录信息	414
代理未响应 SNMP 请求	414
未接收 SNMP 陷阱消息的管理系统	417
绑定失败：地址已被使用	418
CA SystemEDGE： xtrapmon.exe 实用工具被 edgetrapmon.exe 取代	419

附录 A: FIPS 140-2 加密	421
FIPS 140-2 模式	421
FIPS 库安装	421
平台支持	422
支持的加密协议	422
支持的身份验证协议	422
配置 FIPS 140-2 模式	423
FIPS 模式注意事项	423
密钥保护	424
 附录 B: SystemEDGE 高级加密	 425
SystemEDGE 高级加密	425
支持的平台	425
支持的加密协议	425
支持的身份验证协议	426
FIPS 兼容性	426
安装先决条件	426
安装 SystemEDGE 高级加密	426
在 Windows 上安装 SystemEDGE 高级加密	427
在 Windows 上卸载 SystemEDGE 高级加密	427
在 UNIX 上安装 SystemEDGE 高级加密	428
在 UNIX 上卸载 SystemEDGE 高级加密	428
安装的文件	429
 附录 C: 与其他 SNMP 代理共存	 431
支持多个 SNMP 代理	431
支持多个 SystemEDGE 实例	432
与 Microsoft Windows SNMP 代理共存	432
与 HP SNMP 代理共存	432
与 AIX SNMP 代理共存	433
 附录 D: 主机资源 MIB	 435
主机资源 MIB 概述	436
主机资源系统组	437
主机资源存储组	438
主机资源设备组	438
设备表	439
处理器表	439
磁盘存储表	440
分区表	440

文件系统表.....	441
主机资源运行软件组.....	441
主机资源安装软件组.....	442
不支持的主机资源 MIB 对象.....	442

附录 E：私有企业陷阱 443

陷阱 PDU 的格式.....	443
monitor 陷阱.....	443
monitorEntryNotReady 陷阱.....	444
logMonMatch 陷阱.....	444
logMonNotReady 陷阱.....	445
ntEventMonMatch 陷阱.....	445
ntEventMonNotReady 陷阱.....	446
monitorClear 陷阱.....	446
processStop 陷阱.....	447
processStart 陷阱.....	447
processThreshold 陷阱.....	448
processClear 陷阱.....	448
processNotReady 陷阱.....	449
addrChange 陷阱.....	449
procGroupChange 陷阱.....	450
aggregateState 陷阱.....	450
aggregateAdd 陷阱.....	451
aggregateDelete 陷阱.....	451
SNMPv1 陷阱格式.....	452

索引 453

第 1 章：简介

此部分包含以下主题：

[CA SystemEDGE](#) (p. 17)

CA SystemEDGE

SystemEDGE 通过使系统管理人员能够从单个中央位置控制他们网络上的所有服务器来提高他们的生产力，并且自动化系统管理任务和清单跟踪，从而在帮助减少不断增加的系统支持成本的同时提高生产力和系统稳定性。

SystemEDGE 代理通过符合行业标准的 SNMP 提供强大的系统管理。它使远程管理系统能够访问有关系统配置、状态、性能、用户、过程、文件系统等重要信息。代理还包括智能监控和状态管理功能，通过这些功能可以报告和管理异常，并计算对象重要级别和状态信息。

读者

本指南适用于安装、配置和使用 SystemEDGE 代理以管理 UNIX、Linux 和 Windows 服务器的管理员。它假定您基本了解您系统的操作系统环境以及简单网络管理协议 (SNMP)。

相关出版物

CA Virtual Assurance 文档包括以下交付成果：

管理指南

了解如何管理和使用 CA Virtual Assurance 以在您的环境中管理虚拟资源。

安装指南

包含简短的体系结构信息、各种安装方法、后继安装配置信息以及入门说明。

联机帮助

提供有关窗口详细信息和程序说明，以便使用 CA Virtual Assurance 用户界面。

参考指南

提供有关 AutoShell、CLI 命令、MIB 属性以及性能度量标准的详细信息。

版本说明

提供有关操作系统支持、系统要求、已发布的修正、国际化支持、已知问题以及文档路线图。

服务响应监控用户指南

提供 SRM 的安装和配置详细信息。

SystemEDGE 用户指南

提供 SystemEDGE 的安装和配置详细信息。

SystemEDGE 版本说明

提供有关操作系统支持、系统要求和功能的信息。

约定

本指南使用以下约定：

区分大小写

在本指南中提到的所有类、命令、指令、环境参数、函数和属性的名称均区分大小写，您必须按显示的形式正确拼写它们。- 系统命令和环境变量名称 *可能* 区分大小写，视操作系统的要求而定。-

交叉引用

对其他指南或本指南其他部分中信息的引用采用以下格式：

指南名称

指示其他指南的名称。

“章名称”

表示本指南或其他指南中的章节名称。

同义词

属性、对象和对象标识符 (OID) 等术语与本文档中的术语“变量”是同义的。

SystemEDGE 代理和 CA SystemEDGE 等术语与本文档中的 SystemEDGE 是同义的。

语法

语法和用户输入使用以下格式：

斜体

表示必须为其提供实际值的变量名或占位符。

{a|b}

表示选择强制性操作数（a 或 b）。

[] 或 [[]]

表示可选操作数。

语法示例

以下示例使用这些约定：

```
modify -t ZONE [-m zoneserver] -p psetname {-min mincpu|-max maxcpu} pset
-session ssh
```

操作数 **-min** 和 **-max** 是强制性的，但您可以只使用二者之一，具体视希望定义处理器集中 CPU 的最小数量还是最大数量而定。操作数 **-m** 不是该命令运行所必需的。该命令的所有其他部分必须按显示的形式进行输入。

默认目录

在路径语句中使用的 *CASYSEdge* 表示在其中安装 SystemEDGE 的目录。**默认值：** C:\Program Files\CA\SystemEDGE。

安装路径

在路径语句中使用的 *Install_Path* 表示在其中安装 CA Virtual Assurance 或 CA Virtual Assurance 组件的目录。

默认值：

- Windows x86: C:\Program Files\CA
- Windows x64: C:\CA、C:\Program Files (x86)\CA 或 C:\Program Files\CA
- UNIX、Linux: /opt/CA

术语

MIB 对象、MIB 属性

这两个术语作为同义词使用。

第 2 章：功能和集成概述

此部分包含以下主题：

[功能概述](#) (p. 21)

[插件和集成](#) (p. 31)

功能概述

SystemEDGE 在一个轻量级、易于部署的软件包中提供了以下系统管理功能：

- 收集 SNMP 度量标准以供 SNMP 管理器使用。
- 监控对象，例如度量标准值、进程和服务、进程组、日志文件、目录以及 Windows 事件。这些对象是使用通过 SNMP 陷阱触发通知的可配置 MIB 表予以监控的。
- 使用自主监视器表和进程监视器表管理对象模型状态。这有助于合计受监控对象重要级别、更新对象状态以及发送状态更改陷阱。
- 收集历史记录表和和多维数据集文件中的度量标准以进行基准和趋势分析。
- 基于文件的远程代理配置和远程部署（与 CA Virtual Assurance 或 CA Server Automation 集成时）。
- 可扩展的 MIB，允许您添加注册表项等自定义 MIB 对象进行监控。
- 可以在监控条目超过指定的阈值时触发操作。
- 灵活的体系结构，允许使用作为插件模块的 Application Insight Module (AIM) 实现其他监控功能。

详细信息

[概念](#) (p. 37)

监控功能

您可以将 SystemEDGE 代理配置为监控特定管理信息库 (MIB) 对象的阈值或其他异常。在管理具有数百个系统的大型企业时，您可以配置代理监控的信息。允许配置代理监控信息提供了异常管理功能，这对于在不影响管理开销的情况下维护大规模实施是必需的。

注意：有关每个代理的收集和监控功能的信息，请参阅“*概念*”一章。

支持的监视器的最大分辨率是 64 位。SystemEDGE 代理提供以下类型的监控：

自主监控

提供对代理支持的任何基于整数的 MIB 对象的监控。您可以在自主监视器表中创建条目以指定要监控的对象、比较运算符、阈值以及重要级别，代理会根据创建的条目自动监控对象。代理按照轮询时间间隔监控对象，并根据指定的阈值和重要级别值维护当前状态。如果超过状态更改阈值，代理会发送状态更改陷阱。

注意：有关自主监控的详细信息，请参阅“*自主监控*”一章。

进程和服务监控

提供任何单个进程实例或 Windows 服务的监控。您可以在进程监视器表中创建条目以监控进程或服务是否正在运行，或监控进程表 MIB 对象是否超过指定的阈值。代理监控进程，根据指定的阈值和重要级别值维护当前状态。如果超过状态更改阈值或进程的状态（正在运行或已停止）发生更改，代理将发送状态更改陷阱。

注意：有关进程和服务监控的详细信息，请参阅“*进程和服务监控*”一章。要监控一个以上进程实例，请使用进程组监视器，因为进程监视器仅可以监控单个进程实例。

进程组监控

提供定义一组进程并监控该组的功能。您可以在定义进程组的进程组监视器表中创建条目。代理监控组，并在进程组发生更改时发送陷阱。代理监视器还提供进程组度量标准，如一个组中所有进程所消耗的内存或 CPU 时间。必要时，自主监视器可以监控这些进程组度量标准。

注意：有关进程组监控的详细信息，请参阅“*进程组监控*”一章。

日志文件和目录监控

通过搜索指定为正则表达式的字符串，提供对任何 UTF-8 编码的系统或应用程序日志文件的监控。您可以在日志监视器表中创建条目，代理会监控指定的日志文件中是否存在与用户定义的正则表达式相匹配的行，并在存在匹配时发送陷阱。可以将重要级别与监视器相关联，该关联关系包含在已发送的陷阱中。如果您相应地在日志监视器表中配置条目，还可以监控目录中文件的大小和数量。您可以监控大小最大为 2GB 的日志文件。

注意：有关日志文件和目录监控的详细信息，请参阅“*日志文件监控*”一章。

Windows 事件监控

使用不同的筛选（如事件源）提供对 Windows 事件日志条目的监控。您可以在 NT 事件监视器表中创建条目，代理会监控事件日志文件中是否存在与用户定义的正则表达式相匹配的事件，并在存在匹配时发送陷阱。

注意：有关 Windows 事件监控的详细信息，请参阅“*Windows 事件监控*”一章。

历史记录收集

提供历史数据收集，以便进行管理器端基准制定和趋势分析。您可以在历史记录控制表中创建条目，代理会基于某个时间间隔收集度量标准。收集的度量标准用于提供特定时间间隔内平均系统性能概况。

注意：有关历史记录收集的详细信息，请参阅“*历史记录收集*”一章。

64 位度量标准的监视器比例

您可以为监视器指定阈值来监视 64 位度量标准。例如，如果您指定 64 位阈值，如 9,000,000,000,000,001 字节（ $9 * 10^{18}$ 再加 1）是有效的 64 位值。但是，您可以通过定义比例系数 10^{18} 和阈值 9 来定义 9 EB（百亿亿字节）的阈值。比例系数定义监视器的分辨率。您可以定义 10^{16} 的比例系数，对于一个百分点的分辨率，相应的阈值是 900。

如果监视器的比例系数是 2^{32} ，对于 64 位 (2^{64}) 度量标准，最大监视器值是 2^{32} 。

过程监控增强

SystemEDGE 以新变量（即进程表中的 `processTimePermil`）的百分比形式提供进程 CPU 使用率。进程监控和进程组监控功能通过使用 `processTimePermil` 予以提供。

`processTimePermil` 代表上一抽样时间间隔中进程的 CPU 使用率千分比。该变量值依赖于系统。该值是 CPU 记号数除以总 CPU 记号数的差。由于 SNMP 不支持分数，因此报告的值乘以 1000。例如，CPU 千分比为 505.833，则返回 505833。要转换为百分比，请将返回值除以 10000。

对象合计

系统管理 MIB 中的新合计表用于合计具有多个定义了不同重要级别的监视器的对象。合计表使用自主监视器表和进程监视器表中的新类、实例和属性特性来确定监视器是否指同一对象。总体对象状态是通过对具有最高重要级别的违反监视器发送状态更改陷阱来确定的。

多层对象实例层次结构

要将某个对象实例与其托管的层次结构相关联，使用 `//` 和 `/` 作为分隔符，例如 `//hierarchy/instance`。层次结构可以是多层，以便支持两层以上的对象实例。

层次结构可以是主机名 `//host/instance`，它还可以为 `//<any1>/<any2>/.../<anyN>/<instance>` 形式。对于本地系统上的对象实例，仅可以指定实例，并且会自动在前面添加 `//.`。您可以前置命名空间来识别虚拟对象实例（如 `lpar:`）。例如，LPAR 上的对象实例表示为 `lpar://hmc1/hyper1/lpar1/disk1`。

仅对最低的两层提供对象实例的超级合计，而不是对所有层次。例如，`//host/*` 合计主机上的所有实例。创建的特殊超级合计 `//*`（合计所有对象实例）依赖于选定级别的超级合计。

状态管理

SystemEDGE 代理从创建的自主监视器和进程监视器计算状态信息。监视器具有阈值条件和重要级别（monitor1: “大于 70% 为‘警告’”，monitor2: “大于 90% 为‘严重’”）。将多个监视器组合到具有所有关联的监视器的最坏状态的单个 MonitorObject 中。代理通过状态更改陷阱发送状态信息，CA Virtual Assurance 等状态管理器访问并解释该状态信息，以向您提供企业中系统、进程、服务和应用程序的状态的概况。

注意: 有关 SystemEDGE 状态管理模型的详细信息，请参阅“概念”一章。

支持的 MIB

MIB 充当用于存储代理控制元素的信息的信息库。MIB 是结构化的对象集合，其中每个对象均表示一条管理信息。

管理系统通过读取其 MIB 对象的值来监控受管资源。它也可以通过借助于 SNMP SET 请求修改资源 MIB 中对象的值（设置）来控制资源。

MIB 在 MIB 规格中定义，该规格用于描述与特定资源相关的管理对象。MIB 规格还定义如何规划对象集合的结构。MIB 模块类似于由管理系统和代理使用的数据定义文档。

SystemEDGE 代理支持以下 MIB 中的度量标准集合：

- CA Systems Management Empire MIB
- MIB-II (RFC 1213)
- 主机资源 MIB (RFC 1514)
- IP-MIB（部分）(RFC 4293)
- IF-MIB（部分）(RFC 2233)
- TCP-MIB（部分）(RFC 4022)
- UDP-MIB（部分）(RFC 4113)

代理可以通过 SNMP 陷阱使用其灵活监控概念，将从这些 MIB 收集的信息发送至接受 SNMP 数据的任何管理系统。

CA Systems Management Empire MIB

Systems Management Empire MIB 是专用的企业 MIB，包括用于监控基础系统及其应用程序的运行状况和性能的对象。该 MIB 定义了以下对象的管理信息：

Systems Management Empire MIB

- 系统详细信息
- 已安装设备
- 内核和系统参数
- 引导配置
- 网络、数据流和 I/O 缓冲区统计信息
- 网络文件系统 (NFS) 和远程过程调用 (RPC) 统计信息
- 内核性能统计信息，如上下文切换数和页面错误数
- 文件系统
- 用户和组
- 流程
- 进程间通信
- 系统资源
- 逻辑分区
- 磁盘和 CPU 统计信息
- 特定于 Windows 的统计信息，如注册表和 Windows 事件日志

Systems Management Empire MIB 包含以下表，用于配置代理的自治监控和数据存储功能：

- 自主监视器表
- 进程监视器表
- 进程组监视器表
- 日志监视器表
- NT 事件监视器表
- 历史记录控制表

Systems Management Empire MIB 还包含一些表，用于提供来自自己定义监视器和进程监视器的有关受监控对象状态的信息。

有关 SystemEDGE 代理如何使用 Systems Management Empire MIB 的详细信息，请参阅“概念”和“Systems Management Empire MIB”这两章。

注意：Systems Management Empire MIB 对象支持因平台而异。有关每个平台上不支持的 Systems Management Empire MIB 对象的列表，请参阅《SystemEDGE 版本说明》。

MIB-II

MIB-II 是提供有关网络接口和协议统计信息的信息的标准 (RFC 1213)。该 MIB 包括有关以下组的信息：

- 系统
- 接口
- 地址转换
- Internet 协议 (IP)
注意：SystemEDGE 仅支持用于 IF-MIB 文件的 ifXTable。
- Internet 控制消息协议 (ICMP)
- 传输控制协议 (TCP)
- 用户数据报协议 (UDP)
- 简单网络管理协议 (SNMP)

注意：在某些 Windows 版本上，如果 Wireless Zero Configuration 服务处于活动状态，则 SystemEDGE 可能无法从系统中检索 ifXTable 数据。该服务以独占模式使用 NDISUIO 驱动程序，因此 SystemEDGE 可能无法访问 NDISUIO 驱动程序。这种情况不会发生在 Window 2008、Windows Vista 或 Windows 7 上。

如果需要早期 Windows 版本上的 ifXTable 数据，则必须禁用 Wireless Zero Configuration 服务或任何使用 ndisuios.sys 的服务。

MIB-II 对象支持因平台而异。有关每个平台上不支持的 MIB-II 对象的列表，请参阅《SystemEDGE 版本说明》。

主机资源 MIB

主机资源 MIB 由 Internet 工程任务组 (IETF) 定义, 为常规主机系统提供管理信息。MIB 包括对资产管理尤其有用的信息, 例如:

- 存储区域, 如文件系统和磁盘分区
- 正在运行的已安装软件
- 系统设备, 如键盘、磁盘和网卡

您可以将 SystemEDGE 与主机资源 MIB 结合使用, 以使资产跟踪自动化并提供已安装硬件和软件的当前情形。SystemEDGE 可以确定您的系统是否已正确配置, 以及是否包含最新的修补程序和服务包。该信息可以帮助简化系统管理、改进性能并且降低安全风险。

有关 SystemEDGE 代理如何使用主机资源 MIB 的详细信息, 请参阅 *主机资源 MIB 附录*。

注意: 主机资源 MIB 对象支持因平台而异。有关每个平台上不支持的主机资源 MIB 对象的列表, 请参阅 《SystemEDGE 版本说明》。

IP-MIB 表格

SystemEDGE 支持以下 IP MIB 表:

MIB 表	Windows	Solaris	HP-UX	AIX	Linux
ipSystemStatsTable (RFC 4293)	否	是	否	是	是
ipIfStatsTable (RFC 4293)	否	否	是	否	否
ipAddressPrefixTable (RFC 4293)	否	否	是	否	是
ipNetToPhysicalTable (RFC 4293)	否	是	是	否	否
ipAddressTable (RFC 4293)	否	是	是	是	是
ipDefaultRouterTable (RFC 4293)	否	是	是	是	是
icmpStatsTable (RFC 4293)	是	是	否	是	是
tcpConnectionTable (RFC 4022)	是	是	是	是	是
udpEndpointTable (RFC 4113)	是	是	是	是	是

注意: 支持的 IP MIB 对象因平台而异。有关每个平台上不支持的 IP MIB 对象的列表, 请参阅 《SystemEDGE 版本说明》。

MIB 扩展

Systems Management Empire MIB 是用户可扩展的。您可以将扩展添加到 MIB，以便管理基础 MIB 中未包括但对企业非常重要的对象。您可以通过以下方法自定义 Systems Management Empire MIB：

- 创建自己的标量 MIB 对象以进行自定义管理。查询或设置新的 MIB 变量时，您可以通过每个自定义的 MIB 对象编号、MIB 对象类型以及要运行的脚本或程序的名称来配置 SystemEDGE。
- 创建自定义 MIB 对象以管理 Windows 注册表和性能数据。您可以将 SystemEDGE 配置为报告其他注册表参数和性能数据，而无需使用外部程序或脚本。使用该功能，您可以通过性能注册表监控使性能数据可用的应用程序的运行状况。

注意：有关创建自定义 MIB 对象的详细信息，请参阅 *自定义 MIB 对象* 一章。有关创建对象以管理 Windows 注册表和性能数据的详细信息，请参阅 *Windows 注册表和性能 MIB 对象* 一章。

纠正性操作

SystemEDGE 代理可通过调用操作（满足特定条件时运行的特定命令或脚本）来自动响应系统上的紧急情况。例如，可以指定操作，以便使代理能够重新启动失败的进程，在对系统进行未经授权的访问时向管理员发送电子邮件等。

在使用操作时，必须（使用任何参数）指定命令的完整路径。支持使用环境变量。每当条件满足监控表条目时，代理都会运行该命令。

有关指定操作的详细信息和详细示例，请参阅以下各章：

- “自主监控”
- “进程和服务监控”
- “进程组监控”
- “日志文件监控”
- “Windows 事件监控”

插件和集成

SystemEDGE 代理的 SNMP 体系结构支持与所有基于 SNMP 的管理器的集成。在不使用任何特定管理产品的情况下，均可对该代理进行配置，并且该代理可将陷阱发送到任何收集 SNMP 信息的应用程序。许多 CA 产品以与代理的增强集成为特色，以利用其各种功能并提供用于解释其数据以及远程配置和部署代理的接口。

对于特定的管理规程，CA 提供名为 “Application Insight Module (AIM)” 的 SystemEDGE 插件模块。AIM 集成到代理的现有体系结构中，并且通过提供监控特定应用程序或服务的工具来扩展自身的功能。

本节讨论了以与 SystemEDGE 集成为特色的 CA 产品，以及可用作代理插件的 AIM。

Application Insight Module (AIM)

SystemEDGE 代理提供了插件体系结构，代理可在初始化时通过该体系结构加载可选的 Application Insight Module (AIM)。这些 AIM 提供了可扩展的灵活方法，用于支持特定于应用程序的语义知识。

以下 AIM 符合 SystemEDGE 状态管理模型，并支持通过管理器用户界面进行部署和配置：

VMware vCenter 服务器 AIM

提供受 VMware vCenter 服务器控制的系统的管理功能。

远程监控 AIM

提供远程（无代理）系统的管理和监控功能。

服务响应监视器 (SRM)

监视受管节点上的服务响应。服务响应监视器 (SRM) AIM 创建了使用 SystemEDGE 状态管理模型的监视器，并支持通过 CA Virtual Assurance 用户界面进行部署和配置。

Cisco Unified Computing System (UCS)

收集 Cisco UCS Manager 中的设备和统计数据，并为外部应用程序提供 SNMP 支持。

LPAR AIM (AIX PowerVM)

提供管理和监控整个系统（包括 LPAR）的功能。

Microsoft 群集服务器 AIM

从远程系统监控 Microsoft 群集。MSCS 监控需要远程访问群集和用于度量标准收集（如 CPU 和内存使用率）的各个群集节点。

Microsoft Hyper-V AIM

提供管理和监控 Hyper-V 服务器的功能。

Exchange Server 和 Active Directory AIM

提供监控外部和内部基础架构上的 Exchange Server 和 Active Directory 的功能。不适用于 CA Server Automation。

Solaris Zones AIM

提供管理和监控配置为运行容器和区域的 Solaris 10 系统的功能。

KVM AIM

从远程系统监控基于内核的虚拟机环境。KVM AIM 与一个或多个 RHEV 管理器进行通信。不适用于 CA Server Automation。

Citrix XenServer AIM

监控 Citrix XenServer 环境。AIM 仅与池主服务器进行通信。

CA Virtual Assurance

CA Virtual Assurance 是一种基于策略的产品，可自动地监控、重新配置和开通虚拟资源，以便动态满足面向服务的复杂数据中心的负载需求。CA Virtual Assurance 构建于面向服务的体系结构 (SOA) 之上，可持续分析数据中心，帮助确保以最佳方式开通服务器，从而执行所需任务。

CA Virtual Assurance 使用 SystemEDGE 作为监控和管理各种系统的基础。

仅当将代理与 CA Virtual Assurance 用户界面结合使用时，特定的功能才可用。将 SystemEDGE 与这些产品结合使用时可提供以下独特功能：

- 通过平台管理模块查看代理的状态管理数据。您可以查看所有监视器的状态，以及查找和诊断导致状态传播的关键监视器。
- 将代理部署到远程系统。该功能包括：配置诸如 SNMP 社区的设置和其他任何安装设置，以及将具有这些设置的代理部署到任意受管系统。
- 为远程系统配置代理。CA Virtual Assurance 以基于文件的配置解决方案为特色，通过该解决方案，您可以创建监视器条目，配置 SystemEDGE 代理的任意设置，以及将该配置部署到任意多个受管系统。
- 代理热启动。将代理部署到远程系统时，大多数情况下并不需要完全重新启动代理。代理在运行时期间执行热启动并实施配置更改。
- 配置锁定选项（例如在对任意受管代理进行基于 SNMP 的更改时收到通知的功能）和覆盖不通过 CA Virtual Assurance 启动的所有更改的选项。

- 服务响应监视器 AIM 的部署和配置。
- 通过选择各种虚拟环境监控 AIM 插件来执行虚拟平台管理。

CA Virtual Assurance 增强了 SystemEDGE 代理的功能，建议使用这些产品来通过 SystemEDGE 管理环境中所有的物理系统和虚拟系统。

有关 CA Virtual Assurance 的详细信息，请参阅 *CA Virtual Assurance* 文档。

CA Server Automation

CA Server Automation 是一种基于策略的产品，可自动地监控、重新配置和开通物理资源及虚拟资源，以便动态满足面向服务的复杂数据中心的负载需求。CA Server Automation 建立在面向服务的体系结构 (SOA) 的基础上，持续分析您的数据中心，以帮助确保最佳开通服务器来执行必要的任务。

CA Server Automation 包含嵌入式 CA Virtual Assurance 功能，其中包括 SystemEDGE 代理。使用 CA Server Automation 中的 SystemEDGE，可以查看数据中心的以系统为中心的管理视图。通过 SystemEDGE 所提供的信息，可以对自动化数据中心内的物理系统和虚拟系统作出更明智的决定。将代理与 CA Virtual Assurance 功能结合后，可以通过 CA Virtual Assurance AIM 管理虚拟系统和物理系统。

CA Virtual Assurance 平台管理模块从 SystemEDGE 代理收集信息，以供在 CA Server Automation 用户界面上显示。CA Server Automation 用户界面提供了所有物理系统和虚拟系统在多个不同上下文中的拓扑视图。选择该视图中受 SystemEDGE 管理的系统将显示受监控对象的状态摘要、已定义的监视器以及各种系统度量标准。用户界面还提供了用于执行虚拟监控 AIM 所允许的系统操作的选项。您有权访问 CA Server Automation 中的全套 CA Virtual Assurance 功能，还可以查看 CA Server Automation 事件视图中的 SystemEDGE 陷阱。

CA Spectrum

CA Spectrum 支持管理通过 SystemEDGE 代理发现的系统。通过使用 CA Spectrum 管理 SystemEDGE，可以在分布式网络的上下文中管理系统。代理可以提醒 CA Spectrum 注意以下内容：性能不佳的系统度量标准、停止的进程或可能影响网络操作的应用程序。

可以从 Spectrum OneClick 界面查看 SystemEDGE 代理收集的 SNMP 度量标准以及自主监控表中的条目。CA Spectrum 可以收集代理发送的陷阱并将其显示为事件，以便您可以管理来自 OneClick 的异常。您可以使用规则（这些规则会在收集特定 SystemEDGE 陷阱时生成报警）配置 CA Spectrum 报警视图，以便可以按解决网络故障的相同方法发现和解决特定的系统问题。

CA Spectrum 9.2 支持启用状态管理功能的 SystemEDGE r5.0 Systems Management Empire MIB。通过使用 SystemEDGE 5.0 提供的嵌入式 OneClick 表，您可以添加、编辑和删除监视器。CA Spectrum r9.2 动态检查 SystemEDGE 代理的（最新或先前）版本并显示特定于版本的信息。

CA Spectrum 也支持 SystemEDGE AIM。在 CA Spectrum 9.2 中接收 AIM 陷阱时，陷阱将包括代理类型、代理名称和警报类型。

注意：将 SystemEDGE 代理与 OneClick 结合使用时，特定于 CA Virtual Assurance 的多个功能不可用。有关这些功能的详细信息，请参阅 *概念一章*。

要启用代理的完整功能，必须使用 CA Virtual Assurance 管理代理。有关详细信息，请参阅 *CA Virtual Assurance 文档*。

CA eHealth

可以运行 `nhAddSysEdgeMonEntries` 命令来配置 SystemEDGE 代理，以便监控关键的 CA eHealth 进程和系统日志。命令将条目添加到 `sysedge.cf` 文件中，并且停止和重新启动 SystemEDGE 代理以实施更改。

注意：有关 `nhAddSysEdgeMonEntries` 命令的详细信息，请参阅《CA eHealth 管理参考》。

将 SystemEDGE 与 CA eHealth 结合使用时，CA eHealth 轮询器可以从 SystemEDGE 代理收集数据，并将数据存储在 CA eHealth 数据库中，以供 CA eHealth 报告和实时监控工具使用。您可以运行 At-a-Glance (AAG)、Trend、Top N、What-If Capacity Trend、System Health 和 MyHealth 报告，以供系统执行容量规划、准确记录服务问题和解决潜在的问题。

注意：有关 CA eHealth 应用程序的详细信息，请参阅《CA eHealth 系统和应用程序管理指南》。

CA eHealth 支持使用传统监视器（即不使用 SystemEDGE r5.0 功能的监视器）运行 SystemEDGE 代理。将代理与 CA eHealth 和 eHealth AdvantEDGE 视图结合使用时，多个特定于 CA Virtual Assurance 的功能不可用。有关传统监视器和传统模式（在不使用 CA Virtual Assurance 的情况下运行代理）的详细信息，请参阅“概念”一章。要启用代理的完整功能，必须使用 CA Virtual Assurance 管理代理。有关详细信息，请参阅 *CA Virtual Assurance* 文档。

CA NSM

CA NSM 可以与 SystemEDGE 代理集成，以使用代理数据来管理所发现的系统。将 SystemEDGE 代理管理功能与 CA NSM DSM 功能相结合可以提供分层系统监控体系结构，其中每个代理监控唯一的受管对象。

CA NSM 包含 DSM 类和 SystemEDGE 代理的策略，并可以发现和轮询代理来获取系统信息。每当显示对象表条目时，就会创建 DSM 监控的对象，并且 CA NSM 可以处理从对象表发送的陷阱。通过 SystemEDGE 代理视图，您可以查看代理的摘要，以及在 CA NSM 内的“自主监视器”和“进程监视器”表中创建条目。

从 CA NSM r11.2 SP1 开始，CA NSM 便可解释由 SystemEDGE 代理和 SRM AIM 的 SystemEDGE 状态管理模型创建和维护的状态。将 CA NSM 与 CA Virtual Assurance 集成时，您可以通过管理命令中心的上下文链接访问 CA Virtual Assurance 用户界面，从而访问远程代理部署和配置功能。

注意：有关将 CA NSM 与 CA Virtual Assurance 集成的详细信息，请参阅 *CA Virtual Assurance* 文档。

第 3 章： 概念

此部分包含以下主题：

[体系结构概述](#) (p. 37)

[监控体系结构](#) (p. 43)

[状态管理模型](#) (p. 59)

[与 CA Virtual Assurance 集成](#) (p. 67)

[AIM 体系结构](#) (p. 71)

[使用 SystemEDGE 代理的准则](#) (p. 72)

体系结构概述

SystemEDGE 代理使用轻量级的灵活 SNMP 体系结构监控系统。SNMP 是管理基于 TCP/IP 的网络和设备标准。在最基本的级别，SystemEDGE 在 SNMP 框架内执行代理角色：它在受管对象上执行基于 SNMP 的指令。指令可能来自管理系统或代理本身，这些指令可能是关于对象信息的请求或修改系统参数的请求。SystemEDGE 代理执行这些请求并向请求的应用程序发送回复。代理按照标准 SNMP 协议交换信息，因此可通过 SNMP 通信的任何管理器都可以与代理集成。

SystemEDGE 代理通过 MIB 属性访问系统度量标准。MIB 是包含遵守通用属性格式的管理信息的存储库。代理可以收集或修改其支持和维护的所有 MIB 中的属性。

注意：有关支持的 MIB 的详细信息，请参阅“简介”一章。

默认情况下，SystemEDGE 代理在 UDP 端口 161 上通过 SNMP 通信（可以在安装过程中更改该端口）。在 SNMPv1 环境中，它使用以下标准 SNMP 消息类型来交换管理信息：

GetRequest

从 MIB 获取特定对象的值。请求关于系统的特定信息时，管理系统将该消息发送给代理。

GetNextRequest

获取下一个对象实例。

GetResponse

将请求的 MIB 对象信息返回至请求应用程序。

SetRequest

指示代理更改 MIB 对象的值

Trap

通知管理系统出现异常。陷阱允许代理将未经请求的消息发送至管理器。有关详细信息，请参阅 [SNMP 陷阱](#) (p. 39)。

在 SNMPv2 环境中，交换管理信息的标准 SNMP 消息类型如下：

GetBulkRequest

获取一个或多个 MIB 对象的值（表）。这类似于 GetNextRequest。

此通用 SNMP 工作流（获取或设置 MIB 属性 SNMP 回复的请求、检索或修改）包含全部 SystemEDGE 功能的基础。但是，代理可以超出基本 SNMP 集合，进行智能监控和状态管理。有关详细信息，请参阅[监控系统结构](#) (p. 43)和[状态管理模型](#) (p. 59)。

SNMP 版本和访问

SystemEDGE 支持 SNMPv1、SNMPv2c 和 SNMPv3 协议。每份协议包含唯一特征，它们的主要区别在于消息安全性。代理可以使用任何协议的安全访问方法传输 SNMP 消息。

SNMPv1 和 SNMPv2c 社区

SNMPv1 和 SNMPv2c 使用社区在系统之间传输消息。社区是代理和大量管理系统之间的关系，它定义分布式通信的身份验证和访问控制权限。社区通过社区名称进行标识，其名称为显示在每个 SNMP 消息的标头部分中的八进制字符串。

代理检查 SNMP 消息标头中的社区名称以确定消息是否真实，然后处理消息。如果团体名称与代理所接受的名称不匹配，代理将记录身份验证失败并丢弃该消息。

SystemEDGE 代理仅安装有默认定义的名为 **public** 的团体。该团体对任何管理器系统的代理的 MIB 对象值提供只读访问权限。在可以修改 MIB 值之前，您必须在安装期间或安装之后配置读写团体字符串。您可以将 IP 地址列表添加到团体定义中以限制对特定系统的访问。

注意：有关配置 SNMP 团体的详细信息，请参阅“代理配置”一章。

SNMPv3 用户和安全

SNMPv3 在网络中使用身份验证和加密的组合以提供安全设备访问：

身份验证

帮助确保数据包在传输期间不会被更改。

加密

加密软件包内容以防止未经授权的源进行查看。

SNMPv3 使用安全模型和安全级别。安全模型是为用户及该用户所在的组设置的身份验证策略。安全级别是安全模型中允许的安全级别。安全模型和安全级别的组合确定了处理 SNMP 数据包的安全机制。该组合可以确保从 SNMP 设备安全地收集数据，并加密机密信息以防止在网络传输中发生泄漏。

SystemEDGE 代理使用基于用户的安全模型 (USM) 进行基于 SNMP 的通信。您可以将 SystemEDGE 代理配置为使用指定的 SNMPv3 用户、身份验证和加密协议以及密码。有关启用和配置 SNMPv3 的详细信息，请参阅“代理配置”一章。

SNMP 陷阱

SystemEDGE 可以发送 SNMPv1 陷阱，SNMPv2c、SNMPv3 陷阱和通知。当已配置的阈值超过受监控对象更改的聚合状态，或者满足其他指定的条件（例如日志监视器匹配）时，代理将发送状态更改陷阱。

SNMP 陷阱配置包括为 SNMPv1 和 SNMPv2c 陷阱配置陷阱团体（包括目标），以及为 SNMPv3 陷阱配置 SNMPv3 用户。陷阱目标指出哪些管理系统接收代理生成的陷阱消息。代理的监控功能通过向陷阱团体中的管理系统发送陷阱消息报告异常情况。

注意：有关配置 SystemEDGE 代理以发送陷阱的信息，请参阅“代理配置”一章。有关代理可发送的陷阱类型的详细信息，请参阅[监控系统结构](#) (p. 43)。

SystemEDGE 代理将陷阱配置信息存储在 Systems Management Empire MIB 的只读 trapCommunity 表中。有关该表的详细信息（包括如果不希望显示此访问信息，如何禁用），请参阅“代理配置”一章。有关在管理系统中查看该表的详细信息，请参阅 *CA Virtual Assurance* 和 *CA eHealth* 文档。

SNMP 表格

SystemEDGE 代理使用表代表“多实例”项。这些表中的大多数都使用整数值作为表索引。索引非常重要，它可唯一识别表中的项，并在代理重新启动后永久存在。

例如，自主监视器表使用整数索引。以下命令（在 `sysedge.cf` 文件中）为 `devTable` 创建监视器以了解磁盘驱动器的空间使用情况。`index(99)` 与该特定监视器实例关联：

```
monitor devTableEntry 'C:' devCapacity 99 0x100 30 absolute >= 95 'exempl'
```

使用索引值 99，SNMP 查询可以获得监视器条目的当前值。可以使用 SystemEDGE `snmpget` 命令查询 OID `monCurrVal.99`（符号表示法），如下所示：

```
snmpget -h <host> -c public -o 1.3.6.1.4.1.546.6.1.1.6.99
```

还可以使用符号表示法查询 OID `monCurrVal.99`（符号表示法）：

```
snmpget -h <host> -c public -o monCurrVal.99
```


按实例创建监视器

在表格 MIB 对象上创建自主监视器时，您需要知道受监控实例的索引。对于一些重要的 MIB 表，代理提供了查找功能（已在上面的示例中使用）。该功能启用了以下两个选项：

- 使用 `sysedge.cf` 配置文件

例如，虽然 `devTable` 具有整数索引，但通过查找功能可以使用驱动器盘符为 `devTableEntry` 定义监视器。`SystemEDGE` 在 `devTable` 条目中搜索驱动器盘符的匹配项。如果 `devTableEntry` 的索引条目为 3，则监视器的对象标识符将设置为该索引 (`devCapacity.3`)。

- 使用 SNMP

在监视器的 MIB 属性 `monObjClass`、`monObjInstance`、`monObjAttribute` 中提供了表名称、实例名称和属性名称时，`SystemEDGE` 可确定监视器的对象标识符。

以下示例创建了索引为 111 的监视器条目，可使用提供的属性为受监控的 OID 自动派生正确的 `devTable` 索引。监视器实际上是在设置行状态时创建的。

```
set monObjClass.111 = "devTableEntry"
set monObjInstance.111 = "C:"
set monObjAttribute.111 = "devCapacity"
set monRowStatus.111 = active(1)
```

注意：您可以使用 `SystemEDGE` 代理附带的 `snmpset` 命令设置 SNMP 值。有关详细信息，请参阅“*命令行实用工具*”一章。

只有 `mirrorMonitorTable` 和 `mirrorAggregateTable` 两个代理表不是使用整数值编制索引，它们使用对象类、实例和属性值作为索引，以实现特定受管对象的访问。

配置文件

`SystemEDGE` 代理基于以下配置文本文件的内容执行操作：

- `sysedge.cf`
- `sysedgeV3.cf`

代理安装在 `CASYSEDGE\config` 中创建了 `sysedge.cf` 文件，其中包含安装期间定义的基本信息，例如系统信息、SNMP 团体、CA Virtual Assurance 管理器注册以及数据目录（在本文档中称为 `CASYSEDGE_DATA`）。

SystemEDGE 代理启动时，它将读取在安装期间创建的文件以确定它应如何操作，并且在 CASYSEDGE_DATA 的子目录中创建了 sysedge.cf 文件的副本。

使用该文件版本进行后继安装运行时配置更改。您可以编辑在安装期间定义的配置设置以及定义其他设置。

注意：有关安装和数据目录位置的详细信息，请参阅“代理配置”一章。本文档将安装目录称为 CASYSEDGE。

sysedgeV3.cf 文件包含 SNMPv2c 和 SNMPv3 配置信息。您不能在安装期间配置 SNMPv2c 和 SNMPv3，但可以将 sysedgeV3.cf 文件（在最初启动代理时，被复制到数据目录的子目录）修改到数据目录，以启用 SNMPv3 操作并定义 SNMPv3 用户、访问列表和安全协议。我们建议您将 sysedgeV3.cf 保存在 cental 配置存储库中，或（充分利用该文件的自动部署功能）使用 CA Virtual Assurance 基于策略的配置功能。

注意：有关使用 sysedge.cf 和 sysedgeV3.cf 配置代理的基本属性和 SNMPv3 设置的详细信息，请参阅“代理配置”一章。

sysedge.cf 文件还控制代理的监控设置。除了提交 SNMP Set 请求之外，其他任何配置监控条目的方法（通过诸如 CA Virtual Assurance 等用户界面或通过直接文件操作）都最终使用监控信息填充数据目录中的 sysedge.cf 文件。代理使用该文件收集执行所有监控操作的说明。

注意：有关代理的监控功能的详细信息，请参阅[监控体系结构](#) (p. 43)。

sysedge.mon 文件可用作备份文件，以存储通过 SNMP Set 所做的那些未在 sysedge.cf 文件中自动输入的更改。该文件还存储运行时信息，如监视器的当前状态。sysedge.cf 中的条目优先于 sysedge.mon 中的条目。代理执行完全（或冷）重新启动时，它首先针对监控条目读取数据目录中的 sysedge.mon 文件，然后读取 sysedge.cf 文件，覆盖现有监视器并添加其他监视器。代理执行部分（或热）启动时，它仅读取 sysedge.cf 文件。使用 sysedge.cf 文件配置代理，可通过 CA Virtual Assurance 用户界面中的配置或直接操作进行配置。

监控体系结构

除基本 SNMP 度量标准收集之外，SystemEDGE 代理监控功能还进行了扩展以实现智能监控和异常报告两项功能。您可以控制代理监控的内容以及代理将陷阱返回到管理系统的时间。

代理在 Systems Management Empire MIB 的监视器表中维护其监控数据。在一个监视器表中创建监视器条目将指示代理读取该表并对该条目执行操作。相应地，代理每隔一段时间对系统进行轮询，如果受监控对象的返回值与特定条件（例如，阈值、进程状态或正则表达式）匹配，代理会向定义为陷阱目标的所有系统发送陷阱。

Systems Management Empire MIB 包含以下监控表：

自主监视器表 (monitorTable)

包含要监控和与用户指定的阈值进行比较以控制重要级别和结果对象状态的 MIB 对象。

进程监视器表 (processMonTable)

包含要针对状态（即，它们是否正在运行）和资源使用情况进行监控的进程。

进程组监视器表 (processGroupMonTable)

包含要针对状态（即，是否发生更改）进行监控的进程组。

日志监视器表 (logMonitorTable)

包含要针对指定的正则表达式字符串进行监控的日志文件和目录。

NT 事件监视器表 (ntEventMonTable)

包含在其中搜索特定事件的事件日志。

历史记录控制表 (empireHistoryCtrlTable)

包含代理在“历史记录”表中监控和存储的 MIB 对象的抽样间隔和抽样数目，以供管理系统将来检索使用。

每个表都包含创建条目时为其指定值的 MIB 属性。已对表编制索引，因此每个条目都包含唯一属性值。代理读取这些表，条目属性为代理提供有关要监控哪些系统属性以及何时报告异常的说明。

在本地 sysedge.cf 和 sysedge.mon 文件中维护所有监视器表条目。只是出于兼容性原因才使用 sysedge.mon 文件。通过指令，您可以在 sysedge.cf 文件中直接输入条目，也可以在与代理集成的管理产品中配置条目。将代理与 CA Virtual Assurance 结合使用时，您可以创建监视器条目，并使用这些条目将文件部署到一个或多个受管系统上的代理。

SNMPv2 行状态

SystemEDGE 使用 SNMPv2 引入的行状态概念管理任何监控 MIB 表中的行（或条目）的创建、激活和维护。RowStatus 文本约定管理行的创建和删除，并作为概念行、状态列中 SYNTAX 子句的值：

RowStatus ::= TEXTUAL-CONVENTION

STATUS current

在其中一个监控表中定义的每个条目都成为该表中的一行，而且每一个都具有行状态值。行状态确定该行是否处于活动状态且具有可访问数据。行状态的可用值如下所示：

活动

表示行可供受管设备使用。只有监视器条目的行状态为 **active**，代理才能实施条目中的监控说明。默认情况下，使用有效信息创建条目时，行状态将自动设置为 **active**。

notInService

表示行存在于代理中，但不能用于受管设备。

notReady

表示行存在于代理中，但缺少必要的信息而无法用于受管设备。

createAndGo

管理站提供，可用来创建一个行并使其可供受管设备使用。

createAndWait

管理站提供，可用来创建一个概念行实例但不能使其可供受管设备使用。

销毁

管理站提供，可用来删除一行。

您可以在创建行期间提供以上任何状态，但 **notReady** 和 **destroy** 除外。您可以使用 **destroy** 状态删除现有的行。只返回以下三个值以响应管理器 GET 请求：

活动

表示行可供受管设备使用。

notInService

表示行不可用于受管设备，尽管代理具有足够的信息使其可用。

notReady

表示行不可用于受管设备，因为代理缺少足够的信息。

注意：有关控制监视器条目的行状态的详细信息，请参阅本指南中的监控章节。有关 RowStatus 文本约定的详细信息，请参阅 RFC 1443: SNMPv2 的文本约定。

有状态监控的工作方式

有状态监控使用定义的重要级别信息和对象模型为由一个或一组监视器代表的每个对象分配状态。下面的表支持有状态监控：

- [自主监视器](#) (p. 48)
- [进程监视器](#) (p. 53)

在“自主监视器”表中创建条目时，您可以指定要针对特定阈值与重要级别和对象信息进行监控的 MIB 对象 (OID)。在“进程监视器”表中创建条目时，您可以指定要针对其运行状态或进程属性阈值进行监控的进程。该进程介绍了代理如何处理具有重要级别信息的有状态监视器。处理传统监视器（没有重要级别）时，代理会以不同的方式进行操作。

注意：有关自主监控和进程监控的特定信息，请参阅“自主监控”和“进程监控”章节中每个监控类型对应的特定部分。

SystemEDGE 代理按如下所示实施有状态监控说明：

1. 您可以在相应的 Systems Management Empire MIB 表中创建条目，以针对指定的组更改、正则表达式或事件监控进程组、日志文件或 Windows 事件日志。
除非使用 CA Virtual Assurance，否则需要重新启动代理才能识别对 sysedge.cf 文件所做的更改。如果通过 CA Virtual Assurance 创建条目并部署更改，代理将执行热启动并在运行时开始读取条目。
2. （仅进程监控）代理使用在条目中定义的正则表达式确定要监控的进程 ID（或 Windows 服务索引）。

3. 代理根据指定的轮询时间间隔轮询指定的 MIB 属性 (OID) 或进程属性。
4. 当属性超过阈值时，代理将设置监视器条目的当前状态。将当前状态设置为条目的重要级别列中定义的值。

注意：代理聚合具有相同对象信息以及 `monObjClass`、`monObjInstance` 和 `monObjAttribute` 的相同值的监视器状态。有关监视器聚合的详细信息，请参阅“状态管理和聚合的工作方式”。

5. 代理向陷阱团体中的所有系统发送状态更改陷阱。陷阱指定对象信息、超出阈值限制和当前对象状态（所有聚合监视器的最差状态）。
6. 代理执行表条目（最差受监控对象）中定义的任何操作。例如，如果您对已停止的进程定义了进程重新启动操作，代理将重新启动该进程。
7. 代理继续轮询属性，如果阈值条件消失，代理将更新监视器的当前状态值并发送状态更改陷阱（如果最差状态更改）。

有关对具有传统监视器的代理进行操作的详细信息，请参阅[传统监视器](#) (p. 66)。

无状态监控的工作方式

无状态监视器不获取对象状态信息，也不使用对象模型保持总体对象状态。这些监视器确实维护着一个重要级别值，但该重要级别用于跟踪单个监视器的重要性，而并非用于计算对象状态。下列各表支持无状态监控：

- [进程组监视器](#) (p. 55)
- [日志文件监视器](#) (p. 56)
- [NT 事件监视器](#) (p. 57)

在“进程组监视器”表中创建条目时，您可以指定要针对更改进行监控以及要计算组统计信息的一组进程。在“日志文件监视器”表中创建条目时，您可以指定要针对正则表达式进行监控的日志文件。在“日志文件监视器”表中，您可以另外指定要计算条目数和大小的目录。在 NT 事件监视器表中创建条目时，您可以指定正则表达式，以针对特定类型的事件监控特定的 Windows 事件日志。

注意：有关进程组、日志文件和 Windows 事件监控的特定信息，请参阅每个监控类型对应的特定部分。

SystemEDGE 代理按如下所示实施无状态监控说明：

1. 您可以在相应的 **Systems Management Empire MIB** 表中创建条目，以针对指定的组更改、正则表达式或事件监控进程组、日志文件或 **Windows** 事件日志。

注意：有关创建无状态监视器条目的详细信息，请参阅“进程组监控”、“日志文件监控”和“**Windows** 事件监控”等章节。

除非使用 **CA Virtual Assurance**，否则需要重新启动代理才能识别对 **sysedge.cf** 文件所做的更改。如果通过 **CA Virtual Assurance** 创建条目并部署更改，代理将执行热启动并在运行时开始读取条目。

2. 根据指定的轮询时间间隔，代理读取条目，并在系统、指定的日志文件或指定的 **Windows** 事件日志中搜索与指定的正则表达式相匹配的进程、条目或事件。
3. （仅进程组监控）代理使用与正则表达式和受监控进程的 **PID** 相匹配的进程数填充“进程组监视器”表条目。它轮询进程，更新包含聚合组统计信息的进程组监视器条目的只读属性的值，并检查组成员身份更改。
4. 代理检测到进程组更改、匹配日志文件消息、匹配事件时，会向陷阱团体中的所有系统发送陷阱。
5. 代理执行您在表条目中定义的任何操作。
6. 代理继续针对陷阱条件轮询进程组、日志文件、目录或 **Windows** 事件日志。

自主监控

SystemEDGE 代理可以监控受支持的 MIB 或 AIM 实施中列出的属性类型（INTEGER、GAUGE、COUNTER 和 TIMETICKS）。代理可以监控用户定义的阈值，并使用陷阱向管理系统报告违反阈值的情况。使用“自主监视器”表进行监控使您可以配置要监控的对象、要报告的异常条件和每个条件的重要级别。

在“自主监视器”表中创建条目时，需设置轮询时间间隔、比较运算符、阈值和重要级别，代理会自动监控您指定的 MIB 对象。可交替使用术语 MIB 对象和 MIB 属性。代理使用提供的重要级别值来确定对象的状态，并且它可以使用定义的多个监视器聚合对象的状态。

例如，您可以配置 SystemEDGE 以监控特定文件系统上的已用空间。要监控具有多个重要级别的阈值，请按如下所示在 `sysedge.cf` 中创建几个条目：

```
monitor devTableEntry 'C:' devCapacity 30 0x0 30 absolute >= 80 'C:' 'SysHealth'
'sysdrive' 'FS usage' 'ok'

monitor devTableEntry 'C:' devCapacity 30 0x0 120 absolute >= 80 'Used space on
C: drive' 'SysHealth' 'sysdrive' 'FS usage' 'warning'

monitor devTableEntry 'C:' devCapacity 31 0x0 120 absolute >= 85 'Used space on
C: drive' 'SysHealth' 'sysdrive' 'FS usage' 'minor'
monitor devTableEntry 'C:' devCapacity 32 0x0 120 absolute >= 90 'Used space on
C: drive' 'SysHealth' 'sysdrive' 'FS usage' 'critical'
```

该示例针对每个阈值定义多个具有不同重要级别的实体。受监控的对象由参数类 (SysHealth)、实例 (sysdrive) 和属性 (FS usage) 定义，该对象的状态为 ok（如果没有阈值交叉）或轻微、警告或严重（如果违反了相应的阈值）。对象状态发生更改时，代理会发送状态更改陷阱。

注意：有关详细信息和详细的监控示例，请参阅[“自主监控”](#) (p. 187) 一章。

详细信息：

[有状态监控的工作方式](#) (p. 45)

支持的 MIB

在“自主监视器”表中创建条目时，您可以指示代理针对指定的阈值监控任何受支持的 MIB 中的属性：

- CA Systems Management Empire MIB
- MIB-II (RFC 1213)
- 主机资源 MIB (RFC 1514)
- IP-MIB（部分）(RFC 4293)
- IF-MIB（部分）(RFC 2233)
- TCP-MIB（部分）(RFC 4022)
- UDP-MIB（部分）(RFC 4113)
- 任何 SystemEDGE 版本 5.0 AIM MIB

有关主机资源 MIB 中的对象的详细信息，请参阅附录“主机资源 MIB”。有关 MIB-II 中的对象的详细信息，请参阅 RFC 1213。

Systems Management Empire MIB 组

Systems Management Empire MIB 是特定于企业的包含自主监控对象表和组的属性集合。

下列组和表包含您可以在 Systems Management Empire MIB 中监控的对象：

系统组 (sysedgeSystem)

包含基本系统信息，如主机名、CPU 类型、内存量、时区、系统的完全限定域名、已加载插件的列表或 SystemEDGE 的正常运行时间。

挂接设备表 (devTable)

包含有关挂接在主机上的设备和文件系统的信息，如挂接点、块大小、可用块、使用中的 inode 或最大文件名长度。您还可以为文件系统使用率等值创建监视器，或通过设置该表中的列值来卸载已挂接的设备。

内核配置组 (kernelConfig)

包含内核信息，如 CPU 的数目、虚拟内存量、每个进程的内存、每个进程的打开文件、交换空间或时钟频率。使用该组来监控内核属性和配置。

引导配置组 (bootconf)

包含根文件系统、转储文件系统和交换空间的相关信息。监控该表来跟踪根文件系统名称、文件系统块或文件系统类型等值。

数据流组 (streams)

包含数据流 I/O 子系统的相关信息。使用该组来监控使用中的数据流的数目、数据流分配失败次数以及队列中的数据流数目等对象，从而跟踪子系统的运行状况。

用户表 (userTable)

包含系统上用户帐户的相关信息。

组表 (groupTable)

包含系统上用户组的相关信息。

进程表 (processTable)

包含正在运行的进程的相关信息。监控该表来跟踪当前正在运行的进程，并设置特定属性来控制进程。例如，您可以通过将 `processkill` 列的值设置为 9 来停止进程。

用户表 (whoTable)

包含当前登录到系统的用户的相关信息。监控该表中的属性来跟踪在任何给定时间使用系统的人员。

远程外壳组 (remoteshell)

包含用于在远程系统上运行外壳脚本和程序的属性。在该表中设置属性可指定命令及其参数以及输出文件的名称。

内核性能组 (kernelperf)

包含主机操作系统的运行状况和性能的相关信息。您可以监控当前进程和打开文件数、活动作业数以及调度程序的运行队列中的作业数等属性。

进程间通信表 (msgqueTable、shmemTable、semTable)

在不同的表中包含消息队列、共享内存和信号的相关信息。监控这些表可协调进程间的通信。

消息缓冲区分配表 (mbufAllocTable)

包含系统使用消息缓冲区的方式的相关信息。监控该表中的属性可跟踪诸如缓冲区请求被拒绝或延迟的数目等信息。

数据流缓冲区分配表 (strbufAllocTable)

包含有关缓冲区分配和数据流子系统所使用的缓冲区的使用率统计的信息。

I/O 缓冲区缓存组 (ioBufferCache)

包含基本磁盘 I/O 的 I/O 缓冲区分配和使用情况的相关信息。监控该表可跟踪诸如 I/O 缓冲区活动高峰期等信息。

目录名称查找缓存组 (dnlc)

包含目录和文件名缓存性能的相关信息。

AIX 逻辑分区组 (logicalPartition)

包含 IBM AIX 逻辑分区 (LPAR) 的相关信息。可以监控属性，例如每个分区的物理或逻辑 CPU 以及每个分区的 CPU 数目。

陷阱团体表 (trapCommunityTable)

包含 SNMP 信息，例如已配置的团体、用户和陷阱目标。

NT 系统组 (ntSystem)

包含特定于 Windows 系统的信息。该组包含系统、线程、注册表、服务、系统性能、缓存性能、内存性能、页面文件性能以及事件监视器组以监控 Windows 系统上这些方面的属性。

RPC 统计信息组 (rpc)

包含内核远程过程调用的相关信息。监控该表可跟踪属性，例如用于检测 RPC 活动高峰期的计数器和统计信息。

NFS 统计信息组 (nfs)

包含内核的 NFS 工具的相关信息。监视该表可跟踪属性，例如用于检测 NFS 活动高峰期的统计信息和计数器。

磁盘统计信息表 (diskStatsTable)

包含磁盘 I/O 的相关信息。

CPU 统计信息表 (cpuStatsTable)

包含每个 CPU 的性能统计信息。监控该表来跟踪在空闲模式下的时间以及在等待模式下的时间等属性。

Systems Management Empire MIB 还包含监控表和用于支持对象聚合的表。有关对象聚合的详细信息，请参阅[状态管理模型](#) (p. 59)。

自主监视器陷阱

在监控包含重要级别大于无(1)的“自主监视器”表中的条目时，SystemEDGE 代理会发送以下陷阱：

aggregateStateTrap

代理发送该陷阱来表示监视器的状态已发生更改。具有相同对象信息的监视器会进行聚合。对应的聚合状态等于被聚合监视器的最坏状态。

例如，如果某个重要级别为严重的监视器违反阈值，但是聚合监视器的当前状态为致命（这表示相同属性的另一个重要级别为致命的监视器已违反其阈值），则代理不发送陷阱，因为聚合状态不受影响。

注意：有关监视器聚合和其他聚合陷阱的详细信息，请参阅[状态管理模型](#) (p. 59)。

在监控包含重要级别为无的“自主监视器”表中的条目时（传统监视器），SystemEDGE 代理会发送以下陷阱：

monitorTrap

违反受监控属性的阈值条件时，代理发送该陷阱。

monitorEntryNotReadyTrap

“自主监视器”表条目的行状态设置为 `notReady(3)` 时，代理发送该陷阱。`notReady` 行状态可能表示对象不可访问。

monitorClearTrap

代理发送该陷阱来表示先前存在的阈值违反情况已不存在。

如果您在条目中定义重要级别，则不发送传统监视器的陷阱。有关传统监视器和在传统模式中使用代理的详细信息，请参阅[传统监视器](#) (p. 66)。

有关上述陷阱和其他陷阱的详细信息，请参阅附录“私有企业陷阱”。

进程和服务监控

SystemEDGE 代理可以监控任何进程或 Windows 服务，以检查它们的运行状态或进程属性的值。您可以通过在 Systems Management Empire MIB 的“进程监视器”表中创建条目来配置要监控的进程或服务 and 要跟踪的属性。

在“进程监视器”表中创建条目时，需设置轮询时间间隔、指示要监控的进程（或服务）的正则表达式、比较运算符、阈值和重要级别，代理会自动确定进程 ID 并监控您指定的进程属性。代理使用提供的阈值和重要级别值来确定进程属性的状态，并且它可以聚合定义多个监视器的状态以得到总体对象状态。

例如，您可以监控系统调用的任务关键进程，在它们超出提供的阈值时，代理会调整对象的重要级别并发送状态更改陷阱。您还可以监控进程是否正在运行，并且代理可以自动重新启动已停止的进程。

注意：有关详细信息，请参阅“进程和服务监控”一章。

详细信息：

[有状态监控的工作方式](#) (p. 45)

进程监视器陷阱

在监控包含重要级别大于 无(1) 的“进程监视器”表中的条目时，SystemEDGE 代理会发送以下陷阱：

aggregateStateTrap

代理发送该陷阱来表示进程监视器的状态已发生更改。仅当违反阈值条件会更改进程监视器的状态时，代理才会发送该陷阱。例如，如果某个重要级别为严重的监视器违反阈值，但是聚合监视器的当前状态为致命（这表示相同属性的另一个重要级别为致命的监视器已违反其阈值），则代理不发送陷阱，因为聚合状态不受影响。

注意：有关监视器聚合和其他聚合陷阱的详细信息，请参阅[状态管理模型](#) (p. 59)。

在监控包含重要级别为无的“进程监视器”表中的条目时（传统监视器），SystemEDGE 代理会发送以下陷阱：

processThresholdTrap

违反受监控进程属性的阈值条件时，代理发送该陷阱。

processStopTrap

受监控的进程已停止或处于无法运行的状态时，代理发送该陷阱。

processStartTrap

代理发送该陷阱来表示停止的受监控进程已经重新启动。

processClearTrap

代理发送该陷阱来表示先前存在且导致生成陷阱的阈值违反情况已不存在。

processNotReadyTrap

“进程监视器”表条目的行状态设置为 **notReady(3)** 时，代理发送该陷阱。**notReady** 行状态可能表示进程或属性不可访问。

如果您在条目中定义重要级别，则不发送传统监视器的陷阱。有关传统监视器和在传统模式中使用代理的详细信息，请参阅[传统监视器](#) (p. 66)。

有关上述陷阱和其他陷阱的详细信息，请参阅附录“私有企业陷阱”。

进程组监控

SystemEDGE 代理需要其他自主监视器来监控一组进程，从而加总单个进程属性值并跟踪该组中的进程。您可以通过在 **Systems Management Empire MIB** 的“进程组监视器”表中创建条目来配置为以组的形式监控进程（需要额外的自主监视器）。

在“进程组监视器”表中创建条目时，需设置轮询时间间隔、填充进程组的正则表达式和监视器重要级别，代理会以组的形式监控所有与正则表达式相匹配的进程。代理使用总和组统计信息（如进程合计大小、线程数和实际内存使用率）填充“进程组监视器”表的只读属性。如果进程组中的成员身份发生更改（例如，如果代理检测到与正则表达式匹配的新进程，或组中的进程变得不可访问），代理将发送组更改陷阱。在 UNIX 中，您也可以配置代理使其按照用户名和组名匹配进程。

例如，您可以监控一组重要进程，并且在其中一个进程中断时接收通知。如果某个进程中断，您可以针对“进程组监视器”表中的只读属性设置自主监视器。例如，您可以监控一组进程使用的 CPU 总时间，并设置对应于与相应应用程序关联的服务级别协议的阈值。

注意：有关详细信息，请参阅“进程组监控”一章。

详细信息：

[无状态监控的工作方式](#) (p. 46)

进程组监视器陷阱

监控“进程组监视器”表中的条目时，SystemEDGE 代理会发送以下陷阱：

procGroupChange

代理发送该陷阱来表示，由于存在新进程匹配进程组监视器条目规格或先前受监控的条目变得不可访问，进程组成员身份已发生更改。如果您为监视器定义了重要级别，则陷阱包括该重要级别值。

有关该陷阱和其他陷阱的详细信息，请参阅附录“私有企业陷阱”。

日志文件监控

SystemEDGE 代理可以监控对于正则表达式而言是行式的任何文本文件。要配置代理以监控日志文件，需在 **Systems Management Empire MIB** 的文本（日志）文件监视器表中创建条目。通过日志文件监控，您可以自动监控多个系统或应用程序日志文件中需要操作的条目。

在“日志文件监视器”表中创建条目时，需设置轮询时间间隔、要监控的日志文件（模式）、监视器重要级别以及要搜索的正则表达式。代理针对指定的正则表达式搜索日志文件，并在文件包含匹配表达式的消息时发送陷阱。

例如，您可以配置 SystemEDGE 代理，以监控某个应用程序日志文件中的失败消息或表示未经授权或错误使用的消息。

您还可以在“日志文件监视器”表中创建条目以监控文件目录的大小和内容（需要额外的自主监视器）。

注意：有关详细信息，请参阅“*日志文件监控*”一章。

详细信息：

[无状态监控的工作方式](#) (p. 46)

日志文件监视器陷阱

监控日志监视器表中的条目时，SystemEDGE 代理会发送以下陷阱：

logMonMatchTrap

代理发送该陷阱来表示，代理在某个日志文件中检测到与指定的正则表达式相匹配的项。如果您为监视器定义了重要级别，则陷阱包括该重要级别值。

logMonNotReadyTrap

日志监视器表条目的行状态设置为 **notReady(3)** 时，代理发送该陷阱。**notReady** 行状态可能表示文件不可访问。

有关上述陷阱和其他陷阱的详细信息，请参阅附录“私有企业陷阱”。

Windows 事件监控

SystemEDGE 代理可以监控 Windows 事件日志中的重要事件类型、事件标识符或与特定正则表达式相匹配的事件。通过在 Systems Management Empire MIB 的 NT 事件监视器表中创建条目来将代理配置为监控 Windows 事件日志。通过创建 Windows 事件监控条目，您可以管理 Windows 事件日志中的严重事件或需要操作的事件。由于 Windows 事件除文本消息之外还包括若干识别特征，因此通过该监控功能，您可以指定比基本日志文件监控更复杂的匹配类型。

在 NT 事件监视器表中创建条目时，NT 事件监视器会以固定的时间间隔进行轮询。要监控的事件类型和源、监视器重要级别和匹配事件消息文本（包括事件 ID）的正则表达式。代理监控 Windows 事件日志并在出现匹配表达式的事件时发送陷阱。

例如，您可以配置 SystemEDGE 代理以监控 Windows 安全事件日志中的身份验证失败。

注意：有关详细信息，请参阅“Windows 事件监控”一章。

详细信息：

[无状态监控的工作方式](#) (p. 46)

Windows 事件监视器陷阱

监控 NT 事件监视器表中的条目时，SystemEDGE 代理会发送以下陷阱：

ntEventMonMatchTrap

代理发送该陷阱来表示，代理在某个 Windows 事件日志中检测到与指定的正则表达式相匹配的项。如果您为监视器定义了重要级别，则陷阱包括该重要级别值。

ntEventMonNotReadyTrap

NT 事件监视器表条目的行状态设置为 `notReady(3)` 时，代理发送该陷阱。`notReady` 行状态可能表示 Windows 事件日志存储不可访问。

有关上述陷阱和其他陷阱的详细信息，请参阅附录“私有企业陷阱”。

历史记录收集

SystemEDGE 代理可以从任何受支持的 MIB 抽取属性值（度量标准）的样本，您可以使用收集到的样本进行系统基准分析和趋势分析。样本会填充历史记录表，并且您可以选择将样本写入多维数据集文件。

要配置历史记录收集，需在 Systems Management Empire MIB 的历史记录控制表中创建条目。通过配置历史记录收集，可以随时间推移分析度量标准，而无需经常地轮询管理器。写入多维数据集文件的数据可供 CA Virtual Assurance Systems Performance 进行分析。

在历史记录控制表中创建条目时，需设置轮询时间间隔、要收集的 MIB 属性和要保存的抽样数。SystemEDGE 根据配置收集数据，管理器可以在特定的时间间隔内定期检索历史记录以获取属性视图。

例如，您可以配置代理，使其每 10 分钟从 Systems Management Empire MIB 的 CPU 统计信息表收集一次 CPU MIB 属性，并保存最后 288 个抽样。然后，您可以配置管理系统使其每 48 小时上载一次数据，以便获取所有收集的数据进行分析。

有关配置历史记录收集的详细信息，请参阅“历史记录收集”一章。

历史记录收集的工作原理

SystemEDGE 代理按如下所示执行历史记录收集说明：

1. 在 Systems Management Empire MIB 的历史记录控制表中创建条目，以便在指定的时间间隔内监控 MIB 属性。

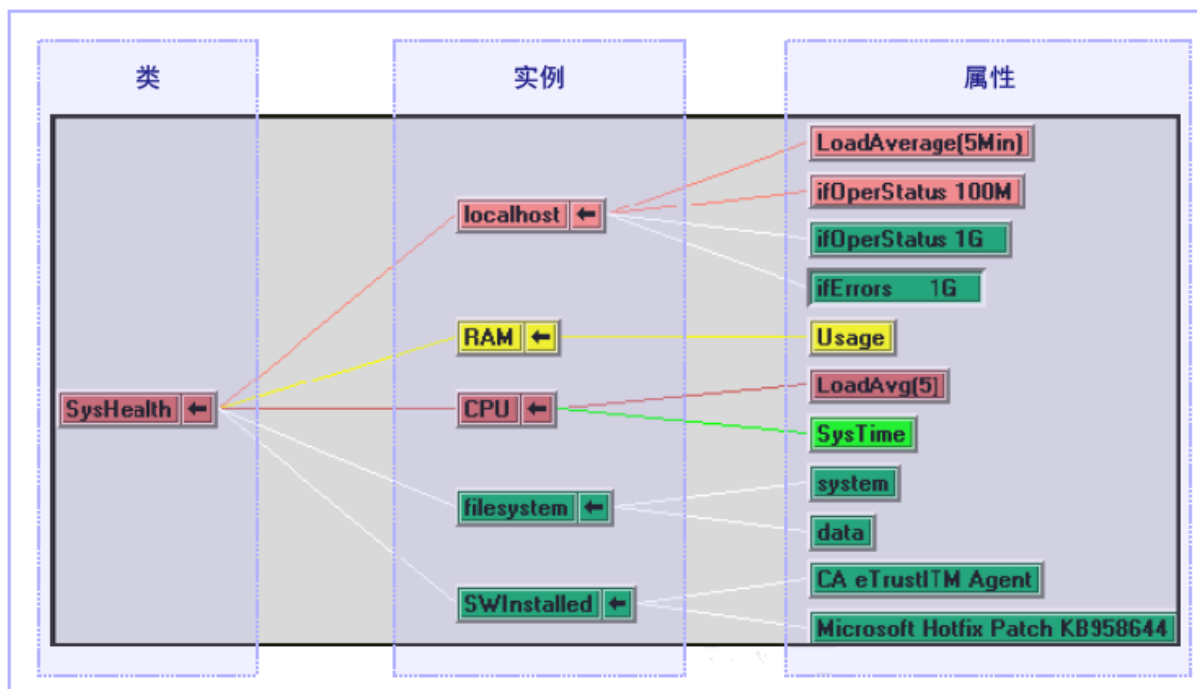
注意：有关创建历史记录收集条目的详细信息，请参阅“历史记录收集”一章。

除非使用 CA Virtual Assurance，否则需要重新启动代理才能识别对 sysedge.cf 文件所做的更改。如果通过 CA Virtual Assurance 创建条目并部署更改，代理将执行热启动并在运行时开始读取条目。

2. 代理根据指定的轮询时间间隔轮询属性，并将属性信息收集到历史记录表（滑动窗口）中。达到指定的抽样数时，会丢弃最早的抽样。

状态管理模型

SystemEDGE 代理支持状态管理模型，这样自主监视器和进程监视器可以与总体 CA Virtual Assurance 管理模型完全集成在一起。代理将不同重要级别的多个监视器聚合为单个受管对象，该受管对象的状态与具有最差重要级别的监视器相对应。下图显示了该新机制提供的灵活性和功能：



图中的“属性”窗格包含几个受监控的对象（如 SysHealth.localhost.LoadAverage）。SystemEDGE 代理可以分配给对象的状态与 CIM ManagedSystemElement 管理状态一致。每个属性对象的状态由相应的 SystemEDGE 监视器定义。例如，如果您为 LoadAverage 阈值 250 分配致命重要级别，并且当前的 LoadAverage 值是 250，则受监控对象的状态为致命。

SystemEDGE 的弹性对象模型使您能够对来自不同源的不同对象进行分组。跨各个组聚合对象的状态。例如，代理可以聚合服务器的 CPU 的运行状况，以及访问网页所用的响应时间（由 SRM AIM 收集）和磁盘性能（由远程监控 AIM 收集）。

图中的“类”窗格显示状态管理模型如何将受管对象分组成聚合受管对象。如图所示，此处 SysHealth 对象聚合由它管理的所有对象的状态。

您可以从 Systems Management Empire MIB 的自主监视器和进程监视器表配置状态管理功能，在这些表中，包含对象和重要级别信息的列控制每个监视器条目的对象状态的计算和传播。

对象状态

在自主监视器和进程监视器表中输入的重要级别值决定受监控对象的对象状态。会将特定的状态分配给违反特定阈值条件的受监控对象。在其他三个监控表中输入的重要级别值不决定受监控对象的状态，而是决定与监视器相关联的事件的重要性（无状态监控）。

您可以为监视器输入以下重要级别值：

- 无 (1)
- 正常 (2)
- 警告 (3)
- 轻微 (4)
- 重大 (5)
- 严重 (6)
- 致命 (7)

重要级别无 (1) 创建监控对象但不发送状态更改陷阱的传统监视器。如果未在 `objectState` 列中指定值，或如果您正在从先前版本升级监视器，则默认重要级别为无。有关详细信息，请参阅[传统监视器](#) (p. 66)。

您可以使用 `ok (2)` 作为临时的“不进行监控”功能，因为当前状态为 `ok(2)` 并且违反受监控属性重要级别时，不会发送状态更改陷阱。

由代理计算并记录为监视器条目中的只读属性的结果对象状态可能是以下各项之一：

- 未知 (1)
- 正常 (2)
- 警告 (3)
- 轻微 (4)
- 重大 (5)
- 严重 (6)
- 致命 (7)
- 运行 (11)
- 关闭 (12)

如果监视器的重要级别为无，状态会在运行 (11) 和关闭 (12) 之间进行切换。

代理在关闭时会保存任何监视器的当前状态。这些保存的状态用于在代理启动期间初始化监视器状态。

对象合计

自主监视器和进程监视器表中的对象类、实例和属性值描述受监控的（或受管）对象。具有相同对象实例、类和属性值的多个监视器被认为已连接且描述相同的受管对象。代理将所有已连接监视器的最差状态聚合为单个受管对象状态。代理在聚合表中为每个聚合创建一个条目（聚合）。

然而，如果两个连接的监视器 A 和 B 具有相同的重要级别并使用

“AND”标志（monFlags 中的 00800 或 pmonFlags 中的 20000），但如果 A 和 B 没有相同的状态，则它们将被排除在最差状态计算之外。如果您不在相应的列中输入值，代理会使用有意义的默认信息填充这些列。它使默认的自主监视器值基于受监控的 OID，并将这些值映射到类、实例和属性值。它使默认的进程监视器值基于进程正则表达式和受监控的属性。有关默认对象信息分配的详细信息，请参阅“自主监控”和“进程和服务监控”这两章。

所有连接的监视器必须具有相同的时间间隔，以帮助确保仅针对所有连接的监视器评估一次聚合状态，并防止不必要的状态更改。

聚合表包含以下信息：

- 对象类、实例和属性值
- 行状态
- 具有最差聚合状态的监视器条目的值，如阈值、运算符和当前值
- 当前的聚合状态
- 监视器统计信息，如发送的状态更改陷阱数、在任何一种可能的状态以及先前状态中花费的时间

注意：有关聚合表中各列的详细信息，请参阅“自主监控”一章。

代理更新聚合表中的当前状态。仅当对所有连接的监视器的阈值评估和随后的状态聚合导致与先前最差状态不同的最差状态时，代理才发送状态更改陷阱。例如，假定代理返回的 CPU 使用率为 82%，而您使用三个监视器监控 CPU 使用率：一个为 60%（分配的重要级别为警告），一个为 80%（重要级别为严重），另一个为 100%（重要级别为致命）。该值会导致 60% 和 80% 监视器超出阈值，但是代理仅对 80% 监视器发送一个状态更改陷阱，并将聚合状态更改为严重。

超级聚合

对于具有相同的对象类、实例和属性但具有不同重要级别（不同阈值）的所有自主监视器和进程监视器，聚合表中默认包含一个条目。该表基于重要级别进行聚合（标准聚合）。

另外，该表还允许类级别、属性级别或实例级别的超级聚合。相应的表条目使用星号 (*) 作为占位符。

属性级别

具有相同的对象类和实例的所有监视器都对应一个条目
(aggObjAttribute 设置为 *)

实例级别

具有相同的对象类和属性的所有监视器都对应一个条目
(aggObjInstance 设置为 *)

类级别

具有相同的对象实例和属性的所有监视器都对应一个条目
(aggObjClass 设置为 *)

可以在 `sysedge.cf` 文件中配置上述三个级别的组合，这将得到总体代理状态的聚合。

超级聚合的索引从 100001 开始。

重要信息！ 将行状态属性 `*,*,*` 条目设置为 `destroy (6)` 可以有效地删除所有重要级别大于无 (1) 的监视器。

根据以下格式，可以在 `sysedge.cf` 文件中配置聚合级别：

```
aggregate_level level
```

级别

指定一个允许指定下列聚合级别的位字段：

001：将所有具有相同类和实例的监视器聚合到一个属性设置为 * 的条目中

002：将所有具有相同类和属性的监视器聚合到一个实例设置为 * 的条目中

004：将所有具有相同实例和属性的监视器聚合到一个类设置为 * 的条目中

010：将所有具有相同类的监视器聚合到一个实例和属性设置为 * 的条目中

020：将所有具有相同实例的监视器聚合到一个类和属性设置为 * 的条目中

040：将所有具有相同属性的监视器聚合到一个类和实例设置为 * 的条目中

100：将所有监视器聚合到一个类、实例和属性设置为 * 的条目中

聚合表的条目（仅一些选定的属性）存储在 `sysedge.mon` 文件中。主要目的是在重新启动之间存储一个条目索引，因为通常管理器使用该索引来轮询对象并依赖该索引不会频繁更改。

状态管理和聚合的工作原理

SystemEDGE 代理在执行本节提及的自主监控和进程监控说明的同时，计算和聚合对象状态。

注意：该过程表示默认的状态管理和聚合行为。有关配置代理使其以不同方式管理状态和聚合对象状态的信息，请参阅[状态管理配置选项](#) (p. 65)。

1. 在自主监视器或进程监视器表中创建一个包含大于无的重要级别的自主监视器或进程监视器条目。您应当定义对象实例、类和属性信息，否则代理将使用有意义的默认信息填充这些列。

除非使用 CA Virtual Assurance，否则需要重新启动代理才能识别对 `sysedge.cf` 文件所做的更改。如果通过 CA Virtual Assurance 创建条目并部署更改，代理将执行热启动并在运行时开始读取条目。

2. 代理根据指定的轮询时间间隔监控条目中定义的属性，根据指定的行状态、阈值违反条件和重要级别获取监视器的当前状态，并使用当前状态填充监视器条目中的只读属性。

3. 代理以下列方式将状态信息添加到聚合表中：
 - 如果聚合表中已存在具有监视器的对象类、实例和属性值的条目，代理会（在内部）将监视器连接到表中的该条目。
 - 如果聚合表中不存在具有监视器的对象类、实例和属性值的条目，代理会向聚合表中添加新条目。
4. 代理向聚合表中添加新条目时，会向陷阱社区中的所有系统发送一个聚合添加陷阱。
5. 监视器超出阈值时，代理会更改自主监视器或进程监视器表中的监视器的当前状态，聚合所有连接的监视器的最差状态，并更新聚合表中的相应条目。
6. 仅当聚合状态发生更改时，代理才发送状态更改陷阱。

注意：聚合遵守指定的延迟设置。如果为监视器条目设置延迟值，则在满足延迟时间之前，单个监视器状态或聚合状态均不会发生更改。有关使用表标志来设置延迟值的详细信息，请参阅 *自主监控* 和 *过程和 服务监控* 这两章。

7. 代理执行与最差监视器关联的操作。
8. 代理继续轮询属性，并在以下情况下按以下方式操作：
 - 如果阈值违反条件持续存在，则代理继续执行与该监视器关联的操作（只要它仍然表示最差状态监视器），但是不发送其他状态更改陷阱。
 - 如果阈值违反条件消失，则代理更新监视器的当前状态，使用最差的当前状态更新聚合表中的当前状态，并发送相应的状态更改陷阱。
 - 如果具有更高重要级别的已连接监视器超过其阈值，代理会更改聚合状态并发送状态更改陷阱。

注意：代理继续在自主监视器和进程监视器表中管理单个自主监视器和进程监视器条目。代理继续轮询每个单独监视器的属性，并在表中维护每个监视器的状态。由于这些原因，您应当为所有连接的监视器设置相同的轮询时间间隔，以便同时评估所有对象监视器并阻止不必要的状态更改（例如，从“正常”到“严重”，然后是“严重”到“致命”，具有交错的轮询时间间隔）。

状态管理陷阱

监控包含大于无的重要级别的自主监视器和进程监视器条目时，SystemEDGE 代理发送以下陷阱：

注意：对于重要级别为无的自主监视器和进程监视器条目（即，传统监视器），代理发送来自前一个代理版本的阈值违反陷阱。有关这些陷阱的详细信息，请参阅“自主监视器陷阱”和“进程监视器陷阱”。

aggregateStateTrap

聚合状态发生更改时，代理会发送该陷阱。在已连接监视器的阈值违反导致聚合状态发生更改时。例如，如果超过了重要级别为严重的监视器，而聚合的当前状态为致命，则代理不发送陷阱，因为聚合状态不受影响且将保持为致命。

注意：即使聚合状态不受影响，您也可以配置代理以发送状态更改陷阱。有关详细信息，请参阅“代理配置”一章。

aggregateAddTrap

向聚合表中添加行时（即，在已创建重要级别大于无的监视器且尚不存在具有该监视器的对象信息的相应聚合时），代理会发送该陷阱。否则，新监视器将作为连接的监视器（在内部）添加到聚合表的现有行中，并且不触发 aggregateAdd 陷阱。

aggregateDeleteTrap

从聚合表中删除行时，代理会发送该陷阱。

状态管理配置选项

您可以配置监视器聚合的以下方面：

- 您可以配置代理以将聚合表中的相同条目聚合为超级聚合对象。这可以在类级别、实例级别和属性级别独立完成。

例如，如果将代理配置为在属性级别聚合（超级），则代理在聚合表中为所有具有相同对象类和实例但是具有不同属性的聚合创建其他条目。超级聚合的属性值为星号 (*)。使用该示例，您可以聚合单个进程的所有受监控属性，以便为进程计算一个聚合状态。您可以在 sysedge.cf 文件中使用参数来配置这些聚合的任意组合。

注意：有关配置聚合级别的示例的详细信息，请参阅 *代理配置* 一章。

- 默认情况下，代理仅发送聚合状态更改陷阱。您可以在 `sysedge.cf` 文件中使用参数来配置代理，以便为所有已连接监视器的阈值违反发送传统陷阱。
- 默认情况下，代理仅为具有最差状态的已连接监视器执行操作。您可以在 `sysedge.cf` 文件中使用参数来配置代理，以便为所有已连接的监视器执行操作。

注意：有关配置聚合陷阱和操作的详细信息，请参阅 *代理配置* 一章。

- `aggRowStatus` 属性是聚合表中唯一可读写的属性。该表聚合了已连接监视器的行状态，其方式类似于聚合对象状态。通过处理行状态属性，您可以更改所有已连接监视器的聚合行状态。您可以将行状态设置为 `active` (1)、`notInService` (2) 或 `destroy` (6)，并且所有已连接的监视器都继承同一种状态。将行状态设置为 `destroy` 会从聚合表中删除一个条目。

注意：有关行状态聚合以及从聚合表中删除条目的详细信息，请参阅 *自主监控* 和 *进程和服务监控* 这两章。

传统模式操作

您可以以传统模式操作 `SystemEDGE`，也就是说，在没有 `CA Virtual Assurance` 管理器控制其配置的情况下。以传统模式运行的代理并不限于传统监视器。

传统监视器

现有管理器可以监控 `SystemEDGE` 属性和进程，而无需合并对象状态模型。自主监视器和进程监视器表中所有的状态管理列都是可选的；如果您希望代理监控属性和进程而不计算状态信息，请将这些字段留空。

在不维护和计算状态的情况下进行操作的自主监视器和进程监视器都是传统监视器。这些监视器的工作方式与 `SystemEDGE` 版本 5.0 之前的所有监视器相同。如果不输入自主监视器或进程监视器的重要级别，则默认情况下监视器继承的重要级别为无 (1)。无论是手动进行定义还是默认情况下的定义，重要级别为无 (1) 的所有监视器都是传统监视器。它们的当前状态只能为运行或关闭，并且发送传统阈值陷阱，而不是状态更改陷阱。

注意：有关传统阈值陷阱的详细信息，请参阅“自主监视器陷阱”和“进程监视器陷阱”。

监视器示例

以下示例显示监视器和生成的聚合状态：

聚合状态 = 严重

监视器	重要级别	AND 标志	违反	状态
M1	警告	否	是	警告
M2	警告	否	否	正常
M3	严重	否	是	严重
M4	严重	否	否	正常

聚合状态 = 正常

监视器	重要级别	AND 标志	违反	状态
M1	警告	是	是	警告
M2	警告	是	否	正常
M3	严重	是	是	严重
M4	严重	是	否	正常

聚合状态 = 警告

监视器	重要级别	AND 标志	违反	状态
M1	警告	是	是	警告
M2	警告	是	是	警告
M3	严重	是	是	严重
M4	严重	是	否	正常

与 CA Virtual Assurance 集成

CA Virtual Assurance 是一种管理解决方案，旨在通过 SystemEDGE 代理来管理物理系统和虚拟系统。该代理与 CA Virtual Assurance 紧密集成，以便提供其他管理器或单机代理所不提供的功能。通过与 CA Virtual Assurance 的集成，您可以查看 SystemEDGE 数据，远程配置 SystemEDGE 并将其部署到 CA Virtual Assurance 用户界面中所有的受管系统。

安装代理（通过 CA Virtual Assurance 或单机）时，可以指定端口和陷阱，它们是通过 CA Virtual Assurance 管理器使代理可见且便于管理所必需的。在安装期间还可指定数据目录和默认端口。CA Virtual Assurance 使用数据目录来传送配置更改。数据目录与默认端口号附加在一起。例如，如果使用默认端口 161，则数据目录将为 C:\Program Files\CA\SystemEDGE\data\port161（在 Windows 上）。

平台管理模块

CA Virtual Assurance 管理器组件通过平台管理模块 (PMM) 实现与 SystemEDGE 代理的集成。SystemEDGE PMM 是使用 SNMP 在所有受管系统上轮询 SystemEDGE 代理的 Web 服务。它将从代理收到的所有信息（包括 MIB 对象值和陷阱）转换为适合 CA Virtual Assurance 体系结构的属性，并且在 CA Virtual Assurance 数据库中创建相应的受管对象表示。SystemEDGE PMM 通过这些受管对象在 CA Virtual Assurance 用户界面中提供 SystemEDGE 信息的可视化。您可以通过用户界面查看已安装的所有代理、系统详细信息、现有监视器和状态信息，以及进行配置更改。

与 CA Virtual Assurance 中的可视化和检测兼容的所有 AIM 均包含单独的 PMM。对于 CA Virtual Assurance，服务响应监视器 AIM、远程监控 AIM 及所有虚拟监控 AIM 均包含 PMM。

有关平台管理模块及其如何将代理信息集成到 CA Virtual Assurance 框架的详细信息，请参阅 CA Virtual Assurance 文档。

远程配置和部署

CA Virtual Assurance 从一个集中位置通过基于文件的配置和部署提供 SystemEDGE 的端对端管理。基于文件的配置会执行以下操作：

- 允许执行 SNMP 管理器无法通过 SNMP 集执行的更改，例如配置系统管理 Empire MIB 的只读区域（如团体字符串、运行参数、陷阱团体、AIM 加载等）
- 启用可验证交付并创建比 SNMP 更少的安全漏洞
- 启用完整的远程代理配置和交付，并将需要本地手工文件处理的工作减至最少，甚至不需要执行此操作

CA Virtual Assurance 管理器为每个支持平台都提供了 SystemEDGE 安装软件包。通过部署解决方案，您可以指定凭据和安装参数，在指定所有需要的信息后，可以保存部署软件包并将其部署到企业的受管系统中。

从 CA Virtual Assurance 用户界面中，您可以设置 `sysedge.cf` 文件中的所有代理配置参数，并使用监视器表的图形表示创建自主监控条目。可以将代理配置另存为配置文件，并将该配置文件应用于部署软件包以部署到受管系统中。还可以对安装的代理进行点配置更改，并立即应用这些更改。

部署配置更改时，CA Virtual Assurance 将在文件中使用模板指令编译 `sysedge.cf` 文件，以便将在 CA Virtual Assurance 用户界面中输入的信息用于配置和自主监控。它使用 CAM（CA 消息传送）作为传输机制将编译的 `sysedge.cf` 文件发送到指定的系统。CAM 将文件发送到在安装期间指定的数据目录。SystemEDGE 代理侦听包含配置文件的 CAM 消息，在检测到文件时，代理将检查文件的语法是否正确，然后自动加载并使用该文件。通过 CA Virtual Assurance 配置代理将触发代理热启动，这样无需重新启动即可自动应用对代理进行的更改。

事件引擎

CA Virtual Assurance 事件引擎收集以下事件，以跟踪 SystemEDGE 和 CA Virtual Assurance 方面的所有活动：

- SystemEDGE 状态更改陷阱
- 跟踪并显示 CA Virtual Assurance 已初始化的部署和配置更改的状态的事件
- 通知 CA Virtual Assurance 管理员对代理配置所做的 SNMP 更改的事件

事件引擎提供了可靠的位置，用于跟踪 SystemEDGE 异常和操作以及审核代理配置更改。

CA Virtual Assurance 安全选项

CA Virtual Assurance 提供了多个 SystemEDGE 安全锁定选项，可用于增强代理安全并使 CA Virtual Assurance 成为其他所有更改必须通过其获得批准的单一配置点。您可以通过 CA Virtual Assurance 配置以下安全选项：

取消 SNMP 写入

取消通过 SNMP（使 CA Virtual Assurance 或本地 sysedge.cf 文件处理成为配置代理的唯一方法）修改代理的功能。CA Virtual Assurance 要求具有 SNMP 写入访问权限。

受管模式

导致任何 CA Virtual Assurance 已初始化的更改覆盖通过其他方法对代理所做的更改。在 SystemEDGE 安装期间输入 CA Virtual Assurance 管理器节点时，代理会处于受管模式。受管模式使 CA Virtual Assurance 成为取代其他所有更改的配置模式。例如，如果用户直接修改 sysedge.cf 文件，并随后通过 CA Virtual Assurance 将某个文件部署到该系统，则 CA Virtual Assurance 传送的文件中的设置会覆盖本地文件中的设置。

SNMP 更改通知

当代理应用通过 SNMP Set 操作启动的基于 SNMP 的更改时通知 CA Virtual Assurance 管理器。该选项仅适用于处于受管模式的代理。默认情况下，当代理处于受管模式时会启用该选项。在 CA Virtual Assurance 用户界面中，您可以决定可接受的更改并覆盖任何不需要的更改。启用该选项后，SNMP Set 更改也将记录到位于代理安装的数据目录的 sysedge_audit.log 文件中。

有关启用和配置这些安全选项的详细信息，请参阅 CA Virtual Assurance 文档。

AIM 体系结构

Application Insight Module (AIM) 将 SystemEDGE 代理功能扩展到特定于应用程序的监控,或提供基础代理中不存在的专用监控功能。各种 AIM 具有不同的体系结构,但以下约定适用于所有 AIM:

- 通常, AIM 在安装后会自动加载。AIM 安装程序应在目录 `SystemEDGE/plugins/plugin-name` 中创建文件 `enable_req`。或者,如果代理处于未受管模式,您可以在 `sysedge.cf` 文件中输入 `sysedge_plugin <plugin name>` 参数。

注意: 必须在 CA Virtual Assurance 安装期间指定写入访问权限。

- SystemEDGE 可以通过热启动加载 AIM,但是代理需要完全重新启动才能删除先前加载的 AIM。
- 每个 SystemEDGE 版本 5.0 AIM 均拥有它自己的且受核心代理支持的 MIB。
- 每个 AIM 都可以有它自己的配置文件,您可在其中输入 AIM 配置说明。

有关使用 AIM 的详细信息,请参阅特定于 AIM 的文档。

按需模块

SystemEDGE 为以下按需模块提供基础代理:

- 监控 Windows
- Perfcube

这些模块不要求 `sysedge.cf` 文件中包含 `sysedge_plugin` 语句。需要该功能时, SystemEDGE 会自动加载按需模块。

注意: 先前包含在顶级进程 AIM 中的功能现已与基础代理集成。有关详细信息,请参阅“进程和服务监控”一章。

有关对模块及其提供的功能进行配置的详细信息,请参阅 *SystemEDGE AIM* 一章。

AIM 与 CA Virtual Assurance 的集成

CA Virtual Assurance 支持 CA Virtual Assurance 附带的所有 AIM 的远程部署，以及 SystemEDGE 和 SRM AIM 的策略配置。AIM 通过创建有状态的监视器来支持 SystemEDGE 状态管理模型，以便可以通过 CA Virtual Assurance 用户界面查看与响应时间测试关联的状态信息。

支持以下 AIM 与 SystemEDGE 版本 5.7.1 结合使用，但不支持通过 CA Virtual Assurance 进行的可视化和配置：

- IIS AIM
- Apache AIM
- Voice AIM
- CA Insight AIM
- Exchange Server 和 Active Directory AIM（不适用于 CA Server Automation）

您可以在本地加载和配置这些 AIM，但 CA Virtual Assurance 没有用于解释 AIM 数据的平台管理模块，并且不提供用于配置的任何接口。

注意：有关详细的 AIM 支持列表，请参阅《SystemEDGE 版本说明》。

使用 SystemEDGE 代理的准则

通过为系统、网络 and 应用程序设置有效的管理策略，您可以充分利用 SystemEDGE 代理。尤其应遵循以下准则：

- 管理 CA Virtual Assurance 中处于受管模式的代理，并限制基于 SNMP 的更改或监控这些更改，以便您拥有一个一致的点进行代理配置。
- 通过访问控制列表并绑定到专用接口来限制对代理的 SNMP 访问。
 - 使用 SNMP Set 的读写社区和只读社区进行查询和轮询。
 - 使用端口 161（SNMP 行业标准端口）或 1691（Internet 编号分配机构 [IANA] 保留用于 SystemEDGE 代理的端口）。

注意：有关指定团体字符串和配置端口的详细信息，请参阅“代理配置”一章。

- 基于以下内容在管理软件中创建组：

- 操作系统
- 数据库系统
- 客户端
- 文件服务器
- 电子邮件服务器
- Web 服务器
- 物理位置

注意：有关创建组的详细信息，请参阅《CA Virtual Assurance 管理指南》或《CA eHealth 管理指南》。

- 按系统组定义配置策略，而不是按单个系统。使用 CA Virtual Assurance 将配置集部署到系统组。
- 在系统位置和 sysedge.cf 文件的联系人字段中包括有意义的信息。例如，包括诸如机架 0、插槽 1、亚特兰大以及电子邮件地址 it@yourdomain.com 之类的信息。
- 在监控表之间使用众所周知的行索引范围。定义这些索引和保留行时，请牢记以下事项：
 - 对每种类型的监控使用大范围的索引号以允许增长。
 - 将标准索引条目用于特定类型的监控条目。例如，始终使用第 10,000 行监控可用的 CPU 总量。
 - 使用 CA Virtual Assurance 将自主监控条目应用于系统组。

注意：有关使用标准索引行的详细信息，请参阅“自主监控”一章中的“为监视器表分配条目行”。

- 在创建可跨多个系统使用的条目时，请注意以下事项：
 - 监控 CPU 使用率总计状态（此操作使配置可以在单处理器系统和多处理器系统之间移植）。
 - 监控阈值并配置 SystemEDGE 以发送有限数量的陷阱（例如，两个或三个），以防止 CA Virtual Assurance 因收到大量陷阱而停止。

注意：有关有效监控阈值和清除陷阱的详细信息，请参阅“自主监控”一章中的“监视器表标志”。

- 如果您要管理大型网络（数百或数千个系统）并且针对粒度数据轮询每个系统（每隔不到 15 分钟），请执行以下操作：
 - 使用历史记录收集获取代理级别的高粒度数据，并通过管理系统轮询历史记录表。您可以将 SystemEDGE 用于短期历史记录收集，将 CA eHealth（或其他管理系统）用于长期历史视图。
 - 有关详细信息，请参阅“历史记录收集”一章。
 - 将监控推送到代理，并配置代理以基于这些自主监控条目发送陷阱。
 - 通过使用有状态的监视器限制单个监控的实体中潜在的陷阱数量。

有关更多的使用提示，请参阅本指南中的以下各节：

- “代理配置”一章中的“针对配置安全性的建议”
- “进程和服务监控”一章中的“针对进程和服务监控的建议”
- “日志文件监控”一章中的“针对日志文件监控的建议”
- “自定义 MIB 对象”一章中的“针对使用扩展的建议”

第 4 章： 安装和部署

本章说明如何在所有支持的平台上执行 SystemEDGE 代理的单机安装，并介绍使用 CA Virtual Assurance 进行代理部署的概念。

此部分包含以下主题：

[安装说明](#) (p. 75)

[安装期间的代理配置](#) (p. 76)

[使用 CA Virtual Assurance 安装](#) (p. 76)

[在 Windows 系统上安装](#) (p. 76)

[在 UNIX 和 Linux 系统上进行安装](#) (p. 88)

[配置和使用响应文件](#) (p. 100)

[以传统模式安装代理](#) (p. 100)

[代理部署](#) (p. 102)

[卸载 SystemEDGE](#) (p. 105)

[重新安装](#) (p. 108)

[升级受管节点和 AIM 服务器](#) (p. 109)

安装说明

在安装 SystemEDGE 代理之前需考虑以下事项：

- 验证系统是否满足《SystemEDGE 版本说明》中列出的系统要求。
- CA_SystemEDGE_Core 软件包会安装 SystemEDGE 代理。
- CA_SystemEDGE_AdvancedEncryption 软件包会提供代理加密库。有关安装高级加密软件包的详细信息，请参阅“SystemEDGE 高级加密”附录。
- 可以更改安装期间在 CASYSEDGE_DATA\port<default-port>\sysedge.cf 文件中定义的大多数值。该文件还提供了其他配置选项。有关使用该文件配置代理的详细信息，请参阅“代理配置”一章。
- 可以在安装期间配置 SNMPv1 和 SNMPv2c 设置，但必须在安装后配置 SNMPv3 设置。有关详细信息，请参阅“代理配置”一章。
- 在任意安装操作中使用特殊字符或保留字符时，请考虑针对各操作环境的自述文件和限制中说明的“特殊字符限制”。操作环境限制包括（但不限于）UNIX shell 或 Windows 命令提示符中的特殊字符或保留字符。

安装期间的代理配置

使用 SystemEDGE 安装，可以在安装代理时执行一些配置任务。例如，可以在安装过程中配置系统说明和位置、只读团体和读写团体以及 SNMPv1 陷阱目标。以后可以通过编辑 `sysedge.cf` 文件修改这些设置中的任何设置。

注意：SNMPv3 用户和陷阱目标不在安装期间进行配置。必须通过在数据目录中手动编辑 `sysedgeV3.cf` 配置文件来输入这些设置。有关为 SystemEDGE 代理配置 SNMPv3 的详细信息，请参阅 [“SNMPv3 配置”](#) (p. 150)。

详细信息

[File-Based 配置](#) (p. 117)

使用 CA Virtual Assurance 安装

本章介绍如何执行 SystemEDGE 代理的单机安装。使用 CA Virtual Assurance 管理代理时，最好的做法是使用 CA Virtual Assurance 用户界面部署代理。有关部署的详细信息，请参阅 [“使用 CA Virtual Assurance 部署代理”](#) (p. 103) 和 《CA Virtual Assurance 实施指南》。

默认情况下，安装 CA Virtual Assurance 管理器时会在管理器系统上安装 SystemEDGE。不支持在本地系统上进行部署。

在 Windows 系统上安装

本节介绍如何在 Windows 系统上手动安装 SystemEDGE。可以使用交互式向导进行安装，也可以使用命令行以无提示方式进行安装。单独的软件包可用于所有支持的硬件体系结构。安装程序会检测硬件体系结构并运行相应的安装软件包。

下列情形要求您手工安装代理，而不是使用建议的 CA Virtual Assurance 部署方法：

- 在不支持远程部署的系统上安装代理。
- 想要以传统模式安装代理。

在本指南中，术语 *Windows* 涵盖支持的 Windows 版本。有关所支持版本的列表，请参阅 《SystemEDGE 版本说明》。

详细信息

[以传统模式安装代理](#) (p. 100)

在 Windows 上安装代理

可以使用交互式向导在 Windows 上手工安装 SystemEDGE 代理。

遵循这些步骤:

1. 以管理员身份登录到 Windows 系统，并执行以下操作之一：
 - 从 CA Virtual Assurance 安装映像的 DVD1 双击 setup.hta，然后在 CA Virtual Assurance 对话框中单击“安装 SystemEDGE 代理”。
 - 打开 DVD1\Installers\Windows\Agent\SysMan\CA_SystemEDGE_Core 文件夹并双击 ca-setup.exe。

注意：对于运行 Windows Vista 或更高版本的系统，您能够以非管理员身份进行安装。操作系统会提示您使用管理员凭据授权安装。

安装向导将打开。安装程序会检测系统的硬件体系结构并运行相应的安装软件包版本。

2. 单击“下一步”。
- 此时将打开“许可协议”页面。
3. 阅读“许可协议”并滚动到协议底部。选择“我接受此许可协议的条款”选项，然后单击“下一步”。
- 此时将打开“安装类型”页面。
4. 选择“典型安装”或“自定义安装”，然后单击“下一步”。

注意：以下过程介绍自定义安装。如果选择“典型安装”，单击“下一步”时将打开“检查设置”页面。

此时将显示“目标位置”对话框。

5. 接受默认值或进行浏览以选择安装和数据目录的位置，单击“下一步”，然后继续执行第 9 步。如果要指定其他位置，请单击“高级”并继续执行第 8 步。

目标位置

指定要安装代理的位置。默认情况下，安装目录是 `Install_Path\SystemEDGE`，运行时程序数据存储在 `config` 子目录中。要指定更多参数，请单击“高级”。

注意：在已安装先前版本代理的系统上安装该代理时，安装程序会选择现有代理的安装目录。

如果跳过“高级”对话框，将出现“配置管理器设置”对话框。

6. 填写“高级目标位置”对话框的以下字段，然后单击“下一步”。

SystemEDGE 二进制路径

为二进制程序和文档指定目录。

SystemEDGE 数据路径

为运行时程序数据指定目录。

CA 共享组件路径

为 CA 共享组件指定目录。一旦由任意 CA 软件设置该目录，就无法对其进行更改并且会禁用用户界面中的相应字段。

此时将显示“配置管理器设置”对话框。

7. 填写以下字段，然后单击“下一步”。

配置管理器主机名

指定要用从管理该代理的配置管理器的主机名。为该参数输入值，以从运行 CA Virtual Assurance 的系统上配置该代理。输入星号 (*) 可接受第一个发现代理系统的管理器。

默认配置策略名称

指定供代理使用的配置策略文件（由 CA Virtual Assurance 管理器维护）的名称。为该参数输入值，以根据管理器中的现有配置文件设置 SystemEDGE 配置。

如果安装程序检测到系统上正在运行本地 Microsoft SNMP 代理，将显示以下“本地 SNMP 选项”对话框。

8. 如果您的系统上已安装简单网络管理协议 (SNMP)，请选择以下选项之一，然后单击“下一步”。

默认来自现有 SNMP 代理

指定是否使用本地 SNMP 代理的默认设置。取消选中该复选框可使用与本地 SNMP 代理不同的团体字符串和陷阱目标。

禁用本地 SNMP 代理

指定是否停止并禁用本地 SNMP 代理。如果使本地 SNMP 代理处于启用状态，则在其他端口上运行 SystemEDGE。

9. 填写以下字段，然后单击“下一步”。

SNMP 端口号

指定要在其上运行 SystemEDGE 代理的端口。

默认值：161

重要信息！该端口号专用于 SystemEDGE 代理。如果其他应用程序使用该端口号，则安装将失败。如果本地 SNMP 代理已使用默认端口，请指定其他端口（例如，1691 或 6665）。

此时将显示“SNMP 系统信息”对话框。

10. 填写以下字段，然后单击“下一步”。

系统说明

指定用于填充 sysDescr MIB-II 对象的系统的信息（如系统名称）。

系统位置

指定用于填充 sysLocation MIB-II 对象的系统位置。

系统联系人

指定用于填充 sysContact MIB-II 对象的系统联系人信息。

此时将显示“SNMP 团体设置”对话框。

11. 填写以下字段，单击“下一步”，然后继续执行第 13 步。如果要指定多个团体字符串，请单击“高级”并继续执行第 12 步。

只读团体

指定 SNMP 只读团体字符串。

默认值：public

读写团体

指定 SNMP 读写团体字符串。

如果跳过“高级”对话框，将显示“SNMP 陷阱设置”对话框。

12. 填写“SNMP 团体设置 - 高级”对话框中的以下字段，然后单击“下一步”。

只读团体

指定 SNMP 只读团体字符串。可以指定多个团体，用分号进行分隔（例如，`public1; public2`）。还可以包括各团体的 IP 地址列表以限制访问（例如，`public 1.2.3.4`）。

默认值： public

读写团体

指定 SNMP 读写团体字符串。可以指定多个团体，用分号进行分隔（例如，`rwcomm1; rwcomm2`）。还可以包括各团体以空格分隔的 IP 地址列表以限制访问（例如，`rwcomm1 1.2.3.4`）。读写团体字符串对于正确操作某些 AIM（例如，RM）和一些远程使用（例如，创建监视器）是必需的。

此时将显示“SNMP 陷阱设置”对话框。

13. 填写以下字段，单击“下一步”，然后继续执行第 15 步。如果要指定多个陷阱目标，请单击“高级”并继续执行第 14 步。

陷阱团体字符串

指定在发送陷阱消息中编码的 SNMP 团体。

默认值： public

目标主机

指定陷阱消息的目标。

默认： 在“配置管理器设置”对话框中设置的 *配置管理器主机名*。

端口号

指定陷阱消息发送到的端口。

默认： 162。

如果跳过“高级”对话框，将显示“杂项设置”对话框。

14. 填写“SNMP 陷阱设置 - 高级”对话框中的以下字段，然后单击“下一步”。

陷阱配置

指定一个或多个陷阱目标。可以指定多个条目，用分号进行分隔（例如，`public server1; public server2 1162`）。

此时将显示“杂项设置”对话框。

15. 填写下列字段，然后单击“下一步”：

安装之后启动

指定安装结束时是否启动代理。

安装文档

指定是否安装文档。

此时将显示“检查设置”页面。

16. 检查安装设置并单击“安装”。

安装完成后将显示“安装完成”页面。

17. 单击“完成”。

安装即会完成。

在 Windows 上从命令行安装代理

可以使用命令行版本的安装程序安装 SystemEDGE Windows 软件包。从命令行安装时，使用参数设置各种安装属性。可从命令行执行以下操作：

- 在预先填写安装参数或不预先填写安装参数的情况下启动安装向导
- 指定安装参数并在不使用交互式向导的情况下运行安装程序

该过程适用第二种方案：从命令行运行无人照管的安装。

遵循这些步骤：

1. 以管理员身份登录到 Windows 系统。

注意：对于运行 Windows Vista 和更高版本的系统，可以非管理员身份进行安装，操作系统会提示您使用管理员凭据授权安装。

2. 打开命令提示符，导航到

DVD1\Installers\Windows\Agent\SysMan\CA_SystemEDGE_Core 文件夹，然后输入以下所需参数（在完成第 3 步之前，请勿运行该命令）：

```
ca-setup CA_SETUP_MODE=UNATTENDED EULA_ACCEPTED="YES" [parameter]
```

注意：有关安装参数的帮助，请在命令提示符下输入 `ca-setup -?` 在命令提示符中。

CA_SETUP_MODE

指定安装模式。将该参数设置为 `UNATTENDED` 以在无提示模式下运行安装。如果省略该参数，则运行该命令后，安装向导打开时会预先填写指定的参数值。

EULA_ACCEPTED

读取许可协议并指定是否接受该许可协议。如果在将安装模式设置为 `UNATTENDED` 时省略该参数或将其设置为除 **“YES”** 以外的任何值，安装将失败。交互式安装中不需要该参数。

3. 根据需要添加可选参数并运行命令。

注意：如果省略以下可选参数，则无需提供这些参数的值：

CA_SETUP_LOG_FILE
CA_SETUP_VERBOSE
CASE_INSTALLDIR
CASE_PUBDATADIR
CASE_SNMP_PORT
CASE_SNMP_SYS_DESC
CASE_SNMP_SYS_LOC
CASE_SNMP_SYS_CONTACT
CASE_SNMP_READ_COMMUNITY
CASE_SNMP_READ_ALLOWED_MANAGERS
CASE_SNMP_WRITE_COMMUNITY
CASE_SNMP_WRITE_ALLOWED_MANAGERS
CASE_SNMP_TRAP_COMMUNITY
CASE_SNMP_TRAP_DESTINATION
CASE_SNMP_TRAP_PORT
CASE_DISABLE_NATIVE_SNMP
CASE_DEFAULT_FROM_NATIVE_SNMP
CASE_MANAGER_HOSTNAME
CASE_MANAGER_POLICY_NAME
CASE_START_AFTER_INSTALL
CASE_INSTALL_DOCS
CASE_LEGACY_MODE

CA_SETUP_LOG_FILE

指定记录安装消息的位置和文件名。

CA_SETUP_VERBOSE

设置为“yes”时将打开详细安装模式。详细模式会将更多信息记录到安装日志文件中。

CASE_INSTALLDIR

指定 SystemEDGE 安装目录。该目录中包含与 SystemEDGE 相关的所有内容（如 AIM 和高级加密），并且绝不会在核心安装程序设置后进行修改。

注意：如果定义非默认的安装目录，安装程序会在不创建 SystemEDGE 子文件夹的情况下将代理文件直接安装到指定的目录。

默认：C:\Program Files\CA\SystemEDGE

CASE_PUBDATADIR

指定 SystemEDGE 数据目录，将在该目录中完成所有配置并存储动态数据（在本文档中称为变量 CASYSEDGE_DATA）。代理的配置文件位于基于 SNMP_PORT 参数值的端口特定的子目录中。

默认：C:\Documents and Settings\All Users\Application Data\CA\SystemEDGE（Windows XP 和 2003），
C:\Users\Public\CA\SystemEDGE（Windows Vista 和 2008）

CASE_SNMP_PORT

指定 SystemEDGE 使用的端口。也可以使用 CASE_DEFAULT_FROM_NATIVE_SNMP 参数从本地 SNMP 代理继承该值。如果使用 CASE_SNMP_PORT 参数指定端口，将会覆盖继承的值。该端口必须唯一，否则安装会失败。如果默认端口 161 已被本地 SNMP 代理使用（并且不打算禁用该代理），必须指定其他的唯一端口（例如，1691 或 6665）。

默认：161

CASE_SNMP_SYS_DESC

指定用于填充 sysDescr MIB-II 对象的系统的信息（如系统名称）。也可以使用 CASE_DEFAULT_FROM_NATIVE_SNMP 参数从本地 SNMP 代理继承该值。如果使用 CASE_SNMP_SYS_DESC 参数指定说明，将会覆盖继承的值。

CASE_SNMP_SYS_LOC

指定用于填充 sysLocation MIB-II 对象的系统位置。也可以使用 CASE_DEFAULT_FROM_NATIVE_SNMP 参数从本地 SNMP 代理继承该值。如果使用 CASE_SNMP_SYS_LOC 参数指定位置，将会覆盖继承的值。

CASE_SNMP_SYS_CONTACT

指定用于填充 sysContact MIB-II 对象的系统联系信息。也可以使用 CASE_DEFAULT_FROM_NATIVE_SNMP 参数从本地 SNMP 代理继承该值。如果使用 CASE_SNMP_SYS_CONTACT 参数指定联系人，将会覆盖继承的值。

CASE_SNMP_READ_COMMUNITY

指定可向代理发送 GET 请求的 SNMP 读团体的名称。可以通过以分号进行分隔来指定多个团体（例如，public1;public2），也可以包含各团体的以空格分隔的 IP 地址列表以限制访问（例如，public 1.2.3.4）。也可以使用 CASE_DEFAULT_FROM_NATIVE_SNMP 参数从本地 SNMP 代理继承该值。如果使用 CASE_SNMP_READ_COMMUNITY 参数指定读团体，将会覆盖继承的值。

默认：snmp_public（仅在全新安装（不是升级）以及未指定读写团体的情况下有效）

CASE_SNMP_READ_ALLOWED_MANAGERS

指定所允许的 SNMP 管理器的以空格分隔的 IP 地址/主机名列表，以使用 SNMP_READ_COMMUNITY 查询代理。指定该项时，SNMP_READ_COMMUNITY 只能包含一个词，即 SNMP 团体。

CASE_SNMP_WRITE_COMMUNITY

指定可向代理发送 GET 和 SET 请求的 SNMP 写团体的名称。可以通过以分号进行分隔来指定多个团体（例如，rwcomm1;rwcomm2），也可以包含各团体的以空格分隔的 IP 地址列表以限制访问（例如，rwcomm1 1.2.3.4）。也可以使用 CASE_DEFAULT_FROM_NATIVE_SNMP 参数从本地 SNMP 代理继承该值。如果使用 CASE_SNMP_WRITE_COMMUNITY 参数指定写团体，将会覆盖继承的值。

CASE_SNMP_WRITE_ALLOWED_MANAGERS

指定所允许的 SNMP 管理器的以空格分隔的 IP 地址/主机名列表，以使用 SNMP_WRITE_COMMUNITY 查询代理。指定该项时，SNMP_WRITE_COMMUNITY 只能包含一个词，即 SNMP 团体。

CASE_SNMP_TRAP_COMMUNITY

指定 SNMP 陷阱团体和陷阱目标地址。可以通过以分号 (;) 进行分隔来指定多个陷阱团体设置。也可以使用

CASE_DEFAULT_FROM_NATIVE_SNMP 参数从本地 SNMP 代理继承该参数的值。如果使用 CASE_SNMP_TRAP_COMMUNITY 参数指定陷阱团体，将会覆盖继承的值。对于该参数，以下值是必需的：

- 团体名称
- 要将陷阱发送到的目标地址

对于该参数，以下值是可选的：

- 要将陷阱发送到的端口号
- 陷阱源编码选项
- 陷阱源主机名

陷阱团体设置的语法：

```
community-string {ip-address/hostname} [port [encoding [source]]]
```

示例：

```
public 1.2.3.4;public 2.3.4.5 1162;trapcom 3.4.5.6 1162 100 4.5.6.7
```

CASE_SNMP_TRAP_DESTINATION

指定要向其发送陷阱的主机名或 IP 地址。指定该项时，SNMP_TRAP_COMMUNITY 只能包含一个词（即 SNMP 团体），同时还必须指定 SNMP_TRAP_PORT。

CASE_SNMP_TRAP_PORT

指定要向其发送陷阱的目标端口号。指定该项时，SNMP_TRAP_COMMUNITY 只能包含一个词（即 SNMP 团体），同时还必须指定 SNMP_TRAP_DESTINATION。

CASE_DISABLE_NATIVE_SNMP

指定是否停止并禁用本地 SNMP 代理。

默认值：no

CASE_DEFAULT_FROM_NATIVE_SNMP

指定是否使用本地 SNMP 代理的默认 SNMP 设置。

默认值：no

CASE_MANAGER_HOSTNAME

指定要用从其管理该代理的配置管理器的主机名。为该参数输入值，可从指定的管理器配置该代理。输入星号 (*) 可接受第一个发现代理系统的管理器。该管理器将可以对代理的配置进行完全控制。默认情况下，未输入（使用）任何管理器主机，并且代理在未受管模式下运行。

CASE_MANAGER_POLICY_NAME

指定代理应使用的配置管理器策略文件的名称。为该参数输入值，可根据管理器中的现有配置文件设置 SystemEDGE 配置。默认情况下，代理使用已安装的策略文件。

CASE_START_AFTER_INSTALL

指定安装完成后是否自动启动代理。

默认值： yes

CASE_INSTALL_DOCS

指定是否要随代理一起安装 SystemEDGE 文档。

默认值： yes

CASE_LEGACY_MODE

指定是否以传统模式安装代理，该模式仅在忽略便于与 CA Virtual Assurance 一起使用的所有材料时才会安装基本代理。如果不希望将代理与 CA Virtual Assurance 一起使用，则使用传统模式进行安装。

通过重新安装或升级并指定 CASE_LEGACY_MODE=no，可以使代理进入受管模式。

默认值： no

在命令行中输入命令以及所有必需的参数和值。按 Enter 键开始安装。安装程序会检测操作系统的硬件体系结构并运行相应的安装程序版本。

注意： 如果不接受安装协议，安装将失败。

要验证是否安装，请确认“Windows 服务”对话框中是否有 CA SystemEDGE 服务、安装目录中是否有 SystemEDGE 文件或者“添加或删除程序”对话框中是否有 CA SystemEDGE Core。还可以检查安装日志文件（如果指定了该文件），以确认成功安装。

注意： 如果是在运行 Windows Vista 或更高版本的系统上进行安装，SystemEDGE 安装程序会自动安装“Microsoft Visual C++ 2005 可再发行软件包”。如果未安装该软件包，SystemEDGE 将无法正常运行。要安装“Microsoft Visual C++ 2005 可再发行软件包”，必须接受所显示的许可协议。“Microsoft Visual C++ 2005 可再发行软件包”的许可协议在交互式版本的 SystemEDGE 安装程序中被删除。

在 UNIX 和 Linux 系统上进行安装

本节介绍如何在 UNIX 和 Linux 系统上手动安装 SystemEDGE 代理。可以使用交互式向导进行安装，也可以使用命令行以无提示方式进行安装。如果 Xserver 可用且正确设置了 DISPLAY 环境，交互式向导将以文本模式显示在控制台上或显示为图形应用程序。

下列情形要求您手工安装代理，而不是使用建议的 CA Virtual Assurance 部署方法：

- 在不支持远程部署的系统上安装代理。
- 想要以传统模式安装代理。

有关支持的 UNIX 和 Linux 平台及版本的详细信息，请参阅《SystemEDGE 版本说明》。

详细信息

[以传统模式安装代理](#) (p. 100)

在 UNIX 和 Linux 系统上安装代理

可以使用交互式向导在 UNIX 和 Linux 上手动安装 SystemEDGE 代理。

注意：

- 本文档将安装目录称为 CASYSEGE，将数据目录称为 CASYSEGE_DATA。
- 安装程序不会修改 /etc/profile 中的系统环境设置。

遵循这些步骤：

1. 以 root 用户身份登录到系统并挂接 DVD2。
2. 打开终端控制台并转到 Installers/platform/Agent/SysMan/CA_SystemEDGE_Core 目录，选择与您的操作系统对应的 platform 目录。
3. 按如下方式从该目录运行安装程序：

```
sh ca-setup.sh
```

此时将显示安装程序的“简介”页面。

- 单击“下一步”。

此时会出现“许可协议”页面。

- 阅读许可协议并选择“我接受此许可协议的条款”。单击“下一步”。

此时将显示“安装类型”页面。

- 选择“典型安装”或“自定义安装”，然后单击“下一步”。

注意：该过程介绍自定义安装。如果选择“典型安装”，单击“下一步”时将显示“检查设置”页面。

此时将显示“目标位置”对话框。

- 接受默认值或进行浏览以选择安装和数据目录的位置，单击“下一步”，然后继续执行第 9 步。如果要指定其他位置，请单击“高级”并继续执行第 8 步。

目标位置

指定要安装代理的位置。默认情况下，安装目录是 `/opt/CA/SystemEDGE`，运行时程序数据存储在 `config` 子目录中。要指定更多参数，请单击“高级”。

注意：如果在已安装了代理先前版本的系统上安装该代理，安装程序会自动选择现有代理的安装目录。

如果跳过“高级”对话框，将出现“配置管理器设置”对话框。

- 填写“高级目标位置”对话框的以下字段，然后单击“下一步”。

SystemEDGE 二进制路径

为二进制程序和文档指定目录。

SystemEDGE 数据路径

为运行时程序数据指定目录。

CA 共享组件路径

为 CA 共享组件指定目录。一旦由任意 CA 软件设置该目录，就无法对其进行更改并且会禁用用户界面中的相应字段。

此时将显示“配置管理器设置”对话框。

9. 填写以下字段，然后单击“下一步”。

配置管理器主机名

指定要用从其管理该代理的配置管理器的主机名。为该参数输入值，以从运行 CA Virtual Assurance 的系统上配置该代理。输入星号 (*) 可接受第一个发现代理系统的管理器。

默认配置策略名称

指定供代理使用的策略文件(由 CA Virtual Assurance 管理器维护)的名称。为该参数输入值，以根据管理器中的现有配置文件设置 SystemEDGE 配置。

如果安装程序检测到系统上正在运行本地 SNMP 代理，将显示“本地 SNMP 选项”对话框。

10. 填写以下字段，然后单击“下一步”。

默认来自现有 SNMP 代理

指定是否使用本地 SNMP 代理的默认设置。取消选中该复选框可使用与本地 SNMP 代理不同的团体字符串和陷阱目标。

禁用本地 SNMP 代理

指定是否停止并禁用本地 SNMP 代理。如果使本地 SNMP 代理处于启用状态，则在其他端口上运行 SystemEDGE。

11. 填写以下字段，然后单击“下一步”。

SNMP 端口号

指定要在其上运行 SystemEDGE 代理的端口。该端口必须未被任何其他应用程序使用，否则安装会失败。如果本地 SNMP 代理已使用默认端口，请指定其他端口（例如，1691 或 6665）。

默认值：161

此时将显示“SNMP 系统信息”对话框。

12. 填写以下字段，然后单击“下一步”。

系统说明

指定用于填充 sysDescr MIB-II 对象的系统的信息（如系统名称）。

系统位置

指定用于填充 sysLocation MIB-II 对象的系统位置。

系统联系人

指定用于填充 sysContact MIB-II 对象的系统联系人信息。

此时将显示“SNMP 团体设置”对话框。

13. 填写以下字段，单击“下一步”，然后继续执行第 15 步。如果要指定多个团体字符串，请单击“高级”并继续执行第 14 步。

只读团体

指定 SNMP 只读团体字符串。

默认值：public

读写团体

指定 SNMP 读写团体字符串。

如果跳过“高级”对话框，将显示“SNMP 陷阱设置”对话框。

14. 填写“SNMP 团体设置 - 高级”对话框中的以下字段，然后单击“下一步”。

只读团体

指定 SNMP 只读团体字符串。可以通过以分号分隔各个团体来指定多个团体（例如，public1;public2）。还可以包括各团体的 IP 地址列表以限制访问（例如，public 1.2.3.4）。

默认值：public

读写团体

指定 SNMP 读写团体字符串。可以通过以分号分隔各个团体来指定多个团体（例如，rwcomm1;rwcomm2）。还可以包括各团体以空格分隔的 IP 地址列表以限制访问（例如，rwcomm1 1.2.3.4）。读写团体字符串对于正确操作某些 AIM（例如，RM）和一些远程使用（例如，创建监视器）是必需的。

此时将显示“SNMP 陷阱设置”对话框。

15. 填写以下字段，单击“下一步”，然后继续执行第 17 步。如果要指定多个陷阱目标，请单击“高级”并继续执行第 16 步。

陷阱团体字符串

指定在发送陷阱消息中编码的 SNMP 团体。

默认值：public

目标主机

指定陷阱消息的目标。

默认：在“配置管理器设置”对话框中设置的配置管理器主机名。

端口号

指定陷阱消息发送到的端口。

默认：162。

如果跳过“高级”对话框，将显示“权限分隔用户”对话框。

16. 填写“SNMP 陷阱设置 - 高级”对话框中的以下字段，然后单击“下一步”。

陷阱配置

指定一个或多个陷阱目标。可以通过以分号分隔各个条目来指定多个条目（例如，`public server1;public server2 1162`）。

此时将显示“权限分隔用户”对话框。

17. 填写以下字段，然后单击“下一步”：

用户名

指定用户名，在 SNMP 通信期间代理使用其凭据运行。

该条目会指示代理（仅限 UNIX）在其他用户帐户下运行 SNMP 通信。此代理还会将该用户的默认组用作有效组。

默认值：代理使用 root 帐户运行。

此时将显示“杂项设置”对话框。

18. 填写下列字段，然后单击“下一步”：

安装之后启动

指定安装结束时是否启动代理。

安装文档

指定是否安装文档。

此时将显示“检查设置”页面。

19. 检查安装设置并单击“安装”。

安装完成后将显示“安装完成”页面。

20. 单击“完成”。

安装即会完成。

在 64 位 Linux 版本上安装 SystemEDGE 失败

症状:

在 64 位 Linux 版本上安装 SystemEDGE 时，安装失败。

解决方案:

要在 64 位 Linux 版本上运行并安装 SystemEDGE，请安装必需的 32 位库:

- 在 RedHat 或 SuSE 发行版上有效:

```
yum install glibc.i686
```

- 在 Debian 发行版上有效:

```
apt-get install ia32-libs
```

在 UNIX 上从命令行安装代理

可以使用命令行版本的安装程序安装 SystemEDGE UNIX 软件包。从命令行安装时，使用参数设置各种安装属性。可从命令行执行以下操作:

- 在预先填写安装参数或不预先填写安装参数的情况下启动安装向导
- 指定安装参数并在不使用交互式向导的情况下运行安装程序

注意:

- UNIX 不支持自动创建响应文件来运行无提示安装。但是，可以手动创建响应文件，并使用该文件进行无人照管的安装。
- 安装程序不会修改 `/etc/profile` 中的系统环境设置。

该过程适用第二种方案：从命令行运行无人照管的安装。

遵循这些步骤：

1. 以 root 用户身份登录到系统。
2. 导航到 `DVD2/Installers/platform/Agent/SysMan/CA_SystemEDGE_Core` 文件夹（指定与您的操作系统对应的 `platform` 文件夹），然后输入以下所需参数（在完成第 3 步之前，请勿运行该命令）：

```
sh ca-setup.sh CA_SETUP_MODE="UNATTENDED" EULA_ACCEPTED="yes" [parameter]
```

注意：有关安装参数的帮助，请在命令提示符下输入 `ca-setup -?` 在命令提示符中。

CA_SETUP_MODE

指定安装模式。将该参数设置为 `UNATTENDED` 以在不显示安装向导的情况下运行安装。如果省略该参数，则运行该命令后，安装向导打开时会预先填写指定的参数值。

EULA_ACCEPTED

读取许可协议并指定是否接受该许可协议。如果在将安装模式设置为 `UNATTENDED` 时省略该参数或将其设置为除 “**YES**” 以外的任何值，安装将失败。交互式安装中不需要该参数。

3. 根据需要添加可选参数并运行命令。

注意：如果省略以下可选参数，则无需提供这些参数的值：

```
CA_SETUP_LOG_FILE
CA_SETUP_VERBOSE
CASE_INSTALLDIR
CASE_PUBDATADIR
CASE_SNMP_PORT
CASE_SNMP_SYS_DESC
CASE_SNMP_SYS_LOC
CASE_SNMP_SYS_CONTACT
CASE_SNMP_READ_COMMUNITY
CASE_SNMP_READ_ALLOWED_MANAGERS
CASE_SNMP_WRITE_COMMUNITY
CASE_SNMP_WRITE_ALLOWED_MANAGERS
CASE_SNMP_TRAP_COMMUNITY
CASE_SNMP_TRAP_DESTINATION
CASE_SNMP_TRAP_PORT
CASE_DISABLE_NATIVE_SNMP
CASE_DEFAULT_FROM_NATIVE_SNMP
CASE_MANAGER_HOSTNAME
CASE_MANAGER_POLICY_NAME
CASE_PRIVSEP_USER
CASE_START_AFTER_INSTALL
CASE_INSTALL_DOCS
CASE_LEGACY_MODE
```

CA_SETUP_LOG_FILE

指定记录安装消息的位置和文件名。默认情况下，消息将记录到 `/opt/CA/installer/log/CA_SETUP_PACKAGE_NAME.log` 中。

CA_SETUP_VERBOSE

设置为 “yes” 时将打开详细安装模式。详细模式会将更多信息记录到安装日志文件中。

CASE_INSTALLDIR

指定 SystemEDGE 安装目录。该目录中包含与 SystemEDGE 相关的所有内容（如 AIM 和高级加密），并且绝不会在核心安装程序设置后进行修改。

注意：如果定义非默认的安装目录，安装程序会在不创建 SystemEDGE 子文件夹的情况下将代理文件直接安装到指定的目录。

默认： `/opt/CA/SystemEDGE`

CASE_PUBDATADIR

指定 SystemEDGE 数据目录，将在该目录中完成所有配置并存储动态数据（在本文档中称为变量 `CASYSEDGE_DATA`）。代理的配置文件位于基于 `SNMP_PORT` 参数值的端口特定的子目录中。

默认： `/opt/CA/SystemEDGE/config`

CASE_SNMP_PORT

指定 SystemEDGE 使用的端口。也可以使用

`CASE_DEFAULT_FROM_NATIVE_SNMP` 参数从本地 SNMP 代理继承该值。如果使用 `CASE_SNMP_PORT` 参数指定端口，将会覆盖继承的值。该端口必须唯一，否则安装会失败。如果默认端口 161 已被本地 SNMP 代理使用（并且不打算禁用该代理），必须指定其他的唯一端口（例如，1691 或 6665）。

默认： 161

CASE_SNMP_SYS_DESC

指定用于填充 `sysDescr` MIB-II 对象的系统的信息（如系统名称）。也可以使用 `CASE_DEFAULT_FROM_NATIVE_SNMP` 参数从本地 SNMP 代理继承该值。如果使用 `CASE_SNMP_SYS_DESC` 参数指定说明，将会覆盖继承的值。

CASE_SNMP_SYS_LOC

指定用于填充 `sysLocation` MIB-II 对象的系统位置。也可以使用 `CASE_DEFAULT_FROM_NATIVE_SNMP` 参数从本地 SNMP 代理继承该值。如果使用 `CASE_SNMP_SYS_LOC` 参数指定位置，将会覆盖继承的值。

CASE_SNMP_SYS_CONTACT

指定用于填充 sysContact MIB-II 对象的系统联系信息。也可以使用 CASE_DEFAULT_FROM_NATIVE_SNMP 参数从本地 SNMP 代理继承该值。如果使用 CASE_SNMP_SYS_CONTACT 参数指定联系人，将会覆盖继承的值。

CASE_SNMP_READ_COMMUNITY

指定可向代理发送 GET 请求的 SNMP 读团体的名称。可以通过以分号进行分隔来指定多个团体（例如，public1;public2），也可以包含各团体的以空格分隔的 IP 地址列表以限制访问（例如，public 1.2.3.4）。也可以使用 CASE_DEFAULT_FROM_NATIVE_SNMP 参数从本地 SNMP 代理继承该值。如果使用 CASE_SNMP_READ_COMMUNITY 参数指定读团体，将会覆盖继承的值。

默认：snmp_public（仅在全新安装（不是升级）以及未指定读写团体的情况下有效）

CASE_SNMP_READ_ALLOWED_MANAGERS

指定所允许的 SNMP 管理器的以空格分隔的 IP 地址/主机名列表，以使用 CASE_SNMP_READ_COMMUNITY 查询代理。指定该项时，CASE_SNMP_READ_COMMUNITY 只能包含一个词，即 SNMP 团体。

CASE_SNMP_WRITE_COMMUNITY

指定可向代理发送 GET 和 SET 请求的 SNMP 写团体的名称。可以通过以分号进行分隔来指定多个团体（例如，rwcomm1;rwcomm2），也可以包含各团体的以空格分隔的 IP 地址列表以限制访问（例如，rwcomm1 1.2.3.4）。也可以使用 CASE_DEFAULT_FROM_NATIVE_SNMP 参数从本地 SNMP 代理继承该值。如果使用 CASE_SNMP_WRITE_COMMUNITY 参数指定写团体，将会覆盖继承的值。

CASE_SNMP_WRITE_ALLOWED_MANAGERS

指定所允许的 SNMP 管理器的以空格分隔的 IP 地址/主机名列表，以使用 CASE_SNMP_WRITE_COMMUNITY 查询代理。指定该项时，CASE_SNMP_WRITE_COMMUNITY 只能包含一个词，即 SNMP 团体。

CASE_SNMP_TRAP_COMMUNITY

指定 SNMP 陷阱团体和陷阱目标地址。可以通过以分号 (;) 进行分隔来指定多个陷阱团体设置。也可以使用

CASE_DEFAULT_FROM_NATIVE_SNMP 参数从本地 SNMP 代理继承该参数的值。如果使用 CASE_SNMP_TRAP_COMMUNITY 参数指定陷阱团体，将会覆盖继承的值。对于该参数，以下值是必需的：

- 团体名称
- 要将陷阱发送到的目标地址

对于该参数，以下值是可选的：

- 要将陷阱发送到的端口号
- 陷阱源编码选项
- 陷阱源主机名

陷阱团体设置的语法：

```
community-string {ip-address/hostname} [port [encoding [source]]]
```

示例：

```
public 1.2.3.4;public 2.3.4.5 1162;trapcom 3.4.5.6 1162 100 4.5.6.7
```

CASE_SNMP_TRAP_DESTINATION

指定要向其发送陷阱的主机名或 IP 地址。指定该项时，CASE_SNMP_TRAP_COMMUNITY 只能包含一个词（即 SNMP 团体），同时还必须指定 CASE_SNMP_TRAP_PORT。

CASE_SNMP_TRAP_PORT

指定要向其发送陷阱的目标端口号。指定该项时，CASE_SNMP_TRAP_COMMUNITY 只能包含一个词（即 SNMP 团体），同时还必须指定 CASE_SNMP_TRAP_DESTINATION。

CASE_DISABLE_NATIVE_SNMP

指定是否停止并禁用本地 SNMP 代理。

默认值：no

CASE_DEFAULT_FROM_NATIVE_SNMP

指定是否使用本地 SNMP 代理的默认 SNMP 设置。

默认值：no

CASE_MANAGER_HOSTNAME

指定要用从其管理该代理的配置管理器的主机名。为该参数输入值，可从指定的管理器配置该代理。输入星号 (*) 可接受第一个发现代理系统的管理器。该管理器将可以对代理的配置进行完全控制。默认情况下，未输入（使用）任何管理器主机，并且代理在未受管模式下运行。

CASE_MANAGER_POLICY_NAME

指定代理应使用的配置管理器策略文件的名称。为该参数输入值，可根据管理器中的现有配置文件设置 SystemEDGE 配置。默认情况下，代理使用已安装的策略文件。

CASE_START_AFTER_INSTALL

指定安装完成后是否自动启动代理。有效值包括：Yes、No 和 PRESERVE。如果安装开始时代理正在运行，此时要升级为仅启动代理，可以使用 PRESERVE。

默认值： PRESERVE

CASE_PRIVSEP_USER

指定用户名，在 SNMP 通信期间代理使用其凭据运行。

该条目会指示代理（仅限 UNIX）在其他用户帐户下运行 SNMP 通信。此代理还会将该用户的默认组用作有效组。

默认值： 代理使用 root 帐户运行。

CASE_INSTALL_DOCS

指定是否要随代理一起安装 SystemEDGE 文档。

默认值： yes

CASE_LEGACY_MODE

指定是否以传统模式安装代理，该模式仅在忽略便于与 CA Virtual Assurance 一起使用的所有材料时才会安装基本代理。如果不希望将代理与 CA Virtual Assurance 一起使用，则使用传统模式进行安装。

通过重新安装或升级并指定 CASE_LEGACY_MODE=no，可以使代理进入受管模式。

默认值： no

安装开始。如果不接受许可协议，安装将失败。

/opt/CA/installer/log/\$CA_SETUP_PACKAGE_NAME.log 中始终存在 lsm 安装程序日志文件，可以使用该文件检查安装是否成功。

如果要验证是否安装，请确认安装目录中是否有 SystemEDGE 文件。

\$CASYSEDGE 变量的传统支持

Linux/UNIX 上的安装程序不会在 `/etc/profile` 中修改系统环境设置。因此，`$CASYSEDGE` 变量不再可用。

如果您的环境必须支持 `$CASYSEDGE`，请执行以下选项之一：

- 在 shell 中创建 `$CASYSEDGE`。

在 `sh`、`ksh` 或 `bash` shell 中运行以下命令：

```
./etc/profile.CA
```

如果您使用 `csh`，请运行以下命令：

```
source /etc/csh_login.CA
```

- 强制安装程序修改 `/etc/profile` 并创建 `$CASYSEDGE`。

打开 `shell` 并输入以下命令：

```
Update_Profile=1;export Update_Profile  
sh ca-setup.sh
```

如果您使用 `csh`，请运行以下命令：

```
setenv Update_Profile 1  
sh ca-setup.sh
```

安装程序将安装 `SystemEDGE` 并在环境中创建 `$CASYSEDGE`。

在升级到版本 5.7.1 期间，`SystemEDGE` 安装程序不会更改现有环境。先前创建的 `$CASYSEDGE` 变量依然在环境中。

配置和使用响应文件

可以创建响应文件，以便在没有用户交互的情况下运行无提示安装。使用响应文件的效果与在命令行中指定 `CA_SETUP_MODE=UNATTENDED` 相同。响应文件会使安装进入无提示、无人照管模式。

安装程序使用响应文件中的属性，在不提示用户进行输入的情况下安装代理。

遵循这些步骤:

1. 以管理员或 `root` 用户身份登录到计算机系统。
2. 根据在“从命令行安装代理”部分中指定的参数创建响应文件。响应文件是包含如下所示参数设置的文本文件：

```
parameter1=value1
parameter2=value2
...
```

3. 导航到 `DVDdrive\Installers\OperatingSystem\Agent\SysMan\CA_SystemEDGE_Core` 目录，然后根据您的操作系统输入以下命令之一：

```
ca-setup CA_SETUP_RESPONSE_FILE="<响应文件的名称>" (Windows)
```

```
sh ca-setup.sh CA_SETUP_RESPONSE_FILE="<响应文件的名称>" (UNIX、Linux)
```

CA_SETUP_RESPONSE_FILE

指定响应文件的路径和名称。

安装将使用响应文件中的设置以无提示方式运行。

以传统模式安装代理

忽略便于与 `CA Virtual Assurance` 一起使用的以下组件时，可以在传统模式下安装 `SystemEDGE` 以获得所有基本代理功能。

- `CAM`，允许通过 `CA Virtual Assurance` 进行远程配置。
- `IDPrimer`，允许通过 `CA Virtual Assurance` 进行远程部署。

仅当您不打算使用 `CA Virtual Assurance` 或类似管理应用程序对 `SystemEDGE` 进行管理时，才以传统模式安装 `SystemEDGE`。如果以传统模式安装代理并且希望以后使用 `CA Virtual Assurance` 对其进行管理，您可以升级该代理。

注意：当您升级以前版本的 `SystemEDGE` 时，它会自动升级到完全代理，除非您另外指定其他代理。

遵循这些步骤:

1. 将 CA_SystemEDGE_Core 目录从安装介质复制到硬盘。
2. 转到 CA_SystemEDGE_Core 目录并使用 ASCII 编辑器打开 ca-setup.dat。
3. 编辑 ca-setup.dat，设置 CASE_LEGACY_MODE=yes。
4. 保存 ca-setup.dat。
5. 如[在 Windows 上安装代理](#) (p. 77)或[在 UNIX 上安装代理](#) (p. 88)中所述运行安装。
6. 完成安装。

要以传统模式从命令行安装代理，请在 `ca-setup` 命令中包括以下参数：

```
CASE_LEGACY_MODE="yes"
```

代理部署

CA Virtual Assurance 提供了全面的解决方案，用于将 SystemEDGE 代理远程部署到所有受管系统。您可以基于提供的包含自定义安装参数的软件包创建部署模板，并同时将这些模板部署到多个受管系统。该自动化部署解决方案提供了一个可在整个企业中部署和配置代理的位置。

CA Virtual Assurance 提供了以下基本部署软件包：

- SystemEDGE 代理核心
- SystemEDGE 高级加密
- CA Citrix XenServer AIM
- SystemEDGE 远程监控 AIM
- SystemEDGE 服务响应监视器 AIM
- CA Systems Performance LiteAgent
- CA IBM LPAR AIM
- CA IBM High Availability Cluster Multiprocessing AIM
- CA KVM AIM（不适用于 CA Server Automation）
- CA Cisco UCS AIM
- CA Microsoft Hyper-V AIM
- CA Microsoft 群集服务支持 AIM
- CA Solaris Zones AIM
- CA VMware vCenter 服务器 AIM
- CA VMware vCloud AIM
- CA Exchange Server 和 Active Directory AIM（不适用于 CA Server Automation）

CA Virtual Assurance 支持以下 SystemEDGE 部署方案：

- 您可以将 SystemEDGE 版本 5.7.1 代理部署到没有预先存在的 SystemEDGE 代理的系统中。
- 您可以将 SystemEDGE 版本 5.7.1 代理部署到具有预先存在的 SystemEDGE 4.3 代理的系统中。部署会自动将现有代理升级到版本 5.7.1。
- 您可以对单个系统或通过配置模板进行配置更改。使用后面的选项可对现有的受管 SystemEDGE 代理或一组受管 SystemEDGE 代理应用更改。

注意：有关如何进行部署的详细信息，请参阅 *CA Virtual Assurance 联机帮助* 和 *《管理指南》*。

CA Virtual Assurance 不支持以下 SystemEDGE 部署方案：

- 您不能部署早于 SystemEDGE 5.0.0 的代理版本。
- 您不能在 CA Virtual Assurance 管理器系统上部署 SystemEDGE 代理。代理会通过 CA Virtual Assurance 管理器安装进行自动安装。

注意：有关部署支持的详细信息，请参阅 *《CA Virtual Assurance 管理指南》*。

使用 CA Virtual Assurance 部署代理

《实施指南》和联机帮助详细介绍了如何从用户界面部署代理软件包。

要将代理部署到系统，请创建部署作业。部署作业包含 CA Virtual Assurance 将部署软件包适时传送给相应系统所需的详细信息。

遵循这些步骤：

1. 依次选择“资源”、“部署”。
“部署”窗格显示“软件包”、“模板”和“作业”。
2. 在“管理资源”窗格中右键单击“作业”文件夹，然后选择“创建新作业”。还可以选择“作业”文件夹，然后单击“作业状态”工具栏上的+（新建）。
此时将显示“作业设置”页面。
3. 在“作业名称”窗格中输入名称，可以选择使该作业基于现有的模板，然后单击“下一步”。
此时将显示“软件包选择”页面。
4. 选择平台和希望部署的软件包。

5. （可选）单击“详细信息”选项卡。

此时将显示“软件包打包程序详细信息”对话框，您可以使用该对话框以内联方式编辑软件包属性。如果软件包打包程序处于不完整或无效状态，可以通过内联编辑修改字段。

- a. 单击“编辑”并修改软件包打包程序属性。
- b. 单击“保存”，然后单击“确定”。

软件包打包程序属性即已更新。

6. 单击向下箭头将软件包打包程序添加到作业中，然后单击“下一步”。

此时将显示“计算机选择”页面。

7. 选择要部署到的系统，然后单击“下一步”。如果您的环境中有许多服务器，则需要多个包含一些条目的页面才能列出所有服务器。在页面上选择服务器并滚动到下一页时，在前一页上选择的所有服务器仍是有效的。

此时将显示“已选择计算机”页面。

8. 单击“设置凭据”，设置建立连接所需的系统凭据，然后单击“下一步”。

注意：使用域凭据部署到 Windows 目标系统必须采用“域\用户名”形式。

此时将显示“高级”页。

9. （可选）设置分发服务器来管理部署。如果不设置，则将进行自动选择。

10. 为该作业选择排定选项：

立即交付

在创建新部署作业后立即启动作业。“立即交付”是默认选项。

交错交付

在特定时间段内交付软件包。

排定交付

排定未来特定时间的部署。

11. （可选）如果先前已使用该部署基础架构将某个软件包成功部署到系统，您可以强制是否再次运行。

12. 单击“下一步”。

此时将显示“摘要”页面。

13. 检查作业的详细信息，然后单击“部署”。

部署作业创建完成。

注意：在创建作业之后，您可以将该作业另存为模板。模板可以省去软件包和机器的选择，因此可以在后续作业中轻松复用。有关更详细的信息，请参阅联机帮助和《实施指南》。

卸载 SystemEDGE

本节介绍如何删除与 SystemEDGE 代理相关联的文件和子目录。

在 Windows 上卸载 SystemEDGE 和 AIM

卸载程序可从系统中删除 SystemEDGE。您可以指定是否从数据目录中删除配置数据。您可以从 Windows 的“添加或删除程序”窗口或从命令行卸载代理和 AIM。

卸载时请考虑以下依存关系：

- 虚拟环境的 AIM，RM AIM 和 SRM AIM 依存于高级加密和 SystemEDGE。
- 高级加密依赖于 SystemEDGE。

基于这些依存关系，卸载顺序如下所示：

1. RM AIM、SRM AIM 或虚拟环境的 AIM
2. 高级加密
3. SystemEDGE 核心

如果使用任何其它顺序，则无法完成该卸载。删除最初通过远程部署安装的组件时，不会卸载 Idprimer 和 CAM。

从 Windows 的“添加或删除程序”窗口卸载 SystemEDGE 或 AIM

1. 选择“开始”、“设置”、“控制面板”、“添加或删除程序”。

显示“添加或删除程序”窗口并列出以下组件：

- 虚拟环境的 AIM
- CA SystemEDGE Core
- CA SystemEDGE AdvancedEncryption
- CA SystemEDGE RM
- CA SystemEDGE SRM

2. 根据卸载顺序，右键单击相应的组件并选择“卸载”。

卸载 SystemEDGE 时，会显示一个对话框，提示您指定是否保留配置文件。

3. 单击“是”或“否”。

对话框中会显示此卸载过程。卸载完成时，对话框将关闭。

从命令行卸载 SystemEDGE 或 AIM

1. 打开命令提示符并转到 DVD1\Installers\Windows\Agent\SysMan 目录。此目录包含以下子目录：

- CA_SystemEDGE_SRM
- CA_SystemEDGE_RM
- CA_SystemEDGE_AdvancedEncryption
- CA_SystemEDGE_Core

2. 运行下列命令：

```
ca-setup -x
```

卸载 SystemEDGE 时，会显示一个对话框，提示您指定是否保留配置文件。

3. 单击“是”或“否”。

对话框中会显示此卸载过程。卸载完成时，对话框将关闭。

无提示卸载 SystemEDGE 或 AIM

- 请执行以下步骤来卸载 SystemEDGE：

- a. 打开“命令提示符”窗口，并转到以下目录路径：

DVD1\Installers\Windows\Agent\SysMan\CA_SystemEDGE_Core

- b. 运行下列命令：

```
ca-setup.exe CA_SETUP_MODE=UNINSTALL CASE_KEEP_DATA=[YES|NO]
```

SystemEDGE 已从计算机上卸载。

注意：要进行验证，请转到“控制面板”，并查看是否已从控制面板项中删除了 SystemEDGE。

- 请执行以下步骤来删除 AIM：

- a. 打开“命令提示符”窗口，并转到相应的 AIM 目录路径：

DVD1\Installers\Windows\Agent\SysMan\AIM

- b. 运行下列命令：

```
ca-setup.exe CA_SETUP_MODE=UNINSTALL
```

AIM 已从计算机上卸载。

注意：导航到“程序文件”、“CA”、“SystemEDGE”、“插件”，并验证是否已删除 AIM 文件夹。

在 UNIX 系统上卸载 SystemEDGE 和 AIM

卸载程序可从系统中删除 SystemEDGE 或 AIM。您可以为 SystemEDGE 指定是否从数据目录中删除配置数据。

卸载时请考虑以下依存关系：

- SRM AIM 依存于高级加密和 SystemEDGE。
- 高级加密依赖于 SystemEDGE。

基于这些依存关系，卸载顺序如下所示：

1. SRM AIM
2. 高级加密
3. SystemEDGE 核心

如果使用任何其它顺序，则无法完成该卸载。删除最初通过远程部署安装的组件时，不会卸载 Idprimer 和 CAM。

使用 ca-setup.sh 卸载代理或 AIM

1. 打开终端控制台并以 root (su) 登录。
2. 转到 DVD2/Installers/<Platform>/Agent/SysMan 目录。此目录包含以下子目录：

- CA_SystemEDGE_SRM
- CA_SystemEDGE_AdvancedEncryption
- CA_SystemEDGE_Core

3. 转到相应目录并运行以下命令：

```
sh ca-setup.sh -x
```

此时会显示一个对话框，提示您指定是否保留配置文件。

4. 单击“是”或“否”。

对话框中会显示此卸载过程。卸载完成时，对话框将关闭。

使用 lsm 卸载代理或 AIM

1. 打开终端控制台并以 root (su) 登录。
2. 运行以下列表中相应的命令：

```
lsm -e CA_SystemEDGE_SRM  
lsm -e CA_SystemEDGE_AdvancedEncryption  
lsm -e CA_SystemEDGE_Core
```

卸载 SystemEDGE 时，系统会提示您指定是否保留配置文件。

3. 单击“是”或“否”。

卸载完成。

无提示卸载代理或 AIM

- 请执行以下步骤来卸载代理：

- a. 打开终端控制台并以 root (su) 登录。转到以下目录路径：

DVD2/Installers/Platform/Agent/SysMan/CA_SystemEDGE_Core

- b. 运行下列命令：

```
sh ca-setup.sh CA_SETUP_MODE=UNATTENDED_UNINSTALL  
CASE_KEEP_DATA=[YES|NO]
```

代理已从计算机上卸载。

- 请执行以下步骤来卸载 AIM：

- a. 打开终端控制台并以 root (su) 登录。转到相应的 AIM 目录路径：

DVD2/Installers/Platform/Agent/SysMan/AIM

- b. 运行下列命令：

```
sh ca-setup.sh CA_SETUP_MODE=UNATTENDED_UNINSTALL
```

AIM 已从计算机上卸载。

重新安装

在您重新安装 SystemEDGE 时，安装程序仅使用以下参数：

CASE_START_AFTER_INSTALL

CASE_SNMP_PORT

CASE_INSTALL_DOCS

忽略其他参数。

升级受管节点和 AIM 服务器

建议的方法是远程部署升级受管节点和远程 AIM 服务器。远程部署允许您升级 SystemEDGE、高级加密和 AIM。

如果系统不支持 SystemEDGE 的受管模式，则您无法使用远程部署。我们建议使用以下选项之一：

- 从安装介质进行手动安装
- 例如，SystemEDGE、高级加密和使用 SSH 的 SRM AIM 的远程安装

请执行以下步骤：

[代理和 AIM 升级](#) (p. 109)

[将 SystemEDGE 监视器导入策略](#) (p. 112)

代理和 AIM 升级

您可以从 SystemEDGE 的任何可升级版本升级到 SystemEDGE 5.7.1 版：

- 4.3.4 及更高版本 (4.3.x)
- 5.1.0 及更高版本 (5.1.x)
- 5.6.0 及更高版本 (5.6.x)
- 5.7.0 及更高版本 (5.7.x)

安装程序会在原始安装路径的根文件夹中创建升级的目录，并将之前版本中的配置文件复制到其中。在代理首次启动时，它会将配置数据迁移到新建的数据目录。将 SystemEDGE 升级到版本 5.7.1 时，高级加密以及所有相应的 AIM 将升级到最新版本。

注意：SystemEDGE 5.7.1 版不会加载先前版本的 CA Virtual Assurance 的 AIM。

您可以通过以下方式之一升级 SystemEDGE：

- 通过 CA Virtual Assurance 远程部署到已安装可升级代理的系统。

注意：如果您想要升级 SystemEDGE、高级加密和 AIM，请使用单个远程部署作业进行升级。将 SystemEDGE、高级加密和所有必需的 AIM 添加到远程部署作业。SystemEDGE 仅加载级别为 5.7.1 的 AIM，或者类似 iddmod 的传统 AIM。也就是说，将隔离 5.7 或更低级别的 AIM。

有关远程部署的详细信息，请参阅《*管理指南*》中的“远程部署”一章。

- Windows 上的自定义管理器安装。该选项升级所有已安装的组件。选择要安装的任何其他代理和 AIM。
- 在已安装可升级代理的系统上的手动安装。
要执行升级，如[在 Windows 上安装代理](#) (p. 77)或[在 UNIX 上安装代理](#) (p. 88)中所述安装代理。

如果不支持升级较旧的代理版本，请执行以下操作：

- 对于 SystemEDGE 的早期版本，请先卸载代理，然后安装 SystemEDGE 版本 5.7.1。您可以在卸载之前保存所有配置数据，并将这些数据应用于 SystemEDGE 版本 5.7.1。

详细信息：

[从 SystemEDGE 4.3.4 升级代理](#) (p. 110)

从 SystemEDGE 4.3.4 升级代理

从以前的 4.3.4 版本升级配置文件时，安装程序将执行以下操作。

- 安装程序检测以下文件的以前版本并将其复制到 CASYSEDGE\upgraded 目录：
 - system32\sysedge.cf (Windows) 和 /etc/sysedge.cf (UNIX)
 - CASYSEDGE\config\sysedgeV3.cf (Windows 和 UNIX)
 - system32\sysedge.mon (Windows) 和 /etc/sysedge.mon (UNIX)
 - system32\sysedge.lic (Windows) 和 /etc/sysedge.lic (UNIX)
 - CASYSEDGE\plugins\monwin\monwin.cf (Windows 和 UNIX)
- 当代理首次启动时，它会直接把以下文件复制到 CASYSEDGE_DATADIR 目录：
 - CASYSEDGE\config\sysedge.cf (Windows 和 UNIX)
 - CASYSEDGE\config\sysedgeV3.cf (Windows 和 UNIX)
- 作为 sysedge.cf 复制操作的一部分，代理会将升级目录中的 sysedge.cf、sysedge.mon 和 monwin.cf 文件中的配置数据迁移到数据目录中的新 sysedge.cf 文件。
- 作为 sysedgeV3.cf 复制操作的一部分，代理会将升级目录中的 sysedgeV3.cf 文件中的配置数据迁移到数据目录中的新 sysedgeV3.cf 文件。

注意：数据目录存储用于运行时配置更改的 sysedge.cf 文件的版本。

当从以前的版本升级时，代理会自动对配置文件设置进行以下更改：

- `procAlive` 进程监视器语法已更改，以遵循阈值进程监视器语法。代理会自动将任何现有的 `procAlive` 条目转换为新格式。

注意：有关 `procAlive` 条目的语法的更多信息，请参阅“*进程和服务监控*”一章。

- 禁用 `no_process_sets` 和 `no_remoteshell_group` 参数将允许对只有一个 SNMP 写团体的代理系统进行关键访问。因此，代理不迁移这些参数以前的设置并且总是启用它们。
- `sysedge_memory` 参数不再存在，在升级的 `sysedge.cf` 文件中已删除。现在通过在 `sysedge.mon` 文件中存储任何监视器的当前状态来处理显著的警报。现有条目被迁移到新警报处理配置中。
- `Tc_publish` 参数被重命名为 `no_trapcommunity_table`（具有反向逻辑）并在默认情况下启用。
- 当您使用远程部署从 SystemEDGE 版本 4.3.4 升级到 SystemEDGE 版本 5.7.1 时，迁移的监视器将存储在 `sysedge.cf.bak` 文件中。
- 根据您的部署策略，可以使用策略配置来在环境中应用迁移的监视器。

注意：有关详细信息，请参阅 CA Virtual Assurance 总目录。

- 当您在本地系统上从 SystemEDGE 版本 4.3.4 升级到 SystemEDGE 版本 5.7.1 时，安装程序会迁移监视器并在新配置中提供它们。

所有以前的语法均与新版本的代理兼容。但是，可提供许多用于增强代理功能的新选项。我们建议您检查较旧的监视器条目并编辑语法来包含附加选项。

注意：在从 4.3.4 版或更高版本 (4.3.x) 升级 SystemEDGE 时，安装程序将仅使用以下参数：

```
CASE_PUBDATADIR
CASE_MANAGER_HOSTNAME
CASE_MANAGER_POLICY_NAME
CASE_START_AFTER_INSTALL
CASE_LEGACY_MODE
CASE_SNMP_PORT
CASE_INSTALL_DOCS
CASE_SNMP_TRAP_COMMUNITY (1)
CASE_SNMP_TRAP_DESTINATION (1)
CASE_SNMP_TRAP_PORT (1)
CASE_SNMP_READ_COMMUNITY (1)
CASE_SNMP_WRITE_COMMUNITY (1)
CASE_SNMP_READ_ALLOWED_MANAGERS (1)
CASE_SNMP_WRITE_ALLOWED_MANAGERS (1)
```

忽略其他参数。

(1) 这些参数很特殊。它们的设置已附加到现有的 SystemEDGE 4.x 设置，允许 SystemEDGE 4.x 管理器和 SystemEDGE 5.x 管理器运行。

将 SystemEDGE 监视器导入策略

在升级代理之前，请验证是否在“管理”选项卡下的“SNMP”中指定了 SystemEDGE 代理使用的 SNMP 团体字符串。也可以在“策略”、“计算机”、“度量标准”、“SNMP”下逐个计算机指定团体字符串。SystemEDGE 安装程序将 SystemEDGE 文件移至“upgraded”目录，代理将在升级后进行读取。将保留 OID。策略配置只能从现有的 sysedge.cf 文件中读取条目。不会导入 sysedge.mon 文件。

远程部署会将监视器的原始 OID 导入实例。

遵循这些步骤：

1. 验证代理的端口的“管理”选项卡下的“SNMP”中是否包括 SystemEDGE 代理团体字符串。
2. 将 SystemEDGE 升级到版本 5.7.1。
3. 验证是否已正确发现 SystemEDGE 5.7.1（“资源”选项卡）且其包含策略。

4. 单击“资源”选项卡，打开“配置”窗格，展开“策略”，然后单击“SystemEDGE”。

此时将显示“SystemEDGE”窗格。

5. 在“可用策略”工具栏上单击+（新建）。

此时将显示“新建 SystemEDGE 策略”对话框。

6. 输入策略的名称，然后单击“导入...”。

此时将显示“SystemEDGE 代理计算机”对话框。

7. 选择已升级的服务器，然后单击“确定”。

将关闭“SystemEDGE 代理计算机”对话框。

8. 在“新建 SystemEDGE”对话框中，单击“确定”。

CA Virtual Assurance 将创建一个包含已导入的监视器的策略。

您可以编辑或更新监视器，并使用 SystemEDGE 版本 5.7.1 状态的模型。

第 5 章：代理配置

本章描述如何设置决定 SystemEDGE 代理操作方式的配置参数。配置文件 `sysedge.cf` 和 `sysedgeV3.cf` 中定义了这些配置参数。

此部分包含以下主题：

[配置基础](#) (p. 115)

[sysedge.cf 和 sysedge.mon 之间的交互](#) (p. 117)

[使用 CA Virtual Assurance 的配置](#) (p. 117)

[File-Based 配置](#) (p. 117)

[监视器和自动监测器的正则表达式示例](#) (p. 142)

[Perl 兼容的正则表达式 \(PCRE\) 支持](#) (p. 145)

[SNMPv3 配置](#) (p. 150)

[SystemEDGE Control Panel for Windows](#) (p. 159)

配置基础

配置文件 `sysedge.cf` 和 `sysedgeV3.cf` 中包含所有 SystemEDGE 配置信息。通过 CA Virtual Assurance 用户界面中的配置或通过直接操作，修改 `sysedge.cf` 文件以配置代理。

配置文件 `sysedge.cf` 和 `sysedgeV3.cf` 中包含所有 SystemEDGE 代理配置信息。

`sysedge.cf` 配置文件是以 UTF-8 编码（纯文本）且可通过文本编辑器查看的文本文件，其中包含本地系统值，如：

- 系统说明、团体字符串和陷阱团体
- 用于报告敏感信息和运行操作脚本的代理行为
- 代理的监控表的条目
- AIM 加载和配置参数

配置文件 `sysedgeV3.cf` 是可以使用 `se_enc` 实用工具加密的未加密文件。该文件包含 SNMPv3 信息，如：

- SNMPv3 用户配置
- SNMPv2c 和 SNMPv3 陷阱目标

注意：

- 当您手动修改 `sysedge.cf` 或 `sysedgeV3.cf` 文件时，要使更改生效，必须重新启动 SystemEDGE 代理。
- `sysedge.mon` 文件是在运行时期创建的，并通过 SystemEDGE 代理自动进行更新。

在 UNIX 和 Windows 平台上，`sysedge.cf` 和 `sysedgeV3.cf` 文件安装在 `CASYEDGE\config` 目录中。`CASYEDGE` 是 UNIX 上安装 SystemEDGE 的目录的占位符。在 Windows 上，位置存储在注册表 `HKLM\Software\ComputerAssociates\SystemEDGE\InstallDir` 中。在安装期间，系统会提示用户提供以下信息：SNMPv1 团体、`sysContact`、代理 SNMP 端口及数据目录。用户提供的信息由安装程序在 `sysedge.cf` 文件中进行更新。

当代理首次启动时，SystemEDGE 会将 *.cf 文件复制到它的数据目录。使用数据目录中的文件配置代理。在 Windows 系统上，您也可以通过 SystemEDGE 控制面板访问这些文件。要访问这些文件，请选择“开始”、“控制面板”、SystemEDGE 并单击 `sysedge.cf` 或 `sysedgeV3.cf`。

在安装期间，安装程序会设置默认代理实例（SystemEDGE 服务）的端口。当您从命令行启动代理时，不要为默认实例指定 `-p` 参数。稍后将自动使用默认端口。只有默认代理实例与 CA Virtual Assurance 管理器进行通信。因此，从 CA Virtual Assurance 管理器接收到的任何配置文件 (`sysedge.cf`、`sysedgeV3.cf`) 将始终被放到 `SystemEDGE\data\portdefault-port` 目录中。例如： `C:\Program Files\CA\SystemEDGE\data\port161`

在升级期间，代理还会合并以前版本中的设置。有关升级配置文件设置的更多信息，请参阅[升级注意事项](#) (p. 110)。

sysedge.cf 和 sysedge.mon 之间的交互

SystemEDGE 代理将文件 `sysedge.mon` 用作备份，用来存储未在 `sysedge.cf` 文件中自动输入的通过 SNMP Set 进行的更改。该文件还存储运行时信息，如监视器的当前状态。`sysedge.cf` 中的条目优先于 `sysedge.mon` 中的条目。

当代理执行完全（或冷）重新启动时，它会首先读取数据中的 `sysedge.mon` 文件，然后读取 `sysedge.cf` 文件并覆盖现有监视器（当前状态信息和 `notInService` 行状态除外），然后添加其他监视器。当使用 CA Virtual Assurance 管理器重新配置代理时，它仅读取已交付的 `sysedge.cf` 文件中的条目。

如果手动删除 `sysedge.cf` 中的监控条目，且 `sysedge.mon` 中也存在该条目，则也必须从 `sysedge.mon` 中将其删除，以防止代理使用它。

使用 CA Virtual Assurance 的配置

您可以使用 CA Virtual Assurance 基于文件的配置功能来远程配置 SystemEDGE 代理。通过 CA Virtual Assurance 用户界面，您可以配置 `sysedge.cf` 文件中的任何参数（包括系统设置、安全设置、监视器及 AIM），并从一个中央位置将配置部署到一个或多个代理实例。

当您通过 CA Virtual Assurance 部署配置文件时，代理可在运行期间应用任何更改，且只需执行热启动。SystemEDGE 配置建议使用通过 CA Virtual Assurance 的远程配置。

有关从 CA Virtual Assurance 用户界面配置代理的详细信息，请参阅《CA Virtual Assurance 管理指南》和联机帮助。有关可以通过 CA Virtual Assurance 实施的代理安全锁定选项的详细信息，请参阅“概念”一章。

File-Based 配置

您可以通过以下方式配置代理：

- 通过 CA Virtual Assurance 策略，使用基于文件的配置远程配置
- 通过 CA eHealth AdvantEDGE View、CA Spectrum、CA NSM 代理视图或 SystemEDGE 命令行实用工具，使用 SNMP Set 远程配置
- 通过直接更新 `sysedge.cf` 文件本地配置

当您通过 CA Virtual Assurance 远程配置代理时，CA Virtual Assurance 管理器会将配置的 `sysedge.cf` 文件发送到指定的代理实例的 `CASYEDGE_DATADIR` 目录。该目录位置包含并完成从 CA Virtual Assurance 部署的任何运行时配置。当代理检测到新文件上传到该目录时，它会完成热启动并应用更改。当 `sysedge.cf` 文件由 CA Virtual Assurance 管理器交付给代理时，覆盖 SNMP Set 所做的任何修改。

当您使用 SNMP Set 配置代理时，将应用更改并将其保存到 `sysedge.mon` 文件中。

注意：您不能使用 SNMP Set 配置某些系统设置，如 SNMP 和安全信息。

当您通过直接更新本地的 `sysedge.cf` 文件来配置代理时，要使更改生效，必须重新启动代理。

以下主题介绍 `sysedge.cf` 文件中的参数及可用于每个参数的配置选项。有关使用 CA Virtual Assurance 的配置过程，请参阅《CA Virtual Assurance 管理指南》和联机帮助。

`sysedge.cf` 文件中的默认条目以模板指令作为前缀。当您通过 CA Virtual Assurance 配置代理时，模板指令可控制实际配置条目的创建。当您通过 CA Virtual Assurance 配置代理时，实际配置条目会在部署文件时自动创建。

当您手动编辑 `sysedge.cf` 文件时，必须手动输入没有模板指示的配置条目来实施参数设置。条目被描述并分到文件中的相关区域里。在模板指令下的适当区域手动输入必要条目。

手动编辑文件时，使用数据目录中的 `sysedge.cf` 文件。

配置系统信息

您可以在安装期间定义系统说明、位置和联系信息。该信息用于填充 MIB-II 中的系统组。您可以在安装之后在 `sysedge.cf` 文件中修改这些值。

要配置系统信息，请在 `sysedge.cf` 文件中输入以下字段或更新字段（如果您在安装期间已输入设置）：

```
sysdescr <description>
syscontact <contact>
syslocation <location>
```

配置 SNMP 信息

`sysedge.cf` 文件包含指导 SystemEDGE 代理如何通过 SNMP 进行通信的 SNMP 访问和配置信息。

您可以在 `sysedge.cf` 文件中配置以下 SNMP 信息：

注意：其中一些信息的配置基于您安装代理时输入的信息来进行。如果需要更改，您可以编辑文件中的该信息。

- 适用于 SNMP 通信的 IP 方法
- SNMP 绑定地址
- 访问团体
- SNMPv1 陷阱社区
- SNMPv1 陷阱源
- SNMPv1 陷阱目标

配置适用于 SNMP 用户数据报协议通信的 IP 族

主机可以有多个 IP 族源套接字和族目标地址，并且您可以设置适用于代理 SNMP 用户数据报协议 (UDP) 通信的首选方式。

要设置 UDP 通信的首选 IP 族方式，请将下列行添加到 `sysedge.cf` 文件中：

```
sysedge_ip_family <method>
```

方式

指定首选通信方式。如果指定的方式不可用，代理首先尝试使用 IPv4，然后尝试使用 IPv6。可用选项如下：

1

指定代理仅尝试使用 IPv4 作为首选 SNMP UDP 通信方式。

2

指定代理仅尝试使用 IPv6 作为首选 SNMP UDP 通信方式。

3

指定代理尝试使用 IPv4 和 IPv6 两者作为首选 SNMP UDP 通信方式。此为默认方式。

示例

输入下列行，以便仅将 IPv4 模式（如果可用）用于 SNMP UDP 通信：

```
sysedge_ip_family 1
```

配置 SNMP 绑定地址

默认情况下，SystemEDGE 绑定到所有接口 (* /UDP-161)。您可以使用 sysedge.cf 文件中的 bind_address 参数将代理绑定到指定接口，如下所示：

```
bind_address <address>
```

address

指定代理绑定到的特定接口。该值可以是 IPv4 或 IPv6 地址。

示例

在仅绑定到 10.1.0.202 地址的 sysedge.cf 文件中输入下列行：

```
bind_address 10.1.0.202
```

在仅绑定到 ea2f:fe90:abcd:0000:230:a2f:200:ad01 地址的 sysedge.cf 文件中输入下列行：

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

设置绑定地址

在您为 SystemEDGE 设置绑定地址时，请考虑以下情况：

- SystemEDGE 安装程序在安装时不支持设置绑定地址。
- 策略配置需要在策略级别（而不是个人计算机级别）定义绑定地址。

策略配置需要在策略内设置绑定地址。如果在代理受管/提高其配置之前设置绑定地址，绑定地址将被从最初配置中拉出并自动包含到已交付的配置中。有几种方法能够实现此过程。

下列术语用于所有这些用例：

静态 sysedge.cf 文件

标识安装程序规定的文件，位于 *Installed_Dir*\SystemEDGE\config 目录中。

默认值：

Windows: C:\Program Files\CA\SystemEDGE\config

UNIX/Linux: /opt/CA/SystemEDGE/config

动态 sysedge.cf 文件

标识正在进行的 SystemEDGE 配置文件，位于 *Data_Dir\port<number>* 目录中。

默认值:

Windows: C:\Users\Public\CA\SystemEDGE\port161

UNIX/Linux: /opt/CA/SystemEDGE/config/port161

选项 1: 在安装并启动代理之前设置绑定地址

如果尚未安装代理，可以按如下方式设置绑定地址：

1. 以未受管模式安装代理（不指定管理器名称）。
2. 清除“安装之后启动”选项，以便不启动代理。
3. 安装后，编辑静态 sysedge.cf 文件，并设置 bind_address 和 manager_name。
4. 通过常规机制启动代理。

选项 2: 在运行的系统受管前在其上设置绑定地址

如果已按未受管模式安装了 SystemEDGE，可以按如下方式设置绑定地址：

1. 使用正常机制停止 SystemEDGE。
2. 编辑动态 sysedge.cf 文件并设置 bind_address1。
3. 编辑静态的 sysedge.cf 文件并根据需要设置 manager_name。
4. 使用常规机制启动 SystemEDGE。

另一个备选方案是，删除动态 sysedge.cf 文件，并在静态 sysedge.cf 文件中设置 bind_address。

选项 3: 在现有受管系统上设置绑定地址

如果 SystemEDGE 已经受策略配置管理，请按如下方式设置绑定地址：

1. 使用正常机制停止 SystemEDGE。
2. 删除动态 sysedge.cf 文件。
3. 删除与动态 sysedge.cf 文件位于相同目录中的 .sysedge.id 文件（这将导致代理再次提高其配置）。
4. 编辑静态 sysedge.cf 文件，并根据需要设置 bind_address。
5. 使用常规机制启动 SystemEDGE。

配置访问团体

可以在安装期间定义只读或读写 **SNMPv1** 访问团体，并在 **sysedge.cf** 文件中修改或定义其他团体。

注意： 在 **sysedgeV3.cf** 文件中定义 **SNMPv2c** 和 **SNMPv3** 访问信息。

要配置 **SNMPv1** 访问团体，请使用以下格式在 **sysedge.cf** 文件中输入它们或编辑安装期间提供的任何条目：

```
community <name> <access> <ip-address-list>
```

name

指定授予已定义地址定义访问权限的团体。可将任意 **ASCII** 字符用于团体名称。

access

指定授予的权限级别，是只读还是读写。

ip-address-list

指定使用给定团体名进行访问的以空格分隔的 **IP** 地址列表。由于系统仍可以欺诈 **IP** 地址，因此访问列表不完全安全。但是，访问列表能够限制合法使用。您可以提供 **IPv4** 或 **IPv6** 地址作为访问列表。

安装之后，**sysedge.cf** 在默认情况下定义一个名为“公共”的只读团体，从而提供对 **MIB** 对象的只读访问权限。

注意： 常用做法将提供使用团体名称“公共”的只读访问权限。

要修改 **MIB** 对象的值，必须定义具有读写访问权限的团体。在以下示例中，**SystemEDGE** 允许对具有以下 **IP** 地址之一的系统拥有使用团体名称“专用”的读写访问权限：45.0.4.10、45.0.8.12、198.130.5.7 或 ea2f:fe90:abcd:0000:230:a2f:200:ad01。**SystemEDGE** 将任何其他尝试使用“专用”的系统视为身份验证失败：

```
团体专用读写 45.0.4.10 45.0.8.12 198.130.5.7 ea2f:fe90:abcd:0000:230:a2f:200:ad01
```

此处“专用”团体名称仅供参考之用。为安全起见，请使用更独特的团体名称。

注意： 团体名称出现在 **SNMP** 消息头时不加密，并且访问列表受潜在 **IP** 欺诈的影响。如果您正在使用 **SNMPv1** 团体，则应该采取配置步骤以限制潜在的安全违规。有关详细信息，请参阅《有关配置安全的建议》。

针对 SNMPv1 团体的访问列表

如果访问列表对于定义的团体为空，SystemEDGE 将授予对任何使用相关团体名的系统的访问权限。下列限制将应用于访问列表：

- 在 `sysedge.cf` 中指定的访问列表仅用于 SNMPv1 通信。
- 在 SNMPv3 配置文件 `sysedgeV3.cf` 中分别配置 SNMPv2c 和 SNMPv3 访问列表。有关如何定义 SNMPv3 用户的访问列表的详细信息，请参阅 [SNMPv3 配置](#) (p. 150)。
- 您必须通过空格字符分隔 IP 地址，而不能使用任何其他字符（包括换行）。
- 团体字符串语句的最大长度（包括任何访问列表）为 1024 个字符，将为 60 个 IPv4 地址提供足够的空间，为 IPv6 地址提供较少的空间。要配置较长的访问列表，请定义单独的团体。
- SystemEDGE 代理包括充当管理器系统并向代理发送请求的 `edgewatch`、`edgemon` 和 `emphistory` 命令行实用工具。如果您在安装代理的同一个系统上使用任何这些实用工具，则将该系统的 IP 地址添加到访问列表。

指定 SNMPv1 陷阱源

您可以指定 SNMPv1 陷阱源（可选），它是作为 SystemEDGE 协议说明单元 (PDU) 中起源的 IP 地址。默认情况下，SystemEDGE 使用 `gethostbyname` 函数调用返回的值。指定 `trap_source` 参数以覆盖默认行为。该参数适用于 SNMPv1 陷阱和 SNMPv2c/SNMPv3 陷阱。

代理只有一个陷阱源值，因此您只需在 `sysedge.cf` 中设置一次该值。所有团体中的所有陷阱都采用指定地址。您指定的地址必须是有效的 IPv4 或 IPv6 地址或主机名，但是 SystemEDGE 代理不执行确定您是否为用作陷阱源的特定系统指定有效地址的错误检查。

要指定陷阱源，请将下列行添加到 `sysedge.cf` 文件：

```
trap_source <address>
```

address

指定发送陷阱时使用的源地址。可以指定 IPv4 或 IPv6 地址或主机名。

默认： 主机名

例如，要将陷阱源设置为 IP 地址为 10.0.7.73 及主机名为 `system1.ca.com` 的系统，您可以输入下列内容之一：

```
trap_source 10.0.7.73
```

```
trap_source system1.ca.com
```

配置 SNMPv1 陷阱目标

`sysedge.cf` 文件包含 SNMPv1 陷阱目标的定义，这将通知 SystemEDGE 代理发送 SNMPv1 陷阱消息的位置。可以配置代理以将陷阱发送到任意数量的管理系统。您可以在安装期间在 `sysedge.cf` 文件中定义 SNMPv1 陷阱目标。

要配置 SNMPv1 陷阱目标，对要发送 SNMPv1 陷阱的每个系统的 `sysedge.cf` 文件添加下列行：

```
trap_community <name> <address> [<port> [<encoding> [<source>]]]
```

name

指定随陷阱发送的团体名称。

address

指定向其发送陷阱的 IPv4 或 IPv6 地址。

port

（可选）指定将陷阱发送到的 UDP 端口。

默认值： 162

编码

（可选）指定如何将 `trap_source` 参数中定义的源地址添加到陷阱。如果 `trap_source` 转换为 IPv6 地址，该参数将非常重要。以三位数的格式 XYZ 输入编码参数，假定起始为零。

默认值：000

X

控制扩展四字节 IPv4 源地址字段（仅 SNMPv1 陷阱）。输入 0 将不扩展源地址字段以包括 16 字节 IPv6 地址，输入 1 将扩展源地址字段。

Y、Z

控制将源信息列入陷阱中的 `varbind` (Y) 或 UDP 数据包（仅限 Z; SNMPv1 陷阱）。为这些数字输入以下内容之一：

0：不修改陷阱的 `varbind` 或外部 UDP 数据包。

1：如 `varbind` 或数据包中一样包含 `trap_source` 参数（IPv4/IPv6 地址或主机名）。

2：最好将 `trap_source` 参数作为 IPv4 地址包含在内（然后依次是 IPv6 地址、主机名）。

3：最好将 `trap_source` 参数作为 IPv6 地址包含在内（然后依次是 IPv4 地址、主机名）。

4：最好将 `trap_source` 参数作为主机名包含在内（然后依次是 IPv4、IPv6）。

5：遵循 2 的首选项并包含主机名。

6：遵循 3 的首选项并包含主机名。

7：遵循 1 的首选项并包含主机名（如果 `trap_source` 是 IPv6 地址）。

源

指定用作陷阱源的 IPv4 或 IPv6 地址或主机名。

默认：`trap_source` 参数值

例如，添加下列行以便通过附加到 UDP 数据包的 `trap_source` 参数和主机名将团体名称为“我的团体”的陷阱发送到端口 1692 上的 IP 地址 10.16.5.26：

```
trap_community mycommunity 10.16.5.26 1692 7
```

注意：`sysedge.cf` 文件仅定义 SNMPv1 陷阱目标。有关配置 SNMPv2c 和 SNMPv3 陷阱的信息，请参阅 SNMPv3 配置。

SystemEDGE 的陷阱代理地址

从 SystemEDGE 代理发送的陷阱的源地址是代理绑定的地址。默认情况下，SystemEDGE 代理绑定到所有网络接口，因此从代理发送的陷阱使用其首个成功的 IP 地址。

如果将 SystemEDGE 配置为将陷阱发送到作为代理运行于同一本地服务器上的陷阱接收者（如 edgetrapmon），源地址将很可能是回送地址（对于 IPv4 来说为 127.0.0.1 或对于 IPv6 来说为 ::1）。

配置身份验证失败陷阱

在 SystemEDGE 代理接收团体名称与其识别的团体之一不匹配的 SNMP 消息时，您可以将该代理配置为发送身份验证失败陷阱。

默认情况下，代理不发送身份验证失败陷阱。要将代理配置为发送身份验证失败陷阱，请通过将井号 (#) 字符添加到行的开头来注释掉 sysedge.cf 中的 no_authen_traps 指令：

```
# no_authen_traps
```

注意：该选项为针对 SNMPv1 陷阱（在 sysedge.cf 中配置）以及 SNMPv2c/SNMPv3 陷阱（在 sysedgeV3.cf 中配置）设置的陷阱目标申请身份验证失败陷阱。

配置 SNMP Set 通知

在使用 CA Virtual Assurance 管理 SystemEDGE 代理时，当对代理配置执行了任意 SNMP Set 修改时，CA Virtual Assurance 管理器默认情况下接收通知。这些通知允许 CA Virtual Assurance 管理员仍可以知道对代理配置所做的所有更改，并能够回滚任何不需要的更改。

注意：该功能仅适用于将 SystemEDGE 代理用于 CA Virtual Assurance。有关配置这些通知的详细信息，请参阅《CA Virtual Assurance 管理指南》。

默认情况下，该功能在通过 CA Virtual Assurance 管理代理时启用。要对 CA Virtual Assurance 管理器禁用 SNMP Set 通知，请将下列行添加到 sysedge.cf 文件中：

```
no_snmp_set_notifications
```

配置 SNMP Set 限制

默认情况下，SystemEDGE 代理不允许在 Systems Management Empire MIB 和主机资源 MIB 中的下列表中存在 SNMP Set 请求：

远程 Shell 组

允许管理系统远程指导代理在代理系统（启用时）上运行 Shell 脚本和程序。泄漏此类信息可能会产生潜在的安全风险。

注意：有关该组的详细信息，请参阅“Systems Management Empire MIB”一章。

进程表和运行软件表

允许访问在代理系统上运行的进程和其他软件。允许这些表中存在 SNMP SET 请求时，请考虑安全风险。

注意：有关进程表的详细信息，请参阅“Systems Management Empire MIB”一章。有关运行软件表的详细信息，请参阅附录“主机资源 MIB”。

您可以对这些表启用 SNMP Set，假定查询将具有读写权限的有效团体用于 SNMPv1 和 SNMPv2c 通信或将有效的 SNMPv3 用户凭据用于 SNMPv3 通信。

要在远程 Shell 组中启用对 SNMP Set 的支持，通过添加井号字符 (#) 来注释掉 sysedge.cf 文件中的下列行，如下所示：

```
# no_remoteshell_group
```

要在进程表和运行软件表上启用对 SNMP Set 的支持，通过添加井号字符 (#) 来注释掉 sysedge.cf 文件中的下列行，如下所示：

```
# no_process_sets
```

您也可以完全删除这些行。

配置 GetBulk 响应消息大小

可以配置 SystemEDGE 代理以限制 GetBulk 请求的响应大小。如果限制 GetBulk 响应消息负载的字节大小，还将限制 SystemEDGE 代理使用的内存。

如果响应消息超出限制，SystemEDGE 代理将迭代以向通过有效响应消息形式请求的 OID 列表提供服务。例如，要避免 IP 数据包的碎片化，您可以将软限制设置为 60000。由于 IP 数据包的最大大小为 65535，因此它仍在溢出限制以下。

要将代理配置为限制 GetBulk 响应，请在 `sysedge.cf` 中添加 `bulk_read_response_soft_limit` 选项，如下所示：

```
bulk_read_response_soft_limit <limit>
```

此处的限制指定响应消息必须局限于的字节数。它是一个无符号整数，默认值为 60000。

注意：该选项为针对 SNMPv2c/SNMPv3 陷阱（在 `sysedge.cf` 和 `sysedgeV3.cf` 中配置）设置的陷阱目标申请身份验证失败陷阱。

配置联邦信息处理标准模式

可以配置 SystemEDGE 代理如何使用 `sysedge_fips_mode` 参数处理加密。可提供以下三种加密方式：

- 确保最低安全功能的内部解决方案（仅限 MD5/SHA 和 DES）
- CA eTrust 公钥基础架构库 (libetpki/libcapki)
- RSA BSAFE Crypto-C Micro Edition 库 (libcryptocme2)（符合 FIPS 140-2 标准）

要配置加密模式，将包含 `sysedge_fips_mode` 参数的行添加到 `sysedge.cf` 文件中，如下所示：

```
sysedge_fips_mode <method>
```

方式

指定要使用的加密类型。可用选项如下：

{0}

启用 CA eTrust 公钥基础架构库，如果该方式失败，返回到内部最低安全解决方案。

1

启用符合 FIPS 标准的加密，如果该方式失败，返回到方式 0。如果不配置该参数，则默认为该方式。

2

指定代理在仅 FIPS 模式中运行。如果其失败，将启用 RSA BSAFE Crypto-C Micro Edition FIPS 兼容库且不执行加密。

例如，输入下列行以运行仅使用 FIPS 认证协议和 FIPS 认证库的代理：

```
sysedge_fips_mode 2
```

有关如何启用 FIPS 加密的详细信息，请参阅附录“FIPS 140-2 加密”。

配置诊断日志记录机制

SystemEDGE 代理包含一个常见的日志记录机制，用于将诊断消息记录到一个主要由 CA 支持部门使用的文件中。您可以配置日志记录机制的以下方面：

- 日志级别
- 日志文件名称、大小和大小限制处理机制
- 为哪个模块记录消息

要配置代理日志记录，请在 `sysedge.cf` 文件中输入下列行：

```
sysedge_loglevel <level>
sysedge_logfile '<file>' [<size> [<number>]]
sysedge_logpattern <modules>
```

级别

指定要记录的消息级别。您可以指定以下任何内容：

- 致命
- 严重
- 警告
- 信息
- 调试
- 调试[1-3]

代理日志消息在您指定的级别开始，并涵盖更高的重要级别。例如，如果您指定警告，代理将记录所有警告、严重和致命消息。

默认值： 信息

'file'

指定日志文件名。您可以在文件名中包含绝对路径名。如果不包含绝对路径名，将在代理的指定端口数据目录中创建日志文件。

默认值： 'sysedge.log'

大小

指定最大文件大小 (KB)。如果将该参数设置为零或小于零，日志文件将没有大小限制。

默认： 10240

number

在达到大小限制时，控制对日志文件的处理。您可以指定以下任何选项：

{0}

在达到大小限制时日志记录将停止。

1

在达到大小限制并转到日志文件的开头时，日志记录会将当前日志文件内容保存到 “<文件>” 中。

>1

日志记录结合使用达指定备份文件数目的环绕集（或环）和包含任意时候最早日志条目的 “<file>.<number>”。

-1

达到大小限制时，日志记录转到日志文件的开头。

-2

在达到大小限制并转到日志文件的开头时，日志记录会将当前日志文件内容保存到 “<file>.<YYYY-MM-DD_hh.mm.ss>” 中。

默认：1

模块

指定用于记录消息的以空格分隔的模块列表。可用模块如下所示：

- **se**: SystemEDGE 代理
- **svcrsp**: 服务响应监视器 AIM
- **rmonwbem**: 远程监控 AIM
- **monwin**: 监视器维护窗口 AIM
- **perfcube**: 性能多维数据集 AIM
- **util**: 通用库/共享库
- **regexpr**: 通用库/共享库
- **shm**: 通用库/共享库
- **默认**: 来自任何模块的日志消息

以下示例将 SystemEDGE 代理和 SRM AIM 重要级别为信息及以上的所有消息记录到名为 `sysedgemsg.log` 的文件中，该文件的大小限制为 30000 KB 并具有保存所有旧文件的打包大小限制机制。

```
sysedge_loglevel info
sysedge_logfile sysedgemsg.log 30000 -2
sysedge_logpattern se svcrsp
```

配置维护模式

SystemEDGE 代理可以在维护模式下运行，在该模式下，它将停止处理所有监视器条目和发送陷阱。如果代理的系统正处于计划停机状态，并且您想避免接收假警报陷阱，则维护模式很有用。

在维护模式下，代理将继续收集度量标准并响应 **SNMP** 请求，但会挂起对所有监视器和历史记录收集的处理。在外部，代理将所有监视器的行状态报告为 **notInService**，而内部保持正确的行状态。代理会保存维护开始时所有监视器的当前值，将其与维护结束时的当前值进行比较，并在必要时发送陷阱以响应当前值。

要启用维护模式，请将下列行添加到 **sysedge.cf** 文件中：

```
sysedge_maintenance
```

在配置文件中管理维护模式没有在 **CA Virtual Assurance** 管理器接口中使用维护模式控制效率高。您可以从接口切换维护模式，在设置更改时会自动更新和部署配置文件。在通过 **CA Virtual Assurance** 配置时，代理执行热启动，而非全面重新启动。有关在 **CA Virtual Assurance** 中启用维护模式的详细信息，请参阅《*CA Virtual Assurance 管理指南*》和 *联机帮助*。

为操作配置支持

默认情况下，SystemEDGE 代理不允许对监控表执行操作命令。由于命令和脚本可以根用户或管理员用户身份运行命令，因此运行操作命令和脚本的功能可能存在潜在的安全问题。如果要在达到某些监控阈值时执行操作，必须启用对操作的支持。

要为操作执行启用支持，请通过添加井号 (**#**) 字符注释掉 **sysedge.cf** 文件中的下列行，如下所示：

```
# no_actions
```

为子程序配置用户权限和组权限（仅 UNIX）

默认情况下，SystemEDGE 代理通过其有效用户权限和组权限（通常为根用户）运行子程序（例如，远程 Shell、操作和扩展对象调用）。根据实行的本地安全策略，您可能希望将代理设置为使用通过不同用户权限和组权限运行的操作和扩展对象。

要通过除根用户之外的用户的有效用户权限和组权限运行子程序，请将下列行添加到 `sysedge.cf` 文件中：

```
subprogram_user_name <name>
subprogram_group_name <name>
```

name

指定用于子程序的用户名或组名。如果提供的名称无效，禁用所有子程序的执行。

配置 MIB 表限制

`sysedge.cf` 文件中存在多个参数，用于防止 SystemEDGE 代理填充 Systems Management Empire MIB 中的某些表。默认情况下不输入任何参数，但是您可以添加它们以限制对包含敏感信息或消耗大量资源的表进行访问。

要禁用对某些 MIB 表的支持，请根据要限制的表在 `sysedge.cf` 文件中输入下列任意行：

no_process_table

限制填充进程表。该表可以发现基础系统上运行的进程，该操作将违反实行的本地安全策略。

no_topprocs_table

限制在顶级进程表中填充对象。

no_who_table

限制填充用户表，提供当前登录系统的用户信息。泄漏此类信息可能会产生潜在的安全风险。

no_usergroup_table

限制填充用户表和组表，提供对系统配置的用户帐户和用户组的相关信息。如果组织将用户和组信息的泄漏视为安全问题，或如果您具有一个包含很多用户和组的分布式系统，您可能希望限制对这些表进行访问。代理定期在内部缓存这些信息，因此存储用户和组信息将消耗大量资源。

no_trapcommunity_table

限制填充陷阱团体表，提供代理的当前 **SNMP** 配置信息（包括版本、团体和陷阱目标）。由于 **SNMP** 设置（如团体）的敏感性质，您可能希望限制对该表进行访问。

no_mirror_monitor_table

限制填充镜像监视器表，提供自主监视器表的只读镜像，从而允许在特定上下文中显示监控条目而无需搜索整个自主监视器表。如果自主监视器表的镜像会消耗太多资源，您可能希望禁用该表。

no_mirror_aggregate_table

限制填充镜像合计表，提供了合计表的只读镜像，从而允许在特定上下文中显示合计条目而无需搜索整个合计表。如果合计表的镜像会消耗太多资源，您可能希望禁用该表。

有关任何这些表的详细信息，请参阅“**Systems Management Empire MIB**”一章。

配置设备状态检查限制

Sysedge.cf 文件中存在多个参数，用于防止 **SystemEDGE** 代理查询某些系统设备的状态，通常为避免潜在的代理阻止。默认情况下，**SystemEDGE** 代理不会自动确定系统上所有软盘设备的状态。

要启用软盘驱动器状态检查，请通过添加井号 (**#**) 字符来注释掉 **sysedge.cf** 文件中的下列行：

```
# no_stat_floppy
```

您也可以完全删除这些行。

您可以禁用默认情况下会启用的其他设备状态检查。要禁用设备状态检查，请根据要限制的设备在 **sysedge.cf** 文件中输入任何下列行：

注意： 由于将不会向各自的对象提供所有信息，因此启用任何这些限制都将影响相应的 **MIB** 表。

no_serial_status

禁用主机系统上所有串行设备的状态检查。代理默认执行该操作，作为对主机资源 **MIB** 的 **hrDeviceTable** 支持的一部分。由于无法通过除自身之外的任何进程处理串行设备的打开和关闭（对于确定状态很有必要），因此一些串行应用程序会在执行该操作时遇到问题。输入此行时，代理将仅检查 **hrDeviceTable** 中的键盘和鼠标，并为串行端口的状态返回 **unknown(1)** 以响应管理系统查询。

no_discover_disks

在 SystemEDGE 上启用磁盘设备的搜索和发现选项。如果已禁用此选项，磁盘设备不会在以下 SystemEDGE SNMP 表中显示：

- hrStorageTable
- hrDeviceTable
- hrDiskStorageTable
- hrFSTable
- devTable
- diskStatsTable

注意：启用 no_discover_disks 选项时，将自动启用 no_probe_disks 选项。

no_probe_disks

禁止检查安装在基础系统上的磁盘和 CD-ROM 的大小、容量、说明和其他属性，以避免驱动器等待状态信息时较旧的 UNIX 系统上的潜在冗长代理阻止。

注意：如果添加此行，代理将发现磁盘设备，但可能无法提供磁盘统计信息、容量信息、设备说明和状态信息。

no_stat_nfs_filesystems

（仅 UNIX）禁止泄漏本地和远程文件系统数据，默认情况下代理会通过 Systems Management Empire MIB 执行此操作。在一些 UNIX 系统上，如果文件系统从不再可用的远程文件服务器安装，检查文件系统数据可能导致代理阻止。在这种情况下，无法取消代理阻止。

注意：如果添加该行，将禁用对所有远程文件系统的检查，而非仅 NFS 文件系统。

no_hpux_probe_graphics

禁止检查 HP-UX 系统上图的状态以避免潜在的代理阻止。

配置热启动发现

使用 CA Virtual Assurance 管理 SystemEDGE 代理时，代理将在您通过 CA Virtual Assurance 管理器更新其配置时执行热启动。热启动可使代理在无需全面重新启动的情况下实施更改。默认情况下，代理执行热启动时不会重新发现所有设备，因为如果您管理的系统具有很多设备，这将花费很长时间。但是，您可以启用该发现以在热启动之后重新发现所有设备。

要启用热启动发现，请通过添加井号 (#) 字符来注释掉 `sysedge.cf` 文件中的下列行：

```
# no_warmstart_discovery
```

热启动功能仅适用于通过 CA Virtual Assurance 配置而处于受管模式的代理。有关详细信息，请参阅《*CA Virtual Assurance 管理指南*》。

配置 Linux 可用内存计算

您可以配置 SystemEDGE 计算 Linux 上的可用内存的方式，以在计算中包括系统缓冲区、已缓存的内存或两者。

要配置 Linux 可用内存计算，请在 `sysedge.cf` 文件中输入下列行：

```
linux_freemem_include [buffers] [cached]
```

buffers

在可用内存计算中包括系统缓冲区。

cached

在可用内存计算中包括磁盘已缓存内存。

例如，要在可用内存计算中包括系统缓冲区和磁盘已缓存内存，请添加下列行：

```
linux_freemem_include buffers cached
```


在多层次结构中配置监视器合计

SystemEDGE 可以合计自主监视器和进程监控条目来为相同属性关联多个监视器，以仅显示最严重的重要级别。合计的监视器具有相同的对象类、实例和属性。默认情况下，在管理合计监控条目时，代理仅对具有最差合计状态的条目发送陷阱和执行操作。您可以在 `sysedge.cf` 文件中更改该默认设置。

要为所有连接的监视器（而非只针对最差的重要级别）发送陷阱和执行操作，请将下列行添加到 `sysedge.cf` 中：

```
aggregate_monitor_traps
aggregate_monitor_actions
```

您也可以配置监视器合计的级别。默认情况下，代理将具有相同对象类、实例和属性的所有自主监视器和进程监视器合计到**标准合计**中。属于一个合计的所有监视器称为**连接的监视器**。您可以在 `sysedge.cf` 文件中配置高级别的合计。高级别的合计称为**超级合计**（使用星号“*”作为占位符）。例如，您可以在对象合计表中创建监视器合计，如下所示：

- 具有相同对象类、属性和任何实例的监视器
- 具有相同类、任何实例和属性的监视器

虚拟对象实例的合计支持两层以上。支持的实例已结构化，如下所示：

```
//<any1>/<any2>/.../<anyN>/<instance>
```

示例

LPAR 上的对象实例：

```
//hmc1/hyper1/lpar1/disk1
```

仅为两层（最右边两层）提供超级合计，而非为所有层提供。对于该示例，创建了以下超级合计：

```
//hmc1/hyper1/lpar1/*
```

```
//hmc1/hyper1/*/disk1
```

```
//hmc1/hyper1/*/*
```

创建特殊超级合计 `//*` 以合计所有实例

注意：对于对象实例，使用“//”和“/”作为分隔符来将实例与其托管的层次结构关联：“//hierarchy/instance”。层次结构可以为多层。对于本地系统实例，前置“//./”。

示例

```
//VM-Host/VM-Guest/Disk1
//Host/Disk1
//./Disk1
```

可以为最右边的一层或两层启用超级合计。创建特殊超级合计 `//*` 以合计所有实例。

关于级别 0002 的示例

```
//VM-Host/VM-Guest/*
//Host/*
//./*
```

关于级别 0004 的示例

```
//VM-Host/*/Disk1
//*/Disk1      (不包括 //./Disk1)
```

关于级别 0008 的示例

```
//VM-Host/*/*
//*//*      (不包括 //./Disk1)
//*
```

要配置监控条目合计级别，请将下列行添加到 `sysedge.cf` 文件中：

```
aggregate_level <level>
```

级别

指定允许以下表合计级别的位字段：

类	实例	属性	级别
类	<code>//hierarchy/instance</code>	属性	(标准合计)
类	<code>//hierarchy/*</code>	属性	0002
类	<code>//hierarchy-1/*/instance</code>	属性	0004
类	<code>//hierarchy-1/*/*</code>	属性	0008
类	<code>//hierarchy/instance</code>	*	0010
类	<code>//hierarchy/*</code>	*	0020
类	<code>//hierarchy-1/*/instance</code>	*	0040
类	<code>//hierarchy-1/*/*</code>	*	0080
*	<code>//hierarchy/instance</code>	属性	0100
*	<code>//hierarchy/*</code>	属性	0200
*	<code>//hierarchy-1/*/instance</code>	属性	0400

类	实例	属性	级别
*	//hierarchy-1/*/*	属性	0800
*	//hierarchy/instance	*	1000
*	//hierarchy/*	*	2000
*	//hierarchy-1/*/*instance	*	4000
*	//hierarchy-1/*/*	*	8000

例如，将下列行添加到 `sysedge.cf` 文件中以应用所有合计规则，从而导致合计达到代理总体状态：

```
aggregate_level 0xffff
```

有关监视器合计和对象类、实例和属性自主监视器表列的详细信息，请参阅“概念”和“自主监控”章节。

监控配置

`sysedge.cf` 文件存储控制 SystemEDGE 代理监控操作的监控条目。您可以使用以下任何方式创建监控条目：

- 在 CA Virtual Assurance 用户界面中动态创建，其中您可以对任何代理系统部署更改而无需重新启动
- 使用 SNMP Set 请求远程创建或在本地创建
- 直接在本地 `sysedge.cf` 文件中创建以便重新启动后实施

您可以将行添加到 `sysedge.cf` 文件中，以便将监控条目添加到下列 MIB 表中：

- 用于监控 MIB 对象阈值的自主监视器表，包括文件系统、接口、处理器等。自主监视器表支持基于 MIB 属性的 OID 或对象类、实例和属性值输入监控条目，以及设置属性阈值。
- 用于监控进程属性和 Windows 服务的进程监视器表。进程监视器表支持基于进程或服务正则表达式输入监控条目，以及设置进程属性阈值。
- 用于定义和监控进程组的进程组监视器表。进程组监视器表支持基于进程正则表达式输入监控条目，以组成一个组来监控更改或组统计信息。
- 用于监控日志文件的正则表达式模式匹配情况的日志监视器表。日志监视器表支持基于日志文件和文件中的正则表达式输入条目。

- 用于监控 Windows 事件日志的正则表达式模式匹配情况的 NT 事件监视器表。NT 事件监视器表支持基于 Windows 事件日志中要匹配的正则表达式模式输入监控条目。
- 用于收集历史数据以进行基准和趋势分析的历史记录控制表。历史记录控制表支持基于任何基于整数的 MIB 属性 OID 输入条目。

代理状态管理模型从自主监视器表和进程监视器表中的条目派生处状态和重要级别信息，并合计连接的监视器以仅显示最差的重要级别状态并对其采取行动。

有关每个监控表如何运行的详细信息，请参阅“概念”一章。有关表列以及在 `sysedge.cf` 文件中为每个表创建条目的详细信息，请参阅有关各监控类型的章节。有关从 CA Virtual Assurance 用户界面创建监控条目的详细信息，请参阅《CA Virtual Assurance 管理指南》和联机帮助。

注意：您也可以定义要在 `sysedge.cf` 文件中监控的 MIB 扩展名和 Windows 注册表或性能数据。有关详细信息，请参阅“自定义 MIB 对象”和“Windows 注册表和性能 MIB 对象”章节。

加载 SystemEDGE AIM

SystemEDGE 代理提供一个插件体系结构，您可以通过该结构在初始化时加载可选的 SystemEDGE AIM。这些 AIM 可提供支持应用程序特定 MIB 变量的灵活的可扩展方法。

您可以通过编辑 `sysedge.cf` 文件指定代理应加载的 AIM。您可以指定绝对路径或仅指定 AIM 名以供 SystemEDGE 代理找到要加载的 AIM。

在每个平台上分发 SystemEDGE 时会随附下列“按需”模块。和 AIM 不同，将在需要该功能时隐式加载这些模块：

- 监视器维护窗口
- 性能多维数据集

注意：先前顶级进程 AIM 中包含的功能现在将集成到基础代理中。有关详细信息，请参阅“进程和服务监控”一章。

要加载 SystemEDGE AIM，请在 `sysedge.cf` 文件的下列行中输入 AIM 名称或 AIM 的名称和完整路径：

```
sysedge_plugin
```

例如，要加载 SRM AIM，请在 `sysedge.cf` 文件中输入下列内容：

```
sysedge_plugin svcrsp
```

您无需输入绝对路径，尽管输入路径仍会创建一个有效条目。SystemEDGE 代理自动为 AIM 加载派生平台和文件位置。

有关启用其他 AIM 的信息，请参阅该 AIM 的文档。有关配置 AIM 的详细信息，请参阅“SystemEDGE AIM”一章。

监视器维护窗口模块配置

“监视器维护窗口”模块提供了 SystemEDGE 监控基于时间的控制。在未停止监视器自身时其不会产生重要意义的期间内，“监视器维护窗口”模块可以从自身、进程、进程组、日志文件、历史记录和 Windows 事件监视器停止生成陷阱。

该“按需”模块随 SystemEDGE 代理一起提供，您可以从 `sysedge.cf` 文件配置其操作。在您将行添加到文件以配置模块时，它将自动启用。

有关该模块的详细信息，请参阅“SystemEDGE AIM”一章。

Perfcube 模块配置

Perfcube 模块收集系统性能多维数据集中的历史记录表度量标准以供 CA NSM 系统性能进行性能分析。该模块读取历史记录表中的条目，基于定义的时间间隔执行度量标准平均，并将平均度量标准值写入多维数据集文件中。

该“按需”模块随 SystemEDGE 代理一起提供，您可以从 `sysedge.cf` 文件配置其操作。当您使用下列参数在文件中配置大于零的多维数据集收集时间间隔时，将自动启用 AIM：

```
emphistory cube_interval interval
```

interval

指定将历史记录信息收集到多维数据集文件的时间间隔（以分钟为单位）。

有关配置 Perfcube 模块的详细信息，请参阅“SystemEDGE AIM”一章。

有关配置安全的建议

出于安全目的我们建议您实施以下配置参数：

```
no_who_table
no_usergroup_table
no_remoteshell_group
no_process_sets
```

有助于实施安全的其他配置选项如下所示：

```
no_actions
no_process_table
no_trap_community_table
no_mirror_monitor_table
no_mirror_aggregate_table
subprogram_user_name
subprogram_group_name
```

监视器和自动监测器的正则表达式示例

本部分提供了针对每个监视器类型的上下文正则表达式示例。

进程监视器

- 匹配以“svchost”开头的进程：
`^svchost`
- 匹配包含“sql”的进程：
`sql`
- 匹配以“svc”结尾的进程：
`svc$`
- 匹配“cmd”和“CMD”进程（需要 PCRE）：
`/cmd/i`

注意：除非您正在监控进程的多个实例的存在状态（具有 0x4000 标志），否则请使用仅匹配单一进程的正则表达式。

Windows 服务监视器

- 匹配显示名称为“Print Spooler”的服务：
`^Print Spooler$`

注意：对于 Windows 服务，请使用仅匹配一种服务的正则表达式。

进程组监视器

- 匹配以“sql”或“svchost”开头的所有进程：

`^sql|^svchost`

Windows 事件监视器（源筛选）

- 匹配来自任何源的事件：

`.*`

- 匹配来自 HTTP 服务器的事件：

`http`

Windows 事件监视器（说明筛选）

- 匹配带有事件 ID 为 277（需要“开头附加事件 ID”标志 - 0x0100）的事件：

`\[277\]`

- 匹配在行中的任何位置包含文本“FATAL”的事件：

`FATAL`

日志文件监视器（搜索筛选）

- 匹配表示 su 尝试失败的文本：

`su.*fail`

- 匹配文本字符串“WARNING:”和被日志文件中的多行（数目未知）分隔的“Invalid login attempt”（需要 PCRE）：

`/^WARNING:(.*\n)*Invalid login attempt/m`

- 匹配文本“error”，不区分大小写（需要 PCRE）：

`/error/i`

常规自动监测器

实例条件 RegEx、RegEx 2、RegEx 3：

- 匹配包含“WiFi”的文本：

`WiFi`

- 匹配包含“C:”、“D:”或“E:”的文本：

`C:|D:|E:`

进程自动监测器

实例条件 RegEx、RegEx 2、RegEx 3:

- 匹配以 “svchost” 开头的进程:
`^svchost`
- 匹配包含 “sql” 的进程:
`sql`
- 匹配以 “svc” 结尾的进程:
`svc$`
- 匹配 “cmd” 和 “CMD” 进程（需要 PCRE）:
`/cmd/i`
- 匹配以 “sql” 或 “svchost” 开头的所有进程:
`^sql|^svchost`

服务自动监测器

实例条件 RegEx、RegEx 2、RegEx 3:

- 匹配显示名称中带有 “SQL” 的所有服务:
`.*SQL.*`
- 匹配显示名称以 “SQL” 开头的所有服务:
`^SQL`
- 匹配显示名称以 “SQL”（不区分大小写）开头的所有服务:
`/^SQL/i`

详细信息:

[Perl 兼容的正则表达式 \(PCRE\) 支持](#) (p. 145)

[多行正则表达式](#) (p. 146)

[配置文本模式排除](#) (p. 147)

[正则表达式示例](#) (p. 147)

Perl 兼容的正则表达式 (PCRE) 支持

Perl 兼容的正则表达式 (PCRE) 使您可以在指定 i18n 兼容的 (UTF-8) 正则表达式，同时定义支持正则表达式的监视器。这些监视器监控日志文件、进程、进程组、Windows 服务和 Windows 事件。您也可以使用该选项来创建更复杂的正则表达式。

您可以在两个正则表达式库之间进行选择：

- 标准 SystemEDGE 正则表达式库（默认）
- Perl 兼容的正则表达式 (PCRE) 库

由于分隔符和各种各样的特殊字符，PCRE 与标准正则表达式库并不是百分之百兼容。

要启用 PCRE 库，您可以通过添加以下行在 SystemEDGE 传统模式下手动编辑配置文件：

```
use_pcre
```

如果您在受管模式下编辑配置文件，则配置会在下次部署策略时清除。

您可以使用 CA Virtual Assurance 策略配置用户界面在受管模式下启用 PCRE 库。在用户界面中，使用“控制设置”选项卡中的“使用 Perl 兼容的正则表达式”选项。指定为该表达式激活 JavaScript 兼容模式的“<JS>”模式修饰符。

PCRE 涵盖 POSIX 基本正则表达式和 POSIX 扩展正则表达式以及 Perl 提供的正则表达式功能的子集。有关 PCRE 的详细信息，请访问 <http://www.pcre.org>。

输入正则表达式的受支持格式：

```
/regular expression/switch
```

正则表达式

注意：使用 PCRE 的建议方式是指定包含分隔符和可选开关 (/regex/i) 的正则表达式。

支持的开关列表

/i

PCRE_CASELESS（Perl 兼容的/i）：匹配大写和小写字母。

/m

PCRE_MULTILINE（Perl 兼容的/m）：跨多行匹配。

/s

PCRE_DOTALL (Perl 兼容的/s)：模式中的点元字符匹配任意值的字符，包括表示换行的字符。

/x

PCRE_EXTENDED (Perl 兼容的/x)：除非对模式中的空白数据字符进行转义或其位于字符类内，否则会将其忽略。

/E

PCRE_DOLLAR_ENDONLY：\$ 仅在末尾匹配。

/U

PCRE_UNGREEDY：重复量词的贪婪会反转，也就是说，默认情况下它们是不贪婪的，但是如果其后跟一个问号将会变为贪婪。

/<JS>

PCRE_JAVASCRIPT_COMPAT：指定 JavaScript 兼容性。

支持的分隔符列表

除默认的正则表达式分隔符 (/) 之外，SystemEDGE 支持备用分隔符以简化在 UNIX 和 Linux 路径指定正斜杠的表达式。备用分隔符有 (#)、(|) 和 (:.)。

例如，以下正则表达式是等效的：

```
\/dir1\/dir2\/dir3\/filename/i
#dir1/dir2/dir3filename#i
|dir1/dir2/dir3filename|i
:dir1/dir2/dir3filename:i
```

多行正则表达式

Perl 兼容的正则表达式 (PCRE) 提供与出现在多行上的文本匹配的功能。您可以通过使用表达式中的 PCRE_MULTILINE (/m) 或 PCRE_DOTALL (/s) 开关使用该功能。

当在日志文件监视器正则表达式中检测到任何一个开关时，代理会在受监控日志文件的最后 8 KB 内搜索指定的表达式。

配置文本模式排除

您可以使用 Perl 兼容的正则表达式 (PCRE) 在正则表达式中配置文本模式排除。PCRE 库支持执行正则表达式模式匹配。通过 PCRE 库可以更加灵活地在模式匹配过程中指定要在正则表达式中包括或排除的模式。

在某些搜索条件中，仅当排除特定文本模式时才需要匹配文本模式（在日志文件中）。您可以通过创建唯一正则表达式并将表达式标识为排除模式以排除文本模式。例如，您可以创建一个表达式以仅识别包含“abc”（不包括“def”）的子字符串匹配。要在正则表达式中排除“def”，请使用否定断言以不匹配（如果主题中包含子字符串“def”）。

示例：配置文本模式排除

以下示例指定搜索选项搜索匹配所有字符串“abc_”（其后不跟随字符串“def”）的文本模式。在此，括号 () 用于对字符串和 ?! 进行分组 表示否定。

```
abc_(?!def)
```

正则表达式示例

该部分提供使用 `sysedge.cf` 文件中的监视日志文件指令将正则表达式添加到日志监视器表的示例。

简单表达式

示例：/the quick brown fox/

以下示例在句子中搜索正则表达式“/the quick brown fox/”：

输入

```
The quick brown fox jumps over the lazy dog.
```

```
the quick brown fox jumps over the lazy dog.
```

结果

```
the quick brown fox
```

不区分大小写的表达式

示例： `/The quick brown fox/i`

以下示例在句子中搜索正则表达式 `“/The quick brown fox/i”`。表达式 `“The quick brown fox”` 的搜索匹配（不区分大小写）。

输入

The quick brown fox jumps over the lazy dog.

the quick brown fox jumps over the lazy dog.

the quick brown FOX jumps over the lazy dog.

结果

The quick brown fox jumps over the lazy dog.

the quick brown fox jumps over the lazy dog.

the quick brown FOX jumps over the lazy dog.

多行模式—行首和行尾元字符

示例： `/^abc$/m`

以下示例在句子中搜索正则表达式 `“/^abc$/m”`。只有模式 `“abc”` 独立成行时该表达式才匹配。

输入

Lazy dog\nquick horse\nabc

结果

abc

多行模式—PCRE_DOTALL

示例： `/two.*three.*four/s`

以下示例在句子中搜索正则表达式 `“/two.*three.*four/s”`。指定 PCRE_DOTALL 模式修饰符之后，点元字符在新行上仍然匹配。

输入

one number\ntwo numbers\nthree numbers\nfour numbers\nfive numbers

结果

Two numbers\nthree numbers\nfour

扩展表达式—忽略空白字符

示例： `/^1234 # comment in input string/x`

以下示例在句子中搜索正则表达式 `“/^1234 #comment in extended re/x”`。该表达式在引用文本上匹配（不管输入中遇到的空白字符）。

输入

```
1234 #comment in input string
1234 #  comment in      input  string
1234#commentininputstring
```

结果

```
1234 #comment in input string
1234 #  comment in      input  string
1234#commentininputstring
```

美元符号仅在末尾匹配

示例： `/X$/E`

以下示例在句子中搜索正则表达式 `“/X$/E”`。模式（`“X”`）中的文本仅在主题字符串末尾匹配。

输入

```
测试 X
X 有哪些用途
```

结果

```
X
X
```

JavaScript 兼容

示例： `/a[^]b/<JS>`

以下示例在句子中搜索正则表达式 `“/a[^]b/<JS>”`。指定为该表达式激活 JavaScript 兼容模式的 `“<JS>”` 模式修饰符。

输入

```
aXb
```

结果

```
aXb
```

SNMPv3 配置

与先前的版本相比，SNMPv3 提供了增强的安全级别。SNMPv3 有以下通信级别：

- **noAuthNoPriv** 类似于 SNMPv1 和 SNMPv2。消息附带有用户名，该用户名在发件人和收件人之间必须保持一致。
- **AuthNoPriv** 使用一致的用户名和密码。
- **AuthPriv** 使用用户名、密码和加密密钥。该密钥对邮件的正文部分进行加密。

为了保持信息一致，发件人和收件人必须同步且必须使用提供的加密工具保证信息安全。

您可以将 SystemEDGE 配置为使用基于 SNMPv3 的通信。该代理基于为 SNMPv3 定义的基于用户的安全模型 (USM)。

`sysedgeV3.cf` 文件包含定义 SNMP 管理员如何处理职责并指定访问主机时预期的安全级别的策略。文件位于 `CASYSEDGE_DATADIR` 目录，通过该文件您可以指定以下内容：

- SNMP 引擎 ID 前缀
- SNMPv3 安全用户
- 安全级别
- 身份验证协议和密码
- 加密协议及其密码

没有用于配置 `sysedgeV3.cf` 文件的用户界面。您必须直接在文件中配置 SNMPv3 设置。但是，您可以从 CA Virtual Assurance 管理器访问文件，并将配置的文件部署到系统。

配置 SNMPv3 引擎 ID

您必须指定一个文本 SNMP 引擎 ID 前缀（该前缀通过代理 SNMP 库与进程 ID 和 IP 地址进行连接），以在 SNMPv3 中运行代理。

要配置 SNMPv3 引擎 ID，请将以下行添加到 `sysedgeV3.cf` 文件：

```
SNMP_V3_ENGINE_ID <engineIdPrefix>
```

`engineIdPrefix` 的默认值为 `SystemEDGEAdmin`。不要在配置的字符串中使用空格。

例如，SNMPv3 配置文件中的以下行将字符串 `CompanySNMPV3ADMIN` 指定为 SNMP 引擎 ID 的前缀：

```
SNMP_V3_ENGINE_ID CompanySNMPV3ADMIN
```

配置 SNMPv3 用户信息

您可以使用 `SNMP_V3_USER_INFO` 关键字在 `sysedgeV3.cf` 文件中指定 SNMPv3 USM 用户和安全信息。该关键字的所有参数必须位于同一行且必须按照空格分隔的指定顺序。

要配置 SNMPv3 用户信息，请将以下行添加到 `sysedgeV3.cf` 文件：

```
SNMP_V3_USER_INFO *|access[addresses] userName securityModel securityLevel  
[authProtocol authPassword [privProtocol privPassword]]
```

access

指定读取或写入访问权限。值“读取”或“写入”为必填项。

addresses

指定一个 IP 筛选以筛选指定的 IP 地址或子网发出的请求。该字段不是必填字段。如果未指定，则所有主机都可以访问代理信息。有关详细信息，请参阅 SNMPv3 用户的地址筛选。

userName

指定允许访问的 SNMPv3 安全用户名。

securityModel

指定正在使用的 SNMPv3 安全模型。`SystemEDGE` 代理当前仅支持基于用户的安全模型 (USM)。仅支持值 3。

securityLevel

受支持的安全级别支持以下值：

noAuthNoPriv

表示没有配置供该 SNMPv3 用户使用的身份验证和隐私（加密）协议。

AuthNoPriv

表示已为该 SNMPv3 用户配置身份验证协议，但并没有配置隐私协议。

AuthPriv

表示已配置供该 SNMPv3 用户使用的身份验证和隐私协议。

authProtocol

指定要使用的身份验证协议。MD5 和 SHA 是当前唯一使用的协议。如果已设置 AuthPriv 或 AuthNoPriv 安全级别，则您应当仅指定该选项。

authPassword

指定身份验证协议使用的 SNMPv3 用户身份验证密码（密钥）。只有当已设置 authProtocol（MD5 或 SHA）时，才需要指定 authPassword。

privProtocol

指定 SNMPv3 用户使用的加密（隐私）协议。DES、3DES 和 AES 是唯一受支持的协议。如果您指定加密协议，则还必须指定 authProtocol 和 authPassword。如果您指定 privProtocol，则 AuthPriv 是唯一受支持的安全级别。

privPassword

指定加密协议使用的 SNMPv3 用户加密密码（密钥）。只有当您设置 privProtocol 时，才需要该参数。

您可以将读取或写入访问权限分配到不同的安全级别。例如，无身份验证和无隐私 (noAuthNoPriv) 的安全级别可相当于公共团体字符串，而身份验证和隐私 (AuthPriv) 可相当于管理员团体字符串。

示例

有效 SNMPv3 用户定义的示例如下：

```
SNMP_V3_USER_INFO *|read joe1 3 AuthPriv MD5 apass AES ppass
SNMP_V3_USER_INFO *|read joe2 3 AuthPriv SHA apass DES ppass
SNMP_V3_USER_INFO *|write|192.168.29.0 joe3 3 AuthPriv SHA apass 3DES ppass
SNMP_V3_USER_INFO
*|write|100.10-255.100.101,e000-ffff:f0ff:bef0:*,192.168.120.0,*:1 joe4 3
AuthNoPriv SHA evansar
SNMP_V3_USER_INFO *|read joe5 3 noAuthNoPriv
```


SNMPv3 用户的地址筛选

SNMPv3 配置文件使您可以指定一个源地址筛选以将对信息的访问限制为仅允许匹配地址筛选的源地址。

以下为适用于地址筛选字段的受支持语法和使用准则：

- 支持完整的 IPv4 地址。

示例：

`130.10.100.101`

该行指定允许源地址 `130.10.100.101` 发出的请求对代理进行访问。

- 支持完整的 IPv6 地址。为 IPv6 地址指定十六进制表示法。您可以指定小写或大写字符。以下是 IPv6 地址示例：

`Ea2f:fe90:abcd:0000:230:a2f:200:ad01`

该行指定允许源地址 `ea2f:fe90:abcd:0000:230:a2f:200:ad01` 发出的请求对代理进行访问。

- 支持主机名。但是，我们强烈建议使用 IP 地址而非主机名作为筛选，因为一个服务器可能有多个与之关联的 IP 地址，但并不是所有的 IP 地址都与 DNS 中的主机名相关。

示例：

`box1.domain.com`

该行指定允许 `box1.domain.com` 发出的请求对代理进行访问。

注意：如果您指定主机名，则代理必须能将接收的 IP 地址解析为指定的源主机名。

- 支持通配符星号 (*)。使用通配符指定可以访问代理的地址范围。

示例：

`130.10.*.101`

该行指定允许以 `130.10` 开头且以 `101` 结尾的源 IPv4 地址（即，`130.10.0.101` 到 `130.10.255.101`）发出的请求对代理进行访问。

示例：

Ea2f:fe90:*.ad01

该行指定允许源 IPv6 地址（以 ea2f:fe90 开头，以 ad01 结尾）发出的请求对代理进行访问。

- 如果您指定通配符 (*), 则它应当是该 IP 字段中的唯一字符。IP 字段是点 “.”（对于 IPv4 地址）之间和冒号 “:”（对于 IPv6 地址）之间的字符。

例如, 130.10.3*.200 条目是无效的, 因为 3 不能与星号在相同的点内。Ea2f:fe*:abcd:0000:230:a2f:200:ad01 条目也是无效的, 因为 “fe” 不能与星号在相同的冒号内。

- 在有多个通配符的情况下, 将只延伸最左边的通配符以填充缺失的字段。例如, 输入 2002:*.12f4:*.1012 的方式与输入 2002:*.*:*.12f4:*.1012 的方式是相同的。
- 会将 0（零）视为通配符。

示例：

130.10.120.0

该行指定允许 IPv4 子网 130.10.120 发出的请求对代理进行访问。按照这种方式输入地址与指定 130.10.120.* 是一样的。

示例：

Ea2f:fe90:0:ad01

该行指定允许 IPv6 地址（以 ea2f:fe90 开头，以 ad01 结尾）发出的请求对代理进行访问。这与输入 Ea2f:fe90:*.ad01 是一样的。

- 使用两个或多个零删除单个零的通配符行为, 并将其视为实际的零。

示例：

130.000.120.00

该行指定允许 IPv4 地址 130.0.120.0 发出的请求对代理进行访问。

示例：

Ea2f:*.0000:000:ad01

该行指定允许 IPv6 地址（以 ea2f 开头，以 0:0:ad01 结尾）发出的请求对代理进行访问。

- 支持部分 IP 地址。部分地址被视为后跟通配符 (*) 的部分地址。允许以部分地址开头的源地址发出的任何请求。

例如, 130.10 与指定 130.10.* 是一样的。此外, Ea2f:fe90 与指定 Ea2f:fe90:* 是一样的。

- 对于 IPv6 地址, 会将两个连续的冒号 (::) 视为两个冒号之间的通配符 (:*)。
- IPv4 和 IPv6 地址支持使用破折号 (-) 的地址范围。任何或所有字段均可以定义范围。

示例:

130.10-255.100.101

该行指定允许源 IPv4 地址范围 (从 130.10 到 130.255) 发出的请求对代理进行访问

e000-ffff:f0ff:bef0:*

该行指定允许源 IPv6 地址范围 (从 e000:* 到 efff:*) 发出的请求对代理进行访问。

- 支持用逗号 (,) 分隔的多个筛选器。

示例:

130.10-255.100.101, e000-ffff:f0ff:bef0:*, 130.10.120.0, box2.domain.com

- 不允许出现空格。

注意: 当一个主机有多个接口或 IP 地址时, 您必须提供访问代理的所有接口和 IP 地址, 因为可以通过任何接口或 IP 地址发送请求 (如, `snmpget`、`walktree`、`snmpset` 等)。您必须在代理 IP 筛选中定义分配给主机的所有 IP 地址, 否则主机将没有权限访问代理。作为定义所有 IP 地址和接口的备用方法, 您可以指定主机名 (只要您了解使用主机名的限制)。

配置 SNMPv2c 和 SNMPv3 陷阱目标

`sysedgeV3.cf` 文件使您可以指定代理应该发送的陷阱类型以及陷阱属性。支持以下陷阱类型：

- SNMPv2c 陷阱
- SNMPv2c 通知（也称为通知请求和已确认陷阱）
- SNMPv3 陷阱
- SNMPv3 通知（也称为通知请求和已确认陷阱）

注意：在 `sysedge.cf` 而非 `sysedgeV3.cf` 中对 SNMPv1 陷阱目标进行配置。

使用 SNMP 团体字符串发送 SNMPv2c 陷阱和 SNMPv2c 通知。必须在 `sysedge.cf` 中定义该团体字符串（例如，团体全局只读）。

使用 SNMPv3 用户凭据发送 SNMPv3 陷阱和 SNMPv3 通知。必须在记录陷阱定义前在 `sysedgeV3.cf` 配置文件中对 SNMPv3 用户进行定义。

要将代理配置为向指定的目标主机发送 SNMPv2 陷阱，请将以下行添加到 `sysedgeV3.cf` 文件：

```
SNMP_V2_TRAP_INFO <destination_host>|<port> <trap_context> <community>  
<trap-encoding>
```

要将代理配置为向指定的目标主机发送 SNMPv2c 通知陷阱（也称为已确认陷阱和通知请求），请将以下行添加到 `sysedgeV3.cf` 文件：

```
SNMP_V2_NOTIFICATION_INFO <destination_host>|<port> <trap_context> <community>  
<timeout> <num_of_retries> <trap-encoding>
```

要将代理配置为向指定的目标主机发送 SNMPv3 陷阱，请将以下行添加到 `sysedgeV3.cf` 文件：

```
SNMP_V3_TRAP_INFO <destination_host>|<port> <trap_context> <SNMPv3_user>  
<trap-encoding>
```

要将代理配置为向指定的目标主机发送 SNMPv3 通知陷阱（也称为已确认陷阱和通知请求），请将以下行添加到 `sysedgeV3.cf` 文件：

```
SNMP_V3_NOTIFICATION_INFO <destination_host>|<port> <trap_context>  
<SNMPv3_user> <timeout> <num_of_retries> <trap-encoding>
```

destination_host

指定要向其发送陷阱的主机。您可以指定一个主机名或 IP 地址。

port

在要发送陷阱的目标主机上指定端口号。

trap_context

*（星号）是该字段唯一支持的值。该值为必填项。

community

（仅限 SNMPv2c）指定要用来发送陷阱的 SNMP 团体字符串。也必须在 `sysedge.cf` 文件中对该字符串进行定义。

SNMPv3_user

（仅限 SNMPv3）指定要用来发送陷阱的 SNMPv3 用户。也必须在 `sysedgeV3.cf` 文件中对该用户进行定义。

timeout

（可选，仅限通知）指定超时前等待通知发送确认消息的时间（秒）。

num_of_retries

（可选，仅限通知）指定超时后重试发送通知的次数。

trap-encoding

（可选）指定发送陷阱时使用的编码类型。这类似于在 SNMPv1 中配置陷阱编码。有关详细信息，请参阅[配置 SNMPv1 陷阱目标](#) (p. 124)。

示例

以下示例在 SNMPv3 用户 `osmanager` 的端口 `6666` 上向主机 `sysmanager` 发送了一个 SNMPv3 陷阱通知，等待确认（超时时间为三十秒），最多重试两次：

```
SNMPV3_NOTIFICATION_INFO sysmanager|6666 * osmanager 30 2
```

以下示例使用专用团体在端口 `162` 上向主机 `localhost` 发送 SNMPv2c 陷阱：

```
SNMPV2_TRAP_INFO localhost|162 * private
```

以下示例使用公共团体在端口 `162` 上向主机 `localhost` 发送 SNMPv2c 陷阱通知，等待确认（超时时间为三十秒），最多重试三次：

```
SNMPV2_NOTIFICATION_INFO localhost|162 * public 30 3
```

加密 SNMPv3 配置文件

使用加密（隐私）协议 DES 作为身份验证协议提供 SNMPv3 加密，使用 CA 定义的默认密钥提供 SHA。可以在中央位置生成加密配置文件并在多个主机间共享该文件。

为了更加安全，您可以加密 SNMPv3 配置文件 `sysedgeV3.cf`。您必须创建该文件的明文版本、将其加密并将加密的文件安装在适当的目录。

如果您使用可选的 `-L` 命令行参数，实用工具将检测控制台和语言目录的当前区域设置是否可用。如果未找到语言目录，实用工具将重新使用英语作为默认语言。

加密 SNMPv3 配置文件

1. 在 `sysedgeV3.cf` 中测试并验证语法。
2. 使用以下命令加密 `sysedgeV3.cf`:

```
se_enc -i sysedgeV3.cf -o sysedgeV3.cf.crypt -m 2 -d 2
```
3. 将明文 `sysedgeV3.cf` 移动到存档区域。
4. 将 `sysedgeV3.cf.crypt` 重命名为 `sysedgeV3.cf`。
5. 将 `sysedgeV3.cf` 复制到代理数据目录，或通过从 CA Virtual Assurance 管理器进行远程部署应用该文件。

注意：可在多个主机间共享 SNMPv3 配置文件。

禁用 SNMPv1 和 SNMPv2c

要关闭代理的 SNMPv1 和 SNMPv2c 通信，请在 `sysedge.cf` 文件中注释掉或删除包含以下内容的行：

```
community <name> <access> <addresses>
```

SystemEDGE Control Panel for Windows

作为单机 Windows 服务，SystemEDGE 有自己的 SystemEDGE 控制面板。要查看 SystemEDGE 控制面板，请打开“控制面板”对话框并双击 SystemEDGE。

您可以使用 SystemEDGE 控制面板执行以下任务：

- 启动和停止代理
- 查看团体字符串和陷阱目标
- 打开配置文件
- 查看 sysedge.log 文件
- 运行 diagsysedge 实用工具
- 查看《用户指南》和《版本说明》

第 6 章： 启动代理

本章说明如何启动 SystemEDGE 代理。在您执行该操作之前，您必须为环境配置代理。有关详细信息，请参阅“代理配置”一章。

注意：在如本章所述启动代理后，您可以使用 `diagsysedge.exe` 程序验证您的代理是否正在运行。有关详细信息，请参阅“故障排除和用法建议”一章。

此部分包含以下主题：

[在 Windows 系统上启动/停止代理](#) (p. 161)

[在 UNIX 系统上启动/停止代理](#) (p. 162)

[命令行启动选项](#) (p. 163)

[在系统启动时自动启动代理](#) (p. 164)

[代理热启动](#) (p. 165)

[维护模式](#) (p. 165)

在 Windows 系统上启动/停止代理

对于 Windows 系统，您可以从命令行、“服务”控制面板或 SystemEDGE 控制面板手动启动 SystemEDGE。

要从命令行启动/停止代理，请从以下列表输入适当的命令：

```
net start sysedge
```

```
net stop sysedge
```

该代理使用您在安装过程中指定的端口和代理安装数据目录中的默认配置文件作为服务在后台启动。

您也可以使用非默认选项或非默认配置文件在前台运行代理。有关详细信息，请参阅[命令行启动选项](#) (p. 163)。

从“服务”控制面板启动（停止）代理

1. 选择“开始”、“管理工具”、“服务”。

出现“Windows 服务”对话框。

2. 右键单击 SystemEDGE 并选择“启动(停止)”。

代理即会启动（停止）。

从 SystemEDGE 控制面板启动代理

1. 访问控制面板并双击 SystemEDGE。
此时将显示 SystemEDGE 控制面板。
2. 单击“启动(停止)代理”。
代理即会启动（停止）。

注意：在 Windows XP 类别视图中，SystemEDGE 控制面板位于“网络和 Internet 连接”下。在 Windows XP 经典视图中，“SystemEDGE 控制面板”图标位于显示器的主级。

在 UNIX 系统上启动/停止代理

在 UNIX 系统上，您应使用提供的服务启动脚本启动或停止 SystemEDGE。在少数情况下（例如，由 CA Support 指导时），您可以直接手动启动代理。

UNIX 系统的服务启动脚本

您可以使用服务启动脚本在 UNIX 系统上自动启动、停止或重新启动代理。将在安装过程指定的端口号上启动代理。

适用于所有 UNIX 平台的服务脚本如下所示：

```
CASYSEDGE/bin/sysedgectl [start|stop|status|restart|probe]
```

要从 UNIX 的命令行启动代理，请从 *Install_Dir/bin* 目录运行以下命令：

```
./sysedgectl start
```

该代理使用您在安装过程中指定的端口和代理安装数据目录中的默认配置文件启动。

您还可以使用非默认选项或非默认配置文件在后台运行代理。有关详细信息，请参阅[命令行启动选项](#) (p. 163)。

示例

```
cd $CASYSEDGE/bin  
  
./sysedgectl start  
./sysedgectl restart  
./sysedgectl stop
```

命令行启动选项

该部分描述在 Windows 和 UNIX 系统上启动 SystemEDGE 代理的命令行选项。使用这些选项，您可以使用端口和配置文件的默认值以外的值启动代理。这些选项使您能够从命令 Shell 运行代理的其他实例（如测试配置更改）。

注意：在代理启动时，将现有日志文件另存为 “*filename.0*”（扩展名为零）。如果通过在受管模式下部署新的配置更改了现有运行时代理日志文件，将会以零扩展名保存该文件。

重要信息！代理安装和 CA Virtual Assurance 部署仅支持默认代理实例。

sysedge 二进制支持以下命令行选项：

```
[-p port] [-f config file] [-e SNMPV3 config file] [-m monitor file] [-h] [-b] [-d]
```

-p port

指定侦听传入 SNMP 消息的端口。指定一个不同于在安装过程中输入的端口，以运行单独的代理实例。保留端口 UDP/1691，用作运行 SystemEDGE 代理的备用端口。如果您使用的是安装过程中指定的端口，请勿指定该参数。

请注意，只有默认代理实例与 CA Virtual Assurance 管理器进行通信。因此，从 CA Virtual Assurance 接收的任何配置文件都将放入与默认端口相对应的 CASYSEDGE_DATADIR 目录。

-f config file

指定用于 sysedge.cf 文件（而非 CASYSEDGE_DATADIR 中的默认配置文件）的路径名。

-e SNMPV3 config file

指定用于 sysedgeV3.cf 文件（而非 CASYSEDGE_DATADIR 中的默认 SNMPv3 配置文件）的路径名。

-m monitor file

指定用于 sysedge.mon 文件（而非 CASYSEDGE_DATADIR 中的默认监视器表配置文件）的路径名。

-h

显示命令帮助。该选项列出可用的命令行选项。

-b

（仅限 UNIX）在后台运行代理。该代理作为后台进程运行并与控制终端断开连接。从启动脚本启动代理时请使用该标志。

-d

以调试模式运行代理。该选项将代理记录级别设置为 **debug3**。有关记录机制和记录级别的详细信息，请参阅“代理配置”一章。

在系统启动时自动启动代理

SystemEDGE 安装会自动将脚本复制到系统启动区域，以便可以将其用于在系统启动时自动启动代理。

某些系统可能需要一些其他步骤才能使代理在系统启动时自动启动。

该部分包括以下操作系统的说明：

- Windows
- UNIX

注意：在代理启动时，将现有日志文件另存为“*filename.0*”（扩展名为零）。如果通过在受管模式下部署新的配置更改了现有运行时代理日志文件，将会以零扩展名保存该文件。

在 Windows 系统上自动启动代理

SystemEDGE 服务在系统启动时默认启动。

验证是否将代理设置为自动启动

1. 打开“Windows 服务”对话框。
2. 右键单击“CA SystemEDGE”，然后选择“属性”。
3. 验证是否在“常规”选项卡的“启动类型”下拉列表中选择“自动”。

在 UNIX 系统上自动启动代理

SystemEDGE 代理在系统启动时会默认使用服务启动脚本在 UNIX 系统上启动。将会连接到您在安装过程中指定的默认端口，并了解安装路径。

代理热启动

与 CA Virtual Assurance 集成后，SystemEDGE 代理可以实施部署的配置文件而无需冷重新启动。相反，从 CA Virtual Assurance 接收到配置更改会触发热启动，其中代理可以在运行时实施更改。在执行热启动时，代理无需断开并重新绑定到接口、重建 OID 变量或重新启动 AIM，因此可以更快地进行配置并消耗更少的资源。

有关通过 CA Virtual Assurance 部署运行时配置更改的详细信息，请参阅《CA Virtual Assurance 管理指南》。

热启动例外

与 CA Virtual Assurance 集成时，SystemEDGE 代理会在接收到大多部署的配置更改时执行热启动。但是，代理在以下任何情况下仍会充分重复利用：

- 通过用户命令手动执行操作。您仍可以随时在 Windows 上使用命令行或 SystemEDGE 控制面板启动或停止代理。
- 如果代理接收到需要重新启动的配置文件更改，则隐式执行操作。删除 AIM 或者更改 `sysedge_fips_mode` 参数会触发冷重新启动。代理使用这些更改触发的进程自动重新启动。

注意：升级 CAM 也会触发代理自动重新启动。

- 如果手动将 CAM 停止，则隐式执行操作，因为 SystemEDGE 是作为 CAM 客户端注册的。代理在这种情况下不会自动重新启动；您必须在 CAM 再次运行后手动启动代理。

维护模式

SystemEDGE 代理以维护模式运行，其中代理会继续收集度量标准并响应 SNMP 请求，但是不处理监视器或发送状态更改陷阱。如果您了解计划停机或维护且不愿在该时间段接收陷阱时，该模式很有用。您可以将代理置于维护模式，以避免在这种情况下停止和重新启动代理。

有关启用维护模式的详细信息，请参阅“代理配置”一章。

第 7 章： Systems Management Empire MIB

本章提供了通过企业特定 Systems Management Empire MIB 可用的管理信息示例。

此部分包含以下主题：

[Systems Management Empire MIB 概述](#) (p. 168)

[系统组](#) (p. 169)

[用户表](#) (p. 170)

[组表](#) (p. 171)

[进程表](#) (p. 171)

[用户表](#) (p. 172)

[远程外壳组](#) (p. 172)

[性能组](#) (p. 173)

[进程间通信组](#) (p. 173)

[缓冲区组](#) (p. 174)

[目录名称查找缓存组](#) (p. 175)

[AIX 逻辑分区组](#) (p. 175)

[工作负荷管理程序 \(WLM\) 类的表](#) (p. 176)

[陷阱团体表](#) (p. 176)

[NT 系统组](#) (p. 176)

[分布式系统组](#) (p. 180)

[历史记录表](#) (p. 181)

[日志监视器表](#) (p. 182)

[磁盘统计信息表](#) (p. 182)

[CPU 统计信息表](#) (p. 183)

[扩展组](#) (p. 184)

[进程监视器表](#) (p. 184)

[进程组监视器表](#) (p. 185)

[聚合状态表](#) (p. 185)

[LPAR 组](#) (p. 185)

[镜像表](#) (p. 185)

[自动监测器表](#) (p. 186)

[不受支持的 Systems Management Empire MIB 对象](#) (p. 186)

Systems Management Empire MIB 概述

Systems Management Empire MIB 模块定义一系列用于管理主机系统的对象。将 MIB 组织成组和表以监控以下类型的信息：

- 系统信息
- 已安装设备
- 内核配置
- 数据流
- 用户和组
- 流程
- 远程命令执行
- 性能
- 进程间通信 (IPC)
- 网络缓冲区的缓冲区统计信息
- 数据流缓冲区
- I/O 缓冲区缓存
- 目录名称查找缓存
- 逻辑分区
- NFS 和 RPC 统计信息
- Windows 特定的统计信息
- 磁盘和 CPU 统计信息

MIB 还包括监控和对象聚合表。您可以使用自主监控条目监控 MIB 组和表的特定对象。

注意：根据本地安全策略，您可以禁用对任意表的支持。有关禁用对表的支持的详细信息，请参阅“代理配置”一章。

您可以通过指定对象 OID 或对象类别、实例和属性创建自主监控条目。使用自主监控功能可以针对定义的阈值监控任何 MIB 对象。您可以保持对象状态，该状态基于当前检索到的值。当违反阈值时，代理会发送状态更改陷阱。有关详细信息，请参阅“自主监控”一章。

Systems Management Empire MIB 中的对象的顶级 OID 是 1.3.6.1.4.1.546。参考 MIB 文件以派生完整的 OID。

本章包含以下信息：

- 所有 Systems Management Empire MIB 组和表的列表

- MIB 组和表的默认类值
- MIB 表的默认实例值

要定义监控条目，请使用各个组和表的默认类和实例值。`sysedge.oid` 文件将保留每个属性的默认值，您可以用来定义自主监控条目而无需指定 OID。

本章为所有组、表定义默认类并为所有表定义默认实例；监控 MIB 组中的属性时不需要该实例值。默认属性值是 MIB 属性的名称。

注意：有关相反的 OID 查找功能的详细信息，请参阅“自主监控”一章。

有关 MIB 对象和各个属性名称的详细说明，请参阅代理安装的 `mib` 子目录中的 `empire.asn1` 文件。

系统组

要监控基本信息，请在 `sysedge` 系统组中检索对象值。

类： `sysedgeSystem`

设备表

设备表提供了安装在主机系统上的设备和文件系统的信息。

类： `devTableEntry`

实例： 特定设备的 `devMntPt` 属性值

文件系统空间监控

设备表的 `devCapacity` 列将显示已用的文件系统空间的百分比。使用 `devCapacity` 列来检查文件系统空间使用是否已满。

可以使用自主监视器表可指示代理监控 `devCapacity` 值，并在该值违反阈值时向管理系统发送陷阱。例如，使用以下指令来配置代理以监控 `/usr` 目录，并在 `/usr` 目录的使用容量超过 90% 时发送关键状态更改陷阱：

```
monitor 11 0x00 60 absolute > 90 'Monitor device space usage' '' devTableEntry  
'/usr' devCapacity
```

有关设置阈值的详细信息，请参阅“自主监控”一章。

卸载已安装的设备

要卸载已安装的设备，针对设备的 `devUnmount` 列发出 **SNMP Set** 命令以将列值设为 **delete (1)**。代理将卸载该设备并从设备表中删除其条目。

注意：对于 **SNMP Set** 操作，请使用向您授予读写访问权限的社区名称。

内核配置组

要识别在系统上运行的内核版本并确定内核配置详细信息，请在“内核配置”组中检索对象。

类：kernelConfig

启动配置组

使用“启动配置”组中的 **MIB** 对象来确定系统会将哪个设备和分区用于根文件系统。您也可以确定转储文件系统、交换空间以及每一系统各自包含的块数。

类：bootconf

数据流组

数据流 I/O 子系统提供了用户空间中的应用程序与主机的内核空间中的驱动程序之间的数据传输/处理路径。您可以通过检索数据流组中的对象来监控数据流子系统的运行状况。

类：streams

用户表

使用用户表可检索有关系统上已创建的用户帐户的信息。该表中的每一行表示一个用户帐户。

类：userEntry

实例：特定用户的 `userLoginID` 属性的值

组表

使用组表可检索有关主机系统上已创建的用户组的信息。该表中的每一行表示 `/etc/group` 文件中定义的一个组。

类: `groupEntry`

实例: 特定组的 `groupName` 属性的值

进程表

使用“进程”表来确定哪些进程当前在系统上运行和进程度量标准。

类: `processEntry`

实例: 特定进程的 `processName` 属性的值

更改进程的 nice 值（仅 UNIX）

要设置或更改进程的 `nice` 列的值，请对特定进程的 `processNice` 列发出 `SNMP Set` 命令。例如，您可以通过在进程的 `nice` 列中设置优先级值来增加或减少进程的排定优先级。

有关值的列表，请参阅 `UNIX nice(1)` 手册页或 MSDN 网站上的 `Windows SetPriorityClass()` 函数说明。

将信号发送到进程

要将信号发送到进程，请对特定进程的 `processKill` 列发出 `SNMP Set` 命令以设置适当的信号值。

注意：在 Windows 上，仅支持信号数 `SIGKILL (9)`。

例如，从远程网络管理站控制台中，您可以通过将未经授权的进程的 `processKill` 列的值设置为 9 来终止该进程。数字 9 代表 `UNIX SIGKILL` 信号，可以终止进程。

有关信号的详细信息，请参阅 `UNIX signal (3V)` 手册页。

用户表

使用用户表可以获取当前登录到主机系统的用户的列表。该表用于在给定时间内获取特定系统的用户详细信息。该表中的每一行表示一个当前用户。

类: whoEntry

实例: 特定用户和设备的 whoName 和 whoDevice 属性的值

远程外壳组

使用远程 Shell 组在远程主机系统上运行 Shell 脚本和程序。

类: remoteShell

远程 Shell 组中的 MIB 变量允许您指定以下命令:

- 命令语法 (shellCmd)
- 输出文件 (shellOutput)

远程 Shell 组中的变量允许您查看以下命令信息:

- 执行 Shell 命令的退出代码 (shellExitstat)
- 输出文件的命令输出 (shellOutputContents)

运行远程命令

要运行远程命令, 请对 shellCmd MIB 变量执行 SNMP Set。在执行 SNMP Set 操作时, 请使用向您授予读写访问权限的社区名称。

在命令提示符处输入以下内容来运行远程命令:

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

代理 fork/execs 指定的命令, 标准输出 (stdout) 和标准错误 (stderr) 重定向到由 shellOutput MIB 变量命名的文件。在命令完成执行后, 该代理会将命令的退出状态置于 shellExitStat。

性能组

使用性能组来跟踪主机的操作系统的运行状况和性能。

类: kernelperf

使用性能组属性来监控在固定的时段内采样得到的统计信息。

注意: 采样间隔取决于系统。

有关这些 MIB 对象的详细说明，请参阅代理分发的 mib 子目录中的 empire.asn1 文件。

进程间通信组

您可以通过 IPC 组检索下表之一来跟踪系统上正在使用的进程间通信 (IPC) 机制：

消息队列表

提供有关当前正在使用的系统消息队列的信息。

类: msgqueEntry

实例: 特定消息队列的 queID 属性的值

共享内存表

提供有关当前正在使用的系统共享内存段的信息。

类: shmемEntry

实例: 特定共享内存段的 shmемID 属性的值。

信号表

提供有关当前正在使用的系统信号的信息。信号是操作系统或内核存储中的一个值，进程可以检查该值并进行更改，以协调多个进程竞争同一操作系统资源的活动。

类: semEntry

实例: 特定信号的 semID 属性的值

每张表中的行表示 IPC 使用情况的一个实例。每张表中的列不同。

删除 IPC 机制

使用 IPC 组表删除消息队列、共享内存段和信号。每一个 IPC 组表都包含一个特定列，供代理用来销毁 IPC 使用情况的实例。

要删除 IPC 机制，请对 `queDel`（消息队列表）、`shmemDel`（共享内存表）或 `semDel`（信号表）属性发出 `SNMP Set` 命令以将属性值设置为 `delete (1)`。

注意：对于 `SNMP Set` 操作，请使用向您授予读写访问权限的社区名称。

缓冲区组

缓冲区组包含用于监控以下类型的缓冲区的组和表：

- 网络或消息缓冲区
- 数据流缓冲区
- I/O 缓冲区缓存

消息缓冲区组

使用消息缓冲区组来监控系统使用消息缓冲区的方式。

类：`mbufs`

该组也包含用于跟踪每个消息缓冲区类型分配的消息缓冲区分配表。

类：`mbufAllocEntry`

实例：特定消息缓冲区类型的 `mbufDesc` 属性的值。

数据流缓冲区组

该表定义了代理所在的节点的 `WLM` 类。请注意，只有当 `SystemEDGE` 代理正在基于 `AIX` 的操作系统上运行，并且操作系统中启用了工作负荷管理程序时，此表的数据才可用。

数据流缓冲区分配表

使用数据流缓冲区分配表可监控由数据流子系统使用的缓冲区的分配和使用情况统计信息。

类: `strbufAllocEntry`

实例: 特定数据流缓冲区的 `strbufAllocIndex` 属性的值

I/O 缓冲区缓存组

使用 I/O 缓冲区缓存组来跟踪基本磁盘 I/O 的 I/O 缓冲区分配和使用情况。

类: `ioBufferCache`

您还可以用图表表示这些计数器的值，以便检测 I/O 缓冲区活动的高峰期。

目录名称查找缓存组

使用目录名称查找缓存 (DNLC) 组可以监控将目录和文件名的映射缓存到 UNIX 上的 `vnode` 的缓存。该组包含该缓存的性能统计信息。

类: `dnlc`

AIX 逻辑分区组

通过 AIX 逻辑分区组，您可以监控有关 AIX 系统上的各个逻辑分区 (LPAR) 的信息。

类: `logicalPartition`

有关使用 AIX 逻辑分区组来监控 LPAR 环境的详细信息（包括每个 MIB 属性的说明），请参阅“AIX 逻辑分区”表。

工作负荷管理程序 (WLM) 类的表

您可以在陷阱社区表中查看当前配置的访问社区和其他 SNMP 特定的配置信息。您可以从 CA Virtual Assurance 用户界面编辑该表中的设置。该表包含支持 SNMPv1、SNMPv2 和 SNMPv3 的信息，具体取决于您正在运行的版本。

类: trapCommunityEntry

实例: 特定陷阱目标和端口的 tcDestinationAddr 和 tcDestinationPort 属性的值

陷阱团体表

您可以在陷阱社区表中查看当前配置的访问社区和其他 SNMP 特定的配置信息。您可以从 CA Virtual Assurance 用户界面编辑该表中的设置。该表包含支持 SNMPv1、SNMPv2 和 SNMPv3 的信息，具体取决于您正在运行的版本。

类: trapCommunityEntry

实例: 特定陷阱目标和端口的 tcDestinationAddr 和 tcDestinationPort 属性的值

NT 系统组

包含组和表的 NT 系统组专为 Windows 系统设计。

类: ntSystem

注意: 大多数情况下，SystemEDGE 代理对 Windows 和 UNIX 支持相同的 MIB 对象。但是，基础 Windows 操作系统不支持某些系统管理 MIB 对象。有关在 Windows 上不受支持的 MIB 对象的列表，请参阅《SystemEDGE 版本说明》。

NT 线程表

在 Windows 操作系统中，内核会排定要在处理器上运行的线程。线程是运行程序代码的进程的一部分。一个进程可以包含一个或多个线程。

使用“NT 线程”表可获取在系统上执行的每个线程的详细状态信息。

类：ntThreadEntry

实例：特定线程的 ntThreadPID 和 ntThreadNumber 属性的值

NT 注册表组

Windows 注册表充当数据库存储库，以便存储系统配置详细信息，例如硬件、用户帐号和应用程序。使用 NT 注册表组以查询 Windows 注册表。

类：ntRegistry

您可以通过查询以下内容确定您的注册表是否有超过为其设置的大小限制的危险：

- 注册表的当前大小 (MB) (ntRegistryCurrentSize)
- 注册表大小限制 (MB) (ntRegistrySizeLimit)

NT 服务表

Windows 提供多个作为 Windows 服务运行的程序。使用 NT 服务表可监控 Windows 服务。

类：ntServiceEntry

实例：特定服务的 ntServiceName 属性的值

NT 性能组

“NT 性能”组存在用于监控以下 Windows 性能度量标准：

- 系统
- 缓存
- 内存
- 页面文件

NT 系统性能组

NT 系统性能组提供可用于跟踪系统的 Windows 操作系统运行状况和性能统计信息。

类： ntSystemPerf

例如，负载过重的系统通常有高级别的系统活动，其中包括许多争用 CPU 时间的进程线程和排队等待运行的作业。您可以通过监控 ntContextSwitches 和 ntRunQLen MIB 对象来跟踪系统活动级别。

ntContextSwitches 计算已发生的上下文切换的次数。每当一个线程放弃 CPU 而由另一个线程代替时，都会发生上下文切换。高比率的上下文切换表示系统负载过高。ntRunQLen 表示是否存在正在等待运行的线程的备份。

NT 缓存性能组

NT 缓存性能组包含系统范围的缓冲区和文件系统缓存统计信息，这些统计信息用于确定系统缓存机制的效力。

类： ntCachePerf

在应用程序请求数据时，首先将数据映射到缓存中，然后从缓存复制到内存。随后，通过 Lazy Writer 系统线程或应用程序的完全写入调用将应用程序更改的数据从缓存写入到磁盘。

NT 内存性能组

NT 内存性能组包含内存和分页性能计数器以及统计信息。使用该组可跟踪系统的内存使用情况。

类： ntMemoryPerf

NT 页面文件性能组

Windows 使用 Pagefile.sys 分页文件处理对内存的任何额外需求。分页文件是操作系统保留的磁盘空间块，以便内存管理器可以在需要时创建内存空间。

您可以使用 NT 页面文件性能组监控分页文件的大小以确定是否可以增加更多系统内存。

类： ntPageFilePerf

NT 事件监视器组

Windows 使用事件日志捕获重要的系统和应用程序状态消息。“NT 事件监视器”组提供对这些事件日志的访问权限。

类：ntEvents

您可以使用“NT 事件监视器”表指示代理连续监控您指定的事件的 Windows 事件日志。只要在系统上生成匹配事件，代理就会使用陷阱消息通知管理系统。代理还可以运行操作命令以立即处理事件。“NT 事件”表提供每个监视器条目的详细状态信息。

类：ntEventMonEntry

实例：特定监视器条目的 ntEventMonDescr 属性的值

有关 NT 事件监视器的更多信息，请参阅“Windows 事件监控”一章。

NT 事件日志监控表

该表包含 NT 事件日志监视器条目。每个条目指定事件日志和正则表达式。代理定期查看事件日志以了解其是否更新。如果事件已更新，代理将扫描正则表达式的新条目。条目匹配正则表达式时，将陷阱 PDU 发送到相应的管理器。

NT 注册表和性能扩展组

SystemEDGE 代理提供了一个用于扩展 Systems Management Empire MIB 的强大机制，以将 Windows 注册表和性能计数器的信息包括在内。这包括配置数据（通常使用 regedit 查看）和性能数据（通常使用 perfmon 查看）。

使用该功能，您可以将代理自定义为返回系统和应用程序的其他配置和性能信息。例如，许多应用程序提供指定应用程序配置的注册表项，代理可以通过 SNMP 使这些项可用。SystemEDGE 代理还可以访问许多应用程序提供的统计信息。

在“NT 注册表和性能”组中提供该支持。

类：ntRegPerf

该组包含 128 个您可以配置的未指定的标量 MIB 变量。为了响应其中一个变量的 SNMP GET 请求，SystemEDGE 代理会读取 Windows 注册表并返回值。

有关使用和配置“NT 注册表和性能”组中的 MIB 对象的详细信息，请参阅“Windows 注册表和性能 MIB 对象”一章。

自主监视器表

使用自主监视器表可配置代理的自主监控功能以监控该表控制的任何基于整数的 MIB 变量。代理会自动监控 MIB 变量，并在违反阈值时使用状态更改陷阱通知管理系统。自主监视器表支持状态监控，该表会维护每个监视器的状态，并聚合其他表的自主监视器以计算对象状态。

类：monitorEntry

实例：特定自主监视器条目的 monDescr 属性的值

有关自主监视器表和监控 MIB 对象的更多信息，请参阅“自主监控”一章。

分布式系统组

分布式系统组提供在所有平台上监控和管理以下分布式系统设施需要的对象：

- RPC
- NFS

RPC 统计信息组

使用 RPC（远程过程调用）统计信息组可跟踪与内核 RPC 设施的使用情况相关的统计信息和计数器。

类：rpc

您可以用图表表示这些计数器的值以检测 RPC 活动的高峰时段。统计信息和计数器是根据其对客户端和服务器端协议操作的适用性划分的。

注意：有关 RPC 的详细信息，请参阅 RFC 1057。

NFS 统计信息组

使用 NFS 统计信息组跟踪与内核 NFS 设施的使用情况相关的统计信息和计数器。

类： nfs

您可以用图表表示这些计数器的值以检测 NFS 活动的高峰时段。统计信息和计数器是根据其对客户端和服务端协议操作的适用性划分的。

注意： 有关 NFS 的详细信息，请参阅 RFC 1094。

历史记录表

SystemEDGE 可以随着时间的推移跟踪各种基于整数的 MIB 对象（计数器、标尺等）的值，并存储这些值供以后检索。该功能（通常称为历史记录采样）可减少网络中管理站轮询的数量。该代理通过两个 SNMP MIB 表提供此功能：

历史记录控制表

类： empireHistoryCtrlEntry

实例： 特定历史记录控制表条目的 empireHistoryCtrlDescr 属性的值

历史记录表

类： empireHistoryEntry

实例： 特定“历史记录”表条目的 empireHistorySampleIndex 属性的值

历史记录控制表包含的参数描述了被采样并存储于历史记录表中的数据。控制表中的每行通过将值分配给表中的参数（列对象）定义特定的数据集合功能。“历史记录”表中的一行或多行（存储的采样）与该单个控制行关联。

每个控制表行都分配有一个唯一的索引值 (empireHistoryCtrlIndex)。行通过指定要采样的对象实例、采样频率（30 秒的倍数）以及要保留的采样数（存储段）来定义数据集合功能。与每个数据集合功能（控制表的行）关联的是“历史记录”表中的一组行。“历史记录”表中的每行（也称为存储段）包含一个采样时间间隔内收集的指定 MIB 对象的值。

每个采样时间间隔内，代理都会将行添加到具有与该数据集合功能的其他行相同的 `empireHistoryIndex` 值的历史记录表中。该新行与历史记录控制表中的单行相对应，且其 `empireHistorySampleIndex` 值比前一个采样的 `SampleIndex` 值大一个整数。

有关这些表中的属性以及将其用于历史记录收集的详细信息，请参阅“历史记录收集”一章。

日志监视器表

可以使用日志监视器表来配置代理，以监控正则表达式的系统日志文件。您还可以监控文件目录的大小和内容。

类： `logMonitorEntry`

实例： 特定日志监视器条目的 `logMonitorLogFile` 属性的值

例如，您可以配置代理以便监控 `/var/adm/sulog` 日志文件中的正则表达式 `su.*fail`。只要代理发现匹配，它就会将 SNMP 陷阱发送到配置的管理系统。

日志监视器表的每行均表示对日志文件中某个特定正则表达式的监控。

有关日志监视器表属性以及使用该表监控日志文件的详细信息，请参阅“日志文件监控”一章。

磁盘统计信息表

磁盘统计信息表提供磁盘的 I/O 统计信息。

类： `diskStatsEntry`

实例： 特定磁盘驱动器的 `diskStatsHostmibDevTableIndex` 属性的值

每个表条目提供了一个磁盘的最新磁盘统计信息。代理会定期（每 60 秒）检查每个磁盘的系统数据结构的状态，并将值记录在表中。

使用磁盘统计信息表时，请查看 `diskStatsLastUpdate` 列以获取有关上次更新该行的时间（以时间刻度为单位）的信息。通过此列提供了自上次查询该表以来统计信息是否已更新的信息。

为 AIX 系统配置磁盘性能统计信息收集

对于 AIX 系统，您必须手动配置系统以收集磁盘性能统计信息。只有在您指示操作系统跟踪这些统计信息时，SystemEDGE 才可监控它们。

注意：启用收集这些统计信息将减少磁盘吞吐量。

为 AIX 系统启用磁盘统计信息收集

1. 使用 root 权限登录到要监控的系统。
2. 在命令提示符下输入以下命令：

```
chdev -l sys0 -a iostat=true
```

为 AIX 系统禁用磁盘统计信息收集

1. 使用 root 权限登录到要监控的系统。
2. 在命令提示符下输入以下命令：

```
chdev -l sys0 -a iostat=false
```

CPU 统计信息表

“CPU 统计信息”表提供每个 CPU 的性能统计信息。

类：cpuStatsEntry

实例：特定 CPU 的 cpuStatsDescr 和 cpuStatsIndex 属性的值

每个表条目提供了一个 CPU 的最新统计信息。代理会定期（每 60 秒）检查每个 CPU 的系统数据结构的状态，并将值记录在表中。使用“CPU 统计信息”表可监控以下统计信息：

- CPU 类型 (cpuStatsDesc)
- 空闲、用户、系统或等待模式下花费的秒数 (cpuStatsIdle、cpuStatsUser、cpuStatsSys、cpuStatsWait)
- 上次统计信息更新时间 (cpuStatsLastUpdate)
- 逻辑 CPU 使用的物理处理器的百分比 (cpuStatsPC)
- 逻辑 CPU 使用的额定容量的百分比 (cpuStatsEC)
- 空闲、用户、系统或等待模式下的时间百分比 (cpuStatsIdlePercent、cpuStatsUserPercent、cpuStatsSysPercent、cpuStatsWaitPercent)

使用 CPU 统计信息表时，请查看 cpuStatsLastUpdate 列以获取有关上次更新该行的时间（以时间刻度为单位）的信息。通过此列提供了自上次查询该表以来统计信息是否已更新的信息。

类: cpuGroup

每个表条目提供了一组中所有 CPU 的最新统计信息。相关的 CPU 总数组将添加表中的度量标准以显示所有 CPU 的总数。它还提供 LPAR 使用的处理器和额定容量的总数。

扩展组

扩展组提供了一个用于扩展 Systems Management Empire MIB 的强大机制，以便将有关您的系统和应用程序的广泛信息包括在内。

类: extensionGroup

该组包含 232 个您可以配置的未指定的标量 MIB 变量。为了响应其中一个变量的 SNMP Get 请求，SystemEDGE 代理将调用为该变量指定的命令，并返回从命令返回的值。使用 SNMP Set 操作，您还可以将参数传递到命令。

有关使用扩展组变量的详细信息，请参阅“自定义 MIB 对象”一章。

进程监视器表

“进程监视器”表提供了一个强大又灵活的机制，用于对应用程序和进程进行监控。您可以监控关键进程和应用程序的各种属性，并在超过特定阈值时发送 SNMP 状态更改陷阱。像“自主监视器”表一样，“进程监视器”表也支持状态监控。

类: processMonEntry

实例: 特定进程的 pmonRegExpr 属性的值

有关进程监视器表属性和监控进程的详细信息，请参阅“进程和服务监控”一章。

进程组监视器表

“进程组监视器”表提供了一个强大又灵活的机制，用于对进程组进行监控。您可以使用正则表达式在进程组监视器表中创建一个条目。该表会汇总匹配表达式的进程的统计信息并监控该组以了解组更改。

类： processGroupMonEntry

实例： 特定进程组的 pgmonProcRegExpr 属性的值

有关进程组监视器表属性和监控进程组的详细信息，请参阅“进程组监控”一章。

聚合状态表

“聚合状态”表将具有相同类、实例和属性值的自主监视器和进程监视器聚合到一个表条目，以便维护聚合对象状态。该表允许代理维护自主监视器表和进程监视器表中条目的状态信息。

类： aggregateEntry

实例： 特定表条目的 aggObjInstance 和 aggObjAttribute 属性的值

注意： 可以配置代理以在更高级别上聚合。有关详细信息，请参阅“代理配置”一章。

LPAR 组

该组为逻辑分区 (LPAR) 提供物理和逻辑 CPU 的统计信息。

类： totalPhysicalCpuStats

类： logicalPartition

镜像表

Systems Management Empire MIB 包含聚合表和自主监视器表的镜像表。这些表包含与主表相同的属性和条目，在对主表进行交叉引用时非常有用。

自动监测器表

自动监测器表说明了代理的自动监控活动。此表中的一行会导致代理基于指定的条件自动为所有匹配实例创建监视器条目。

不受支持的 Systems Management Empire MIB 对象

SystemEDGE 在所有受支持的操作系统上支持 Systems Management Empire MIB，但该 MIB 模块中的某些 MIB 对象在基础操作系统上不受支持。此外，某些特定于 UNIX 的 MIB 对象不适用于 Windows。因此，无法使用 Windows 版本的代理来实施这些对象。

有关所有操作系统上不支持的 Systems Management Empire MIB 对象的列表，请参阅《*SystemEDGE 版本说明*》。

第 8 章： 自主监控

本章介绍如何使用 SystemEDGE 代理针对用户指定的阈值监控 MIB 变量。

此部分包含以下主题：

[自主监控概述](#) (p. 187)

[自主监视器的状态管理](#) (p. 188)

[自主监视器表](#) (p. 190)

[自主监控配置](#) (p. 204)

[用于自主监控的 edgemon 命令](#) (p. 223)

[删除自主监控条目](#) (p. 227)

自主监控概述

通过 SystemEDGE 代理，您可以动态地监控受其控制的任何基于整数的 MIB 变量。使用“自主监视器”表可指定要监控的变量、轮询时间间隔、比较运算符（大于、等于等）、阈值、重要级别、对象信息以及其他值。SystemEDGE 代理会自动监控该变量，并在满足您所指定条件的情况下将陷阱发送到管理系统。

注意：如果监视器使用状态模型，则 SystemEDGE 将针对受管对象发送陷阱，而不是针对每个已定义监视器发送陷阱。

SystemEDGE 代理可维护并更新您指定的任何自主监视器的状态值，并使用对象类、实例和属性值聚合定义了多个监视器的对象的状态。

代理还可以在受管系统上运行命令，以便执行管理功能以立即更正发生异常的原因。例如，您可以配置代理以监控根文件系统的百分比容量并在容量过满时通知管理器。

自主监视器的状态管理

SystemEDGE 状态管理模型可维护“自主监视器”表中所有条目的状态值，并允许您定义受管对象以派生总体对象状态。状态管理模型为“自主监视器”表中的条目提供以下功能：

- 可定义每个自主监视器条目的重要级别（代理使用其维护每个条目的状态值）。重要级别可转换为 CIM 兼容的状态。
- 可使用对象类、实例和属性值通过三级别命名方案将一个或多个受监控的资源组为一个或多个受管对象。该机制是灵活的，这样您就可以动态地分组不同功能区域的资源。
- 可创建超级受管对象的层次结构，其中特定对象的状态将传播至层次结构的更高级别上的对象。
- 可在受管对象的总体状态更改时通过状态更改陷阱发送通知

状态管理模型集成于 CA Virtual Assurance 对象模型中。AOM 数据库中的每个受管对象都拥有一个 CIM 兼容的状态，SystemEDGE 受管对象和状态通过 SystemEDGE 平台管理模块反映在 CA Virtual Assurance 中。通过 CA Virtual Assurance 管理 SystemEDGE 可使受管对象状态信息在 CA Virtual Assurance 用户界面中可视化，并可通过 CA Virtual Assurance 事件视图查看状态更改陷阱。

有关使 SystemEDGE 受管对象在 CA Virtual Assurance 中可视化的详细信息，请参阅 CA Virtual Assurance 文档。

本章的其余部分提供了有关配置自主监视器以使用状态管理模型的详细信息，包括属性值和语法示例。

受管对象创建

可使用对象类、实例和属性值控制“自主监视器”表中受管对象的创建。默认情况下，SystemEDGE 为每个唯一类、实例和属性自主监视器创建受管对象。如果多个自主监视器包含的这三个属性值相同，代理会将这些监视器组为单个受管对象并派生对象的总体状态。

为了说明使用该机制的灵活性，请考虑以下情形：

- 使用同样的类、实例和属性值来定义三个自主监视器条目，以监控 CPU 使用率。监视器将 CPU 使用率大于 85% 的重要级别定义为警告，将 CPU 使用率大于 90% 的重要级别定义为重要，将 CPU 使用率大于 95% 的重要级别定义为严重。代理将这些监视器组合到一个受管对象中，并且将最差的当前重要级别保留为对象状态。
- 使用同样的类、实例和属性值来定义三个自主监视器条目，以监控在企业中连接的不同功能区域的度量标准。例如，您可以组合基本磁盘空间自主监视器、来自 SRM AIM 的反应时间监视器以及来自远程监控 AIM 的文件系统监视器，将警告重要级别分配给每个自主监视器。在该方案中，只有当违反所有的三个自主监视器阈值（AND 标志）时，代理才发送警告状态更改陷阱。
- 通过针对对象类使用相同的值以及针对对象实例或属性使用星号值，定义自主监视器条目的超级聚合。该方案为每个聚合集合大量对象。例如，您可以将所有对象与顶级系统类和“文件系统”实例聚合，以便所有文件系统自主监视器将最差的当前状态传播到顶级对象。

注意：还可以将代理配置为聚合更高级别的所有对象。有关详细信息，请参阅“代理配置”一章。

有关这些方案的语法示例，请参阅[自主监控示例](#) (p. 209)。

如果您不定义对象类、实例和属性值，SystemEDGE 将使用位于代理安装的配置目录中的 sysedge.oid 文件为这些属性分配默认值。该文件为每个代理支持的 MIB 中的所有属性定义默认类和属性。它还为 [MIB 表](#) (p. 40) 中的每个 MIB 属性定义默认实例。默认分配确保代理仍为监控相同 MIB 属性的自主监视器创建对象。

有关 Systems Management Empire MIB 中的组、表和属性的默认分配的详细信息，请参阅“Systems Management Empire MIB”。

详细信息

[对象状态](#) (p. 60)

[对象合计](#) (p. 61)

[超级聚合](#) (p. 62)

[状态管理和聚合的工作原理](#) (p. 63)

聚合状态表

“聚合状态”表执行创建受管对象代表以及获取总体对象状态所必需的工作。“聚合状态”表是 **SystemEDGE** 自主监视器和进程监视器的已筛选视图。该表包括基于每个监视器条目提供的受管对象命名和状态信息计算的聚合状态信息。

该表使用相同的对象类、实例和属性值，为所有自主监视器和进程监视器创建“聚合状态”表条目（或在更高级别上在一起分组的对象）。自主监视器和进程监视器仅作为单独条目存在于“自主监视器”和“进程监视器”表中。“聚合状态”表维护这些监视器的分组，对表进行连续轮询，并在受管对象状态发生更改时发送状态更改陷阱。

注意：创建新的受管对象或删除现有的受管对象时，“聚合状态”表也发送陷阱。

有关“聚合状态”表如何维护受管对象及其状态的详细信息，请参阅“概念”一章。有关表列的说明，请参阅聚合状态表。

自主监视器表

Systems Management Empire MIB 的“自主监视器”表提供有关代理当前监控的每个 **MIB** 属性的信息。每一行表示定义的自主监视器条目。

对于每个条目，该表提供以下类型的信息：

- 索引
- 描述
- 代理轮询变量的间隔
- 示例类型（绝对值或增量值）
- Object ID
- 代理正在监控的当前值
- 阈值
- 最后示例调用
- 已经发送的陷阱数
- 最后陷阱

- 行状态
- 抽样的最小值和最大值
- 发送陷阱时采取的操作
- 标志
- 被监视器的索引取代
- 对象类
- 实例
- 属性
- 定义的重要级别
- 当前监视器状态

自主监视器表列

以下列表定义“自主监视器”表的列。有关自主监视器表及其字段的完整说明，请参阅位于代理安装的 `mib` 子目录中的 `empire.asn1` 文件。

monIndex

为该条目定义唯一的行索引（11 到 MAXINT）。保留行 1 到 10 以供代理内部使用。

权限：只读

monDesc

定义被监控对象的说明（长度为 0 到 512 个字符）。

权限：读写

默认：默认条目：

monInterval

定义代理轮询（监控）变量的频率（以秒计）。该值必须至少为 30，且是 30 秒的倍数。

权限：读写

默认值：60

monSampleType

定义代理是对对象的绝对值 (**absoluteValue(1)**) 进行抽样, 还是接受连续样本 (**deltaValue(2)**) 之间的差异。例如, 使用 **deltaValue** 监控计数器变量, 因为它说明更改的速度。使用 **absoluteValue** 来监控标尺, 因为它说明对象的精确值。

权限: 读写

默认: **absoluteValue(1)**

monOID

定义表示受监控的 MIB 对象的完整对象标识符。需要对象标识符的实例部分 (例如, 标量 0)。必须存在对象实例, 并且其必须包含在 SystemEDGE 代理支持的 MIB 中。对象应当是整数类型 (包括计数器、标尺、时间刻度)。

权限: 读写

monCurrVal

指定上次为被监控的 MIB 变量抽样的值。每 **monInterval** 秒, 代理更新该字段来反映变量的最新值。

权限: 只读

monOperator

定义用于对表达式 *currval operator value* 进行求值的布尔运算符。运算符可以为以下值之一:

- **nop(1)** (无运算; 检索对象的值, 但不对布尔表达式进行求值)
- **gt(2)** (大于)
- **lt(3)** (小于)
- **ge(4)** (大于或等于)
- **le(5)** (小于或等于)
- **eq(6)** (等于)
- **ne(7)** (不等于)

权限: 读写

默认: **nop(1)**

monValue

定义整数值，代理在每次监控周期内使用受监控的 MIB 变量的当前值 **monOperator** 与该值进行比较。如果比较的值为 **true**，则代理更改监视器状态（针对严重性为无以外的自主监视器），或代理发送阈值违反陷阱（针对严重性为无的自主监视器）。

例如，要在标尺的值超过 100 时接到通知，请将 **monValue** 设置为 100。

权限：读写

monLastCall

定义代理上次对正在监控的 MIB 变量进行抽样的时间（基于 **sysUpTime**）。0 表示 MIB 变量尚未被抽样。

权限：只读

monNumTraps

定义代理已为该条目发送的陷阱数。该值跟踪当重要级别为无的自主监视器（传统监视器）中发生阈值违反时代理发送的 **monitorTrap**。该值可用于确定传统监视器发生异常情况的频率。它还提供检测错过的陷阱消息的方法。

权限：只读

注意：有关为有状态自主监视器的受管对象发送状态更改陷阱数目的信息，请参阅聚合状态表。

monLastTrap

定义代理针对该条目最后发送陷阱的时间（基于 **ysUpTime**）。该值跟踪当重要级别为无的自主监视器中发生阈值违反时代理发送的 **monitorTrap**。

权限：只读

注意：有关为有状态自主监视器的受管对象发送最后一次状态更改陷阱的信息，请参阅聚合状态表。

monRowStatus

定义行状态，行状态可以是以下状态之一：

- 活动
- notInService
- notReady
- createAndGo
- createAndWait
- destroy(6)

通常，行处于 **active** 或 **notInService** 状态。这些值的含义与由 **SNMPv2 SMI RowStatus** 文本约定定义的值的含义相同。

权限：读写

默认：createAndWait(5)

monMinValue

定义自代理轮询 MIB 变量以来，已经观察到的最低（最小）值。

权限：只读

monMaxValue

指定自代理轮询 MIB 变量以来，已经观察到的最高（最大）值。

权限：只读

monAction

定义当表达式的值为 **True** 并且代理发送阈值违反陷阱（传统监视器）或状态更改陷阱（有状态监视器）时，将运行的具有任何参数的引用的命令（长度为 0 到 2048 个字符）。对于有状态自主监视器，如果阈值违反不导致对象状态更改，则命令无法运行。如果字符串为空，则代理不为该条目执行操作。

注意：请勿使用 Windows 批处理文件进行操作；它们施加严格的编程限制，并且经常错误地使用桌面应用程序。应使用更强大且灵活的脚本语言，如 Perl 或 Visual Basic。

权限：读写

monFlags

定义整数标志，这些标志表示条目在操作过程中遵循的其他行为语义。有关可用的标志和设置标志的详细信息，请参阅[监视器表标志](#) (p. 199)。

权限：读写

monSupersededBy

指定取代该条目的监视器的索引。取代的监视器在该表条目之前被评估，如果超过取代的监视器的阈值，则不再评估该条目。通过关联监视器的这种顺序，可以忽略陷阱的无用顺序。值零 (0) 表示未进行任何监视器条目关联。

注意：使用对象状态模型会废弃 (并禁用) 该功能。

权限：读写

monObjClass

定义用于对象状态管理模型的受监控 MIB 对象的对象类。代理使用 `sysedge.oid` 文件来分配默认值。类名可用于指定被监控对象表。如果 `SystemEDGE` 未确定具有指定名称的现有表，`SystemEDGE` 会创建类别。该类别可以指定监控对象标识符 (SNMP OID)，也可出于聚合目的而被引用。

重要信息！只有当监控条目解析为现有 SNMP OID 时，新建的引用才有效。

monObjInstance

定义用于对象状态管理模型的受监控 MIB 对象的对象实例。如果不配置该值并且代理无法确定一个默认值，则该值为“未知”。非表对象的默认值为“空”。对于远程对象的无缝监视，实例名以前缀 `//hostname/` 开始。

对于本地对象，代理则会在每个实例名前面添加前缀 `//.` (包括“null”)。该属性不能设置为空字符串或“*”。任何此类请求都将被忽略。如果在解析时实例名无法与相应实例相匹配，`SystemEDGE` 将使用指定的实例名创建对此监控条目的 OID 的引用。

重要信息！只有当监控条目解析为现有 SNMP OID 时，新建的引用才有效。

monObjAttribute

定义用于对象状态管理模型的受监控 MIB 对象的对象实例。代理使用 `sysedge.oid` 文件为与 MIB 属性名称相应的对象分配默认值。如果解析监控条目期间属性名称未知，`SystemEDGE` 将使用指定的名称创建属性。

权限：读写

默认：受监控的 OID 的 ASCII MIB 属性名称

重要信息！只有当监控条目解析为现有 SNMP OID 时，新建的引用才有效。

monSeverity

定义将用于该对象状态管理模型的重要级别。该值定义发生阈值违反时将分配给条目的状态。有效值如下：

- 无(1)
- 正常(2)
- 警告(3)
- 轻微(4)
- 重大(5)
- 严重(6)
- 致命(7)

值“无”创建传统监视器，它将条目排除在对象状态管理模型之外。

权限：读写

默认值：无

monCurrState

定义条目的当前派生的状态，由重要级别和当前违反条件评估，同时考虑 **monFlags** 中定义的任何延迟值。有效值如下：

- 未知(1)
- 正常(2)
- 警告(3)
- 轻微(4)
- 重大(5)
- 严重(6)
- 致命(7)
- 运行(11)
- 关闭(12)

仅为传统监视器显示值“运行”和“关闭”（重要级别为“无”）。只有当该状态是“聚合状态”表中相应受管对象的所有监视器中最差状态时，该列中的状态更改才导致状态更改陷阱。

权限：只读

示例自主监视器表条目

本节提供“自主监视器”表中条目的示例，指示 SystemEDGE 代理监控目标系统的 1 分钟平均负载（为易于查看而分离表）：

索引	Desc	间隔	抽样类型	OID	CurrVal
21	“1 分钟 平均负 载”	60	绝对值	loadAvg 1Min.0	320
Oper	值	最后调 用	Num 陷阱	最后陷阱	行状态
gt	300	Od 22:40:00	{0}	{0}	活动
最小值	最大值	操作	标志	取代对象	对象类
1	3	""	0x0	{0}	系统
对象实例		对象属性		重要级别	当前状态
加载		LoadAverage1Min		警告	警告

该条目提供以下信息：

- 这是表中第 21 行。
- 其目的是监控系统的 1 分钟平均负载。
- 每隔 60 秒，代理检查 loadAvg1Min MIB 变量的绝对值（当前值是 320）以查看是否大于 300（指定的阈值）。如果是，则代理将当前状态更改为警告（由重要级别值定义），并且如果该监视器表示聚合状态表中相应受管对象的最差状态，则发送状态更改陷阱。
- 随着每次轮询，代理使用它刚检索的值更新 CurrVal 列，在 LastCall 列中记录时间，并且如果合适（即，如果刚轮询的值是迄今观察到的最低或最高值），则更新 MinValue 和 MaxValue 列。
- RowStatus 列显示该条目处于活动状态。在该示例中，操作为空，因此发送陷阱时，将不会运行命令。
- 表示默认自主监视器表行为的“标志”列设置为零。
- 类、实例和属性值创建该自主监视器条目，作为“整合状态”表中具有这些值的受管对象的一部分。您可以定义具有这些相同值的其他自主监视器条目以监控不同的重要级别，或者您可以将该条目与其他相关的对象聚合。

行创建

自主监视器表的 **monIndex** 列充当关键字段(或行索引)以区分表中的行。保留 1 到 10 行以供 **SystemEDGE** 代理内部使用。您可以配置范围 11 到 **MAXINT** 的行。

您可以使用具有自主监视器表的下列 **MIB** 对象来优化行创建:

monUnusedIndex

在对象上执行 **SNMP Get** 时, 返回自主监视器表的未使用的索引号。查询该对象来获取可用于创建行的索引号。

monMatchDescr

确定与 **monDescr** 属性中特定的条目说明对应的索引号。在具有 **monDescr** 属性值的该 **MIB** 对象上执行 **SNMP Set**, 从而导致代理在“自主监视器”表中搜索条目, 并且将最后一个匹配条目的索引值放入 **monMatchIndex** **MIB** 对象。

monMatchIndex

用于 **monMatchDescr** 时将特定的条目说明与其索引号相对应。

按照本地策略, 您可以选择为系统管理预留行块。然后可以在保留的行块中定义条目, 而无需担心行可能已经被其他用户条目使用。按照本地的策略, 当所有其他用户定义用户配置的条目时, 应使用在保留范围以外的行索引。

通过保留行块, 您可以定义要跨所有计算机监控的一组一致的条件(行条目), 以便同样的条件定义在每台计算机的相同行数中。例如, 可以使用第 31 行(**monIndex** = 31)来定义监控 **swapCapacity** 变量的条目, 并且可以将该配置分配给每个系统, 以便第 31 行监控每个系统上的 **swapCapacity** 变量。索引 11-16 用于预定义的监视器。远程监控 **AIM** 使用大于 8000 的索引号。

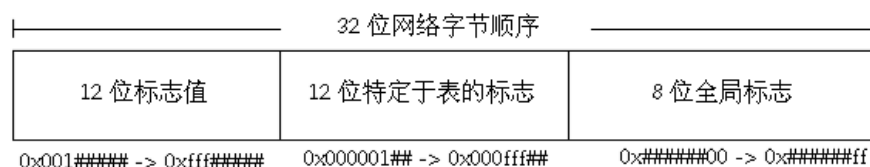
在 **CA Virtual Assurance** 上配置代理可以自动执行该过程。您可以使用特定行号配置条目块, 并将该配置部署到一个操作中的所有受管系统。

自主监视器表标志

自主监视器表中的 **monFlags** 列是 32 位无符号整数，这些无符号整数可以为相应的自主监视器表行指定其他行为语义。默认情况下，自主监视器表行执行以下内容：

- 监控系统资源
- 评估状态
- 发送 SNMP 陷阱（仅限传统监视器）
- 将陷阱记录到 **sysedge.log** 文件
- 调用操作（如果进行了配置）

可以设置标志位来更改这些默认值。**SystemEDGE** 代理以十六进制（基数为 16）表示法解释所有标志。下图显示了自主监视器表标志字段 (**monFlags**) 的构成。



标志值由三个字段组成：

- 字段 1 包含为 **Systems Management Empire MIB** 的监控表定义的常用表标志。该部分是标志的低 8 位。
- 字段 2 包含分别为每个监控表定义的表特定的标志。该字段定义常用表标志后面接下来的 12 个低位。
- 字段 3 包含为整数值保留的 12 个高位，以便用于表特定的标志。

以下部分解释了每个标志位。您可以通过逻辑 **OR** 运算组合这些标志值。

以下列表介绍了自主监视器表标志：

0x00000001

禁止执行该条目的操作。

0x00000002

禁止发送该条目的 **SNMP** 陷阱。

注意：自主监视器表仅控制为传统监视器发送陷阱。对于有状态自主监视器，合计状态表仍可能发生状态更改陷阱。

0x00000004

禁止尝试重新初始化该条目。默认情况下，代理通过尝试查询正在监控的 MIB 对象，来定期地试图重新初始化该条目。如果查询失败，则进行重新初始化。

0x00000008

禁止在 `sysedge.log` 文件中记录该条目的陷阱。设置该位对发送陷阱没有影响。在事件频繁发生或将特定条目用作代理监控信号时，禁用事件记录会很有用。

0x00000010

每次代理尝试重新初始化监控并且无法查询 MIB 对象时，对该条目连续发送 `monitorEntryNotReady` 陷阱。当代理监控的 MIB 对象不再存在，甚至随后尝试重新初始化该条目仍失败时，代理的默认行为是发送单个 `monitorEntryNotReady` 陷阱。如果启用该功能，则每次重新初始化失败时，代理都会发送一个额外的 `monitorEntryNotReady` 陷阱。

0x00000020

禁止将默认参数传递给操作脚本或程序。`SystemEDGE` 通常传递默认操作参数（表示陷阱类型、说明字段等）。有关操作参数的详细信息，请参阅[自主监视器表操作参数](#) (p. 201)。

0x00000040

禁止发送该条目的 `notReady` 陷阱。这包括不记录以及不执行 `notReady` 陷阱的操作。

0x00000100

当自主监视器表达式从 `True` 转变为 `False` 时，为该条目发送 `monitorClear` 陷阱。这包括记录并执行 `monitorClear` 陷阱的操作。

0xXXX00200

只有在连续 X 次违反时发送监视器阈值陷阱或更改当前状态。连续 X 次违反后，代理会为每个后续的 `True` 表达式计算发送监视器陷阱。如果阈值表达式从 `True` 转变为 `False`，则代理将从零开始计算后续违反。该标志也适用于操作执行和日志记录。您可以通过标志值字段指定 X 的值。有关该行为的示例，请参阅[自主监控示例](#) (p. 209)。

0xXXX00400

发送最多 X 个连续监视器陷阱，然后不再发送。如果启用该功能，则会设置当超过阈值时可发生的连续监视器陷阱和操作执行数目的上限。阈值表达式从 `True` 转变为 `False` 之后，代理将从零开始计算后续违反。该标志也适用于操作执行和日志记录。您可以通过标志值字段指定 X 的值。有关该行为的示例，请参阅[自主监控示例](#) (p. 209)。

0x00000800

聚合相同对象信息（类、实例、属性）以及相同重要级别的监视器的状态，使其具有 AND 关系。设置有该标志且具有相同重要级别的所有连接的监视器必须具有违反条件，才能相应地更改聚合状态。

0x###00000

多个标志位使用值 *X* 来发送陷阱并执行操作和日志。值 *X* 被指定为高阶 12 位标志字段。利用该字段的标志位是互斥的。

自主监视器表操作参数

SystemEDGE 代理可在操作命令被调用时向其提供若干默认参数。这些参数是对您在操作字符串中指定的任何参数的补充，并在您指定的那些参数之后通过命令行传递。默认操作参数与代理针对自主监视器表发送的 SNMP 陷阱中提供的参数一样。

注意：通过在表条目中设置 0x00000020 标志，可以防止代理在命令中包含这些操作参数。

以下列表介绍自主监视器表操作的默认参数：

trapType

定义所发送陷阱的类型，例如 ntEventMonNotReadyTrap 或 ntEventMonMatchTrap。

monDescr

定义表条目的说明。

monOID

定义表条目的 OID。

monCurrVal

定义表条目的当前值。

monValue

定义表条目的当前被监控的值。

monRowStatus

定义表条目的当前行状态。

monOperator

定义表条目的当前使用的运算符。传递给操作脚本的值是实际运算符的数字表示。例如，如果运算符是 >，则传递给操作脚本的值是 2。

monIndex

定义表条目的索引。

monFlags

定义与表条目关联的标志。传递给操作脚本的值采用以 16 为基数（十六进制）的表示法，前面的 0x 表示它是十六进制数（例如，0x00000020）。

monObjClass

定义用于对象状态管理模型的受监控 MIB 对象的对象类。代理使用 sysedge.oid 文件来分配默认值。类名可用于指定被监控对象表。如果 SystemEDGE 未确定具有指定名称的现有表，SystemEDGE 会创建类别。该类别可以指定监控对象标识符 (SNMP OID)，也可出于聚合目的而被引用。

重要信息！ 只有当监控条目解析为现有 SNMP OID 时，新建的引用才有效。

monObjInstance

定义用于对象状态管理模型的受监控 MIB 对象的对象实例。如果不配置该值并且代理无法确定一个默认值，则该值为“未知”。非表对象的默认值为“空”。对于远程对象的无缝监视，实例名以前缀 //hostname/ 开始。

对于本地对象，代理则会在每个实例名前面添加前缀 //./（包括“null”）。该属性不能设置为空字符串或“*”。任何此类请求都将被忽略。如果在解析时实例名无法与相应实例相匹配，SystemEDGE 将使用指定的实例名创建对此监控条目的 OID 的引用。

重要信息！ 只有当监控条目解析为现有 SNMP OID 时，新建的引用才有效。

monObjAttribute

定义用于对象状态管理模型的受监控 MIB 对象的对象实例。代理使用 sysedge.oid 文件为与 MIB 属性名称相应的对象分配默认值。如果解析监控条目期间属性名称未知，SystemEDGE 将使用指定的名称创建属性。

权限： 读写

默认： 受监控的 OID 的 ASCII MIB 属性名称

重要信息！ 只有当监控条目解析为现有 SNMP OID 时，新建的引用才有效。

monCurrState

定义表条目的当前状态。

查看自主监视器表

您可以使用 CA Virtual Assurance 来查看受管代理的自主监视器表的内容。

在 CA Virtual Assurance 中查看自主监视器表

1. 访问 CA Virtual Assurance 用户界面。
2. 单击“资源”选项卡。
此时将显示“受管资源”窗格和“数据中心”页面。
3. 在“受管资源”窗格中展开“受管”，然后选择代理驻留的服务器。
此时将显示包含服务器详细信息的页面。
4. 单击“配置”，然后单击“自主监视器”选项卡。
此时将出现自主监视器表，其中包含选定服务器上的所有现有的自主监视器。

自主监控配置

您可以控制 SystemEDGE 代理监控的 MIB 变量和条件，方法是在自主监视器表中添加、删除或修改相应的条目。

可以按下列方式配置自主监视器表：

动态使用 CA Virtual Assurance 中基于文件的配置

CA Virtual Assurance 可以配置安装在它从 CA Virtual Assurance 用户界面管理的任何系统上的代理。建议使用该配置方法，因为该方法可以利用一种用户友好且安全的机制动态地定义和传送监视器，该机制不需要代理重新启动。您可以为单个代理配置监控，也可以定义监控策略以在多个受管系统上应用。当您执行单个配置更改或应用配置策略时，CA Virtual Assurance 会将新的配置文件传送给代理系统。CA Virtual Assurance 可以在整个数据中心中充当代理配置的中心点，您可以覆盖或拒绝通过其他方式执行的更改。

有关使用 CA Virtual Assurance 配置代理的详细信息，请参阅 CA Virtual Assurance 文档。

从管理系统（例如 eHealth AdvantEDGE View、CA Spectrum 或 CA NSM）动态使用 SNMP Set 命令

各种管理系统均支持通过 SNMP Set 命令将监视器添加到 SystemEDGE 中。eHealth 通过 AdvantEDGE View 接口提供该功能，CA NSM 通过 Agent View 启用配置，CA Spectrum 通过 OneClick 启用配置。通过 CA Virtual Assurance 进行动态配置所提供的所有功能都无法通过这些方法获得。

有关 SNMP Set 配置的详细信息，请参阅相应管理系统的文档。

通过在本本地 sysedge.cf 文件中指定自主监视器表的条目来执行手动配置

手动配置要求您使用特定指令将条目添加到本地 sysedge.cf 文件中。当从 sysedge.cf 文件直接配置代理时，必须执行完全重新启动才能使更改生效。请使用位于代理数据目录中的 sysedge.cf 文件进行直接配置。

本节介绍如何在本地 sysedge.cf 文件中直接配置自主监控，但介绍的属性是所有形式的配置所必需的。

monitor 指令—将条目添加到自主监视器表

monitor 指令可用于通过 `sysedge.cf` 文件将自主监视器条目直接添加到自主监视器表。下列参数表示自主监视器表中的各列。

使用如下所述的语法向代理数据目录中的 `sysedge.cf` 文件添加一行，保存文件，然后重新启动代理以使更改生效。

使用 **monitor** 指令将条目添加到自主监视器表，如下所示：

```
(1) monitor oid oid index flags interval stype oper thresh ['descr' ['action' [supIndex]]]
```

```
(2) monitor oid oid index flags interval stype oper thresh ['descr' ['action' ['objClass' 'objInst' 'objAttr' severity]]]
```

```
(3) monitor table inst attr index flags interval stype oper thresh ['descr' ['action' [supIndex]]]
```

```
(4) monitor table inst attr index flags interval stype oper thresh ['descr' ['action' ['objClass' 'objInst' 'objAttr' severity]]]
```

oid

指定要监控的 **OID**。您可以使用完整的点分十进制值（例如，1.3.6.1.2.1.25.1.5.0）或符号 **MIB** 名称（例如，hrSystemNumUsers.0）来指定 **OID**。不管使用哪一种方式，您都必须指定对象实例，对象实例对于非表格 **MIB** 变量通常为零，或是对应于属于 **MIB** 表的某一属性的特定实例。

示例：

- 1.3.6.2.1.25.1.5.0
- hrSystemNumUsers.0

table 'instance' attribute

指定要监控的对象的 **tableEntry**、实例和属性名称。如果属性没有实例，则输入 " 作为实例属性。

示例：

- processGroupMonEntry 'iexplore|IEXPLORE' pgmonRSS
- devTableEntry 'C:' devCapacity

有关详细信息，请参阅[反向 OID 查找](#) (p. 208)。

index

指定要用于该条目的监视器表的行（索引）。表中各行由索引号唯一标识。保留行 1 至 10 以供代理内部使用，因此索引值必须大于 10 且在表中唯一。

flags

使用十六进制标志值（例如，0x00000001）指定该条目的任何其他操作说明。有关可用标志的详细信息，请参阅[自主监视器表标志](#) (p. 199)。

interval

指定应进行监控的时间间隔（秒）。例如，如果值为 30，则指示代理每隔 30 秒对该条目进行监控。该值必须至少为 30，且是 30 秒的倍数。

stype

指定代理应对属性的绝对值 (*absoluteValue*) 进行抽样，还是采用连续抽样之间的差 (*deltaValue*)。使用 *deltaValue* 可监控计数器变量的变化速率，使用 *absoluteValue* 可监控计量器的精确值。

oper

指定根据所提供的阈值评估当前条目值的布尔运算符。

运算符的有效值如下：

- nop（无运算）
- >（大于）
- <（小于）
- >=（大于或等于）
- <=（小于或等于）
- =（等于）
- !=（不等于）

默认值： nop

thresh

使用特定运算符指定要与所监控属性的当前值比较的整数阈值。

'descr'

指定表条目的任意说明（长度为 0 至 512 个字符）。

'action'

指定当满足阈值条件且代理发送阈值违反陷阱（传统监视器）或状态更改陷阱（状态监视器）时要运行的命令（长度为 0 到 2048 个字符），包括完整路径和任何参数。对于有状态自主监视器，如果阈值违反不导致对象状态更改，则命令无法运行。如果字符串为空，则代理不为该条目执行操作。

注意：当代理执行操作时，您可以更改默认设置。有关详细信息，请参阅“代理配置”一章。

supIndex

指定充当该条目的父伙伴或覆盖伙伴的表索引。有效值是现有的或将来的 SystemEDGE 监视器索引（正整数值）。默认值是零 (0)，表示默认行为。

请注意，对象状态模型（输入无以外的重要级别）的使用已过时，请禁用该功能。

'objClass'

定义用于对象状态管理模型的受监控 MIB 对象的对象类。代理使用 sysedge.oid 文件来分配默认值。类名可用于指定被监控对象表。如果 SystemEDGE 未确定具有指定名称的现有表，SystemEDGE 会创建类别。该类别可以指定监控对象标识符 (SNMP OID)，也可出于聚合目的而被引用。

重要信息！只有当监控条目解析为现有 SNMP OID 时，新建的引用才有效。

'objInst'

指定用于状态管理模型的受监控对象的对象实例。定义该属性（以及 objClass 和 objAttr）的值，以创建一个聚合具有相同值的所有条目的受管对象。代理会使用 sysedge.oid 文件基于特定属性值为表格对象分配默认值，并分配 null 作为非表格对象的默认值。

'objAttr'

指定用于状态管理模型的对象属性。定义该属性（以及 objClass 和 objInst）的值，以创建一个聚合具有相同值的所有条目的受管对象。代理会使用 sysedge.oid 文件来分配对应于 MIB 属性名称的默认值。

默认：MIB 属性名称

重要级别

指定要用于对象状态管理模型的重要级别。该值定义发生阈值违反时将分配给条目的状态。有效值如下：

- 无
- 正常
- 警告
- 轻微
- 重大
- 严重
- 致命

值“无”创建传统监视器，它将条目排除在对象状态管理模型之外。

默认值：无

反向 OID 查找

位于代理安装的 `config` 目录中的 `sysedge.oid` 文件负责维护每个支持的 SystemEDGE MIB 属性的类、实例和属性的默认值。该文件将使用以下约定来分配类、实例和属性的默认值：

类

使用属性所属的 MIB 组或表的名称。

示例：

- `kernelConfig`
- `userEntry`

实例

对于不属于 MIB 表的属性，使用值“`null`”。对于表格对象，代理会基于特定属性值或 MIB 表中属性的值获取默认实例值。

示例：

- `devMntPt` 属性的值（设备表）
- `processName` 属性的值（进程表）

属性

使用 MIB 属性名称。

示例:

- devDevice
- maxProcs

有关反向 OID 查找的示例，请参阅[自主监控示例](#) (p. 209)。

自主监控示例

本节为自主监视器表提供了示例条目，以监控目标系统上的阈值。每个示例都介绍如何定义条目和说明被监控的条件。您可以将这些条目添加到 `sysedge.cf` 中。

示例：监控一分钟内的平均负载

以下示例配置代理以监控系统的一分钟内的平均负载：

```
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' '' 'kernelperf' 'null' 'loadAverage1Min' warning
```

1.3.6.1.4.1.546.1.1.7.8.26.0

对应于 Systems Management Empire MIB 中包含的
`loadAverage1Minute` 变量的 OID。

11

指示该条目将占用监视器表的第 11 行 (`monIndex=11`)。

60

指定应每隔 60 秒对平均负载采样一次。

绝对

表示代理应使用对象的值，而不是连续抽样之间的差。

300

指示与当前平均负载相比较的值。如果当前采样值大于 (>) 300，则会发生一个事件。

注意：代理返回的平均负载为基础系统的平均负载乘以 100 得到的值。例如，如果平均负载为 3，则将返回 300。

kernelperf

指示该条目将 `kernelperf` 对象类用于对象状态模型。该对象类将替换为默认值。在此，“*”是无效值。

空

指示该条目将空对象实例用于对象状态模型。该对象实例将替换为默认值。在此，“*”是无效值。

注意：“//”和“/”可以用作分隔符以将实例与系统相关联：

“//hostname/instance”。对于本地系统实例，始终前置“//.”以将实例与本地系统相关联。“//hostname/*”、“//*/instance”和“//*/*”均为无效值。

loadAverage1Min

指示该条目将 loadAverage1Min 对象属性用于对象状态模型。该对象类将替换为默认值。在此，“*”是无效值。

警告

指示该条目将警告重要级别用于对象状态模型。

示例：监控五分钟内的平均负载

以下示例配置代理以监控系统的五分钟内的平均负载：

```
monitor oid loadAverage5Min.0 12 0x00500300 300 absolute > 200 'Monitor 5 minute  
load average' '' 'kernelperf' 'null' 'loadAverage5min' fatal
```

loadAverage5Min.0

对应于 Systems Management Empire MIB 中包含的 loadAverage5Minute 变量的 OID。该条目将占用监视器表的第 12 行 (monIndex=12)。

0x00500300

- 配置代理，以便当阈值表达式从 True 转变为 False 时发送 monitorClear 陷阱。
- 配置代理，以便仅在连续第 X 次发生该事件时开始发送陷阱并运行操作（如果已配置）。

指定 X 等于 5。因此，代理在第五次出现该事件时开始发送陷阱。

300

指定应每隔 300 秒对平均负载采样一次。

绝对

表示代理应使用对象的值，而不是连续抽样之间的差。

200

指示与当前平均负载相比较的值。如果当前采样值大于 (>) 200，则代理将向所有已配置的管理器发送陷阱。

“

指示未指定任何操作。

kernelperf

指示该条目将 kernelperf 对象类用于对象状态模型。该对象类将替换为默认值。在此，“*”是无效值。

空

指示该条目将空对象实例用于对象状态模型。该对象实例将替换为默认值。在此，“*”是无效值。

注意：“//”和“/”可以用作分隔符以将实例与系统相关联：

“//hostname/instance”。对于本地系统实例，始终前置“//.”以将实例与本地系统相关联。“//hostname/*”、“//*/instance”和“//*/*”均为无效值。

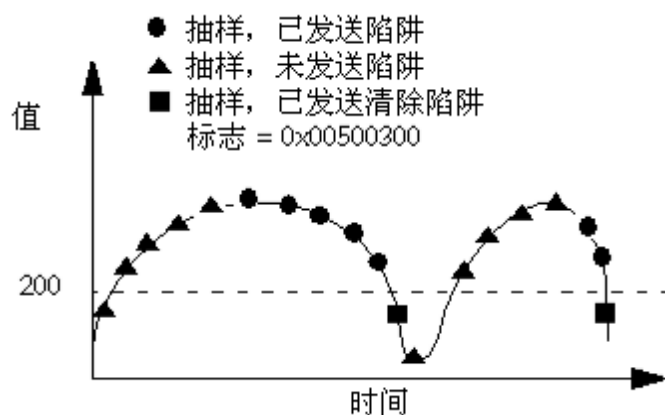
loadAverage5Min

指示该条目将 loadAverage5Min 对象属性用于对象状态模型。该对象类将替换为默认值。在此，“*”是无效值。

致命

指示该条目将致命重要级别用于对象状态模型。

下图显示了代理如何基于该 monitor 指令发送和清除陷阱：



示例：监控十五分钟内的平均负载

以下示例配置代理以监控系统的十五分钟内的平均负载：

```
monitor oid loadAverage15Min.0 13 0x0 900 absolute > 200 'Monitor 15 minute load average' '' '' 'kernelperf' 'null' 'loadAverage15Min' critical
```

loadAverage15Min.0

对应于 Systems Management Empire MIB 中包含的 loadAverage15Min 变量的 OID。该条目将占用监视器表的第 13 行 (monIndex=13)。

900

指定应每隔 900 秒对平均负载采样一次。

绝对

表示代理应使用对象的值，而不是连续抽样之间的差。

200

指示与当前平均负载相比较的值。如果当前采样值大于 (>) 200，则代理将向所有已配置的管理器发送陷阱。

kernelperf

指示该条目将 kernelperf 对象类用于对象状态模型。该对象类将替换为默认值。在此，“*”是无效值。

空

指示该条目将空对象实例用于对象状态模型。该对象实例将替换为默认值。在此，“*”是无效值。

注意：“//”和“/”可以用作分隔符以将实例与系统相关联：

“//hostname/instance”。对于本地系统实例，始终前置“//.”以将实例与本地系统相关联。“//hostname/*”、“//*/instance”和“//*/*”均为无效值。

loadAverage15Min

指示该条目将 loadAverage15Min 对象属性用于对象状态模型。该对象类将替换为默认值。在此，“*”是无效值。

严重

指示该条目将严重重要级别用于对象状态模型。

示例：监控系统的中断率

以下示例配置代理以监控本地系统上硬件中断的发生率：

```
monitor oid numInterrupts.0 1003 0x00500400 60 delta > 1000 'Monitor Interrupt Rate' '' 'kernelperf' 'null' 'numInterrupts' warning
```

numInterrupts.0

对应于 Systems Management Empire MIB 中包含的 numInterrupts 计数器对象的 OID。该条目是监视器表中的索引 14。

0x00500400

- 配置代理以在该监视器表达式为 True 时最多发送 X 个连续陷阱。
- 指定 X 等于 5。因此，代理会发送 5 个连续 monitorEvent 陷阱，然后停止发送陷阱，直到该条目重置自身。该条目会在表达式从 True 转变为 False 时重置自身。

不指定代理应发送 monitorClear 陷阱；因此，当表达式从 True 转变为 False 时，不会发送 monitorClear 陷阱。

60

指示应每隔 60 秒对中断率采样一次。

增量

告诉代理测量中断次数发生改变的速率。因为该对象是计数器，所以采用增量采样类型比较合适。

1000

指示与当前中断次数相比较的值。

kernelperf

指示该条目将 kernelperf 对象类用于对象状态模型。该对象类将替换为默认值。在此，“*”是无效值。

空

指示该条目将空对象实例用于对象状态模型。该对象实例将替换为默认值。在此，“*”是无效值。

注意：“//”和“/”可以用作分隔符以将实例与系统相关联：

“//hostname/instance”。对于本地系统实例，始终前置“//.”以将实例与本地系统相关联。“//hostname/*”、“//*/instance”和“//*/*”均为无效值。

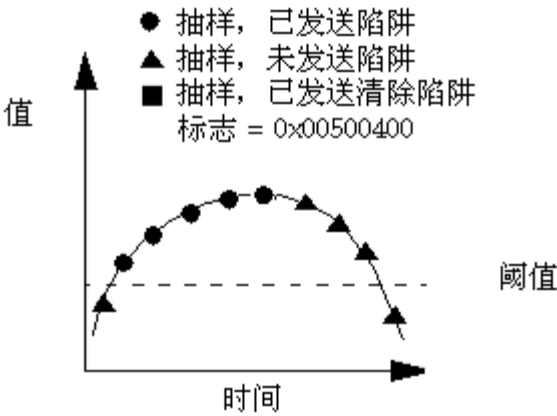
numInterrupts

指示该条目将 numInterrupts 对象属性用于对象状态模型。该对象类将替换为默认值。在此，“*”是无效值。

警告

指示该条目将警告重要级别用于对象状态模型。

下图显示了代理如何基于该 monitor 指令发送和清除陷阱。



示例：监控系统的页面故障率

以下示例配置代理以监控本地系统上硬件页面中断的发生率：

```
monitor oid numPageFaults.0 15 0x00500500 60 delta > 1000 'Monitor Page-fault Rate' '' 'kernelperf' 'null' 'numPageFaults' major
```

numPageFaults.0

对应于 Systems Management Empire MIB 中包含的 numPageFaults 计数器对象的 OID。该条目是监视器表中的索引 15。

0x00500500

- 配置代理，以便当表达式从 True 转变为 False 时发送 monitorClear 陷阱。
- 指定代理应最多发送 X 个连续 monitorEvent 陷阱，然后停止发送，直到该条目重置自身。
- 指定 X 等于 5。因此，代理会发送 5 个连续 monitorEvent 陷阱，然后停止发送陷阱，直到该条目重置自身。

60

指示应每隔 60 秒对中断率采样一次。

增量

告诉代理测量中断次数发生改变的速率。因为该对象是计数器，所以该条目适合采用增量采样类型。

1000

指示与当前中断次数相比较的值。

kernelperf

指示该条目将 **kernelperf** 对象类用于对象状态模型。该对象类将替换为默认值。在此，“*”是无效值。

空

指示该条目将空对象实例用于对象状态模型。该对象实例将替换为默认值。在此，“*”是无效值。

注意：“//”和“/”可以用作分隔符以将实例与系统相关联：

“//hostname/instance”。对于本地系统实例，始终前置“//.”以将实例与本地系统相关联。“//hostname/*”、“//*/instance”和“//*/*”均为无效值。

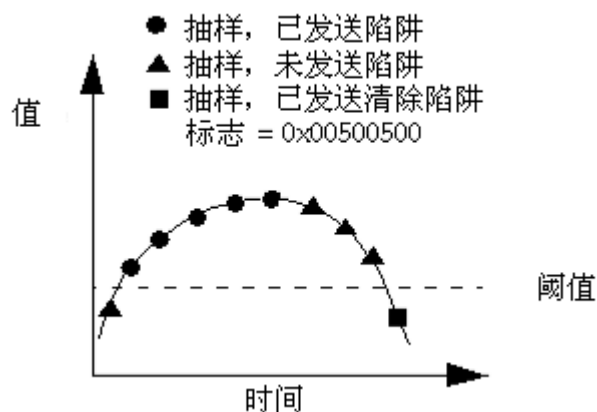
numPageFaults

指示该条目将 **numPageFaults** 对象属性用于对象状态模型。该对象类将替换为默认值。在此，“*”是无效值。

重大

指示该条目将重大重要级别用于对象状态模型。

下图显示了代理如何基于该 **monitor** 指令发送和清除陷阱：



示例：监控接口上的传入数据包数量

以下示例配置代理以监控第一个以太网接口（对于 Sun 系统为 `le0`）接收数据包的速率：

```
monitor oid ifInUcastPkts.2 1004 0x0 60 delta > 1000 'Monitor le0 Incoming Packets'  
' 'ifEntry' 'WAN MiniPort' 'ifInUcastPkts' minor
```

ifInUcastPkts.2

指示要采样的特定 MIB 对象实例。

16

指示该条目是监视器表中的索引 16。

60

指示代理应每隔 60 秒计算一次速率。

增量

告诉代理测量传入数据包数量发生改变的速率。

1000

指定在比较中要使用的值。如果变化率大于 (>) 1000，则代理发送陷阱。

ifEntry

指示该条目将 `ifEntry` 对象类用于对象状态模型。该对象类将替换为默认值。在此，“*”是无效值。

WAN MiniPort

指示该条目将 `WAN MiniPort` 对象实例用于对象状态模型。该对象实例将替换为默认值。在此，“*”是无效值。

注意：“//”和“/”可以用作分隔符以将实例与系统相关联：

“//hostname/instance”。对于本地系统实例，始终前置“//.”以将实例与本地系统相关联。“//hostname/*”、“//*/instance”和“//*/*”均为无效值。

ifInUcastPkts

指示该条目将 `ifInUcastPkts` 对象属性用于对象状态模型。该对象类将替换为默认值。在此，“*”是无效值。

轻微

指示该条目将轻微重要级别用于对象状态模型。

示例：监控接口上的传出数据包数量

以下示例配置代理以监控第一个以太网接口（对于 Sun 系统为 le0）传输数据包的速率。

```
monitor oid ifOutUcastPkts.2 1005 0x0 60 delta > 1000 'Monitor le0 Outgoing
Packets' '' 'ifEntry' 'WAN MiniPort' 'ifOutUcastPkts' ' ok
```

ifOutUcastPkts.2

指示要采样的特定 MIB 对象实例。

60

指示代理应每隔 60 秒计算一次速率。

增量

指示采样类型，因为被监控对象是 MIB-II ifEntry 计数器。

1000

指定在比较中要使用的值。如果变化率大于 (>) 1000，则代理将向所有已配置的管理器发送陷阱消息。

ifEntry

指示该条目将 ifEntry 对象类用于对象状态模型。该对象类将替换为默认值。在此，“*”是无效值。

WAN MiniPort

指示该条目将 WAN MiniPort 对象实例用于对象状态模型。该对象实例将替换为默认值。在此，“*”是无效值。

注意：“//”和“/”可以用作分隔符以将实例与系统相关联：

“//hostname/instance”。对于本地系统实例，始终前置“//.”以将实例与本地系统相关联。“//hostname/*”、“//*/instance”和“//*/*”均为无效值。

ifOutUcastPkts

指示该条目将 ifOutUcastPkts 对象属性用于对象状态模型。该对象类将替换为默认值。在此，“*”是无效值。

ok

指示该条目将正常重要级别用于对象状态模型。

示例：监控接收的 SNMP 数据包数量

以下示例配置代理以监控代理接收 SNMP 请求的速率：

```
monitor oid snmpInPkts.0 18 0x0 30 delta > 4000 'Monitor SNMP Packets' '' 'snmp'  
'null' 'snmpInPkts' minor
```

snmpInPkts.0

指示要采样的 MIB II 对象实例。

18

指定该条目是监视器表中的索引 18。

30

指示代理应每隔 30 秒计算一次速率。

增量

指示采样类型，因为该对象是计数器。

4000

指定在比较中要使用的值。如果变化率大于 (>) 4000，则代理将向所有已配置的管理器发送陷阱消息。

snmp

指示该条目将 snmp 对象类用于对象状态模型。该对象类将替换为默认值。在此，“*”是无效值。

空

指示该条目将空对象实例用于对象状态模型。该对象实例将替换为默认值。在此，“*”是无效值。

注意：“//”和“/”可以用作分隔符以将实例与系统相关联：

“//hostname/instance”。对于本地系统实例，始终前置“//.”以将实例与本地系统相关联。“//hostname/*”、“//*/instance”和“//*/*”均为无效值。

snmpInPkts

指示该条目将 snmpInPkts 对象属性用于对象状态模型。该对象类将替换为默认值。在此，“*”是无效值。

轻微

指示该条目将轻微重要级别用于对象状态模型。

示例：监控根文件系统上的空间

以下示例配置代理以监控根 (/) 文件系统，并在根文件系统的使用率超过 95% 时发送陷阱消息：

```
monitor filesystem / devCapacity 19 0x0 120 absolute > 95 'Monitor / Filesystem'
'' 'devTableEntry' '/' 'devCapacity' minor
```

devCapacity

指示要监控的特定 MIB 对象实例；在这种情况下，对象实例是 Systems Management Empire MIB devTable 中的 devTableEntry.devCapacity。对象实例未被指定，因为它是基于文件系统的名称而自动确定的。

19

指示该条目是监视器表中的第 19 行。

120

指示代理应每隔 120 秒采样一次。

绝对

指示合适的采样类型，因为代理对代表文件系统使用率的整数（而非计数器）值进行采样。

95

指定在比较中要使用的值。如果文件系统使用率大于 (>) 95%，则代理将向所有已配置的管理器发送陷阱消息。

devTableEntry

指示该条目将 devTableEntry 对象类用于对象状态模型。该对象类将替换为默认值。在此，“*”是无效值。

'/'

指示该条目将 '/' 对象实例用于对象状态模型。该对象实例将替换为默认值。在此，“*”是无效值。

注意：“//”和“/”可以用作分隔符以将实例与系统相关联：

“//hostname/instance”。对于本地系统实例，始终前置“//.”以将实例与本地系统相关联。“//hostname/*”、“//*/instance”和“//*/*”均为无效值。

devCapacity

指示该条目将 devCapacity 对象属性用于对象状态模型。该对象类将替换为默认值。在此，“*”是无效值。

轻微

指示该条目将轻微重要级别用于对象状态模型。

示例：监控 /usr 文件系统上的空间

以下示例配置代理以监控 /usr 文件系统，并在该文件系统的使用率超过 95% 时发送陷阱消息：

```
monitor filesystem /usr devCapacity 20 0x00100500 120 absolute > 95 'Monitor /usr
Filesystem' '' 'devTableEntry' '/usr' 'devCapacity' fatal
```

devCapacity

指示要监控的特定 MIB 对象实例；在这种情况下，对象实例是 Systems Management Empire MIB devTable 中的 devTableEntry.devCapacity。对象实例未被指定，因为它是基于文件系统的名称而自动确定的。

20

指示该条目是监视器表的第 20 行。

0x00100500

- 配置代理，以便当表达式从 True 转变为 False 时发送 monitorClear 陷阱。
- 指定代理应最多发送 X 个连续 monitorEvent 陷阱，然后停止发送，直到该条目重置自身。
- 指定 X 等于 1，这指示代理在等待该条目重置自身之前应只发送一个 monitorEvent 陷阱。

120

指示代理应每隔 120 秒采样一次。

绝对

指示合适的采样类型，因为代理对代表文件系统使用率的整数值进行采样。

95

指定在比较中要使用的值。如果文件系统使用率大于 (>) 95%，则代理将向所有已配置的管理器发送陷阱消息。

devTableEntry

指示该条目将 devTableEntry 对象类用于对象状态模型。该对象类将替换为默认值。在此，“*” 是无效值。

/usr

指示该条目将 /usr 对象实例用于对象状态模型。该对象实例将替换为默认值。在此，“*”是无效值。

注意：“//”和“/”可以用作分隔符以将实例与系统相关联：“//hostname/instance”。对于本地系统实例，始终前置“//./”以将实例与本地系统相关联。“//hostname/*”、“//*/instance”和“//*/*”均为无效值。

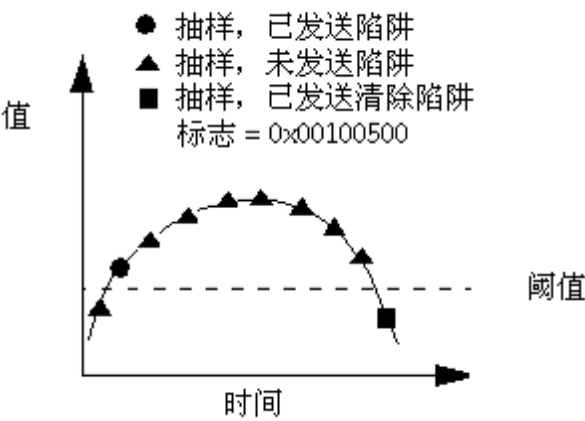
devCapacity

指示该条目将 devCapacity 对象属性用于对象状态模型。该对象类将替换为默认值。在此，“*”是无效值。

致命

指示该条目将致命重要级别用于对象状态模型。

下图显示了代理如何基于该 monitor 指令发送和清除陷阱：



示例：监控进程数

以下示例配置代理以监控当前在系统上执行的进程的数量，并在进程数大于 120 时发送陷阱：

```
monitor oid hrSystemProcesses.0 21 0x0 60 absolute >= 120 'Monitor Number of Processes' '' 'hrSystem' 'null' 'hrSystemProcesses' warning
```

hrSystemProcesses

指示要监控的变量。

21

指示该条目将是监视器表中的索引 21。

60

指示代理应每隔 60 秒对进程数进行一次采样。

绝对

指示采样类型，因为该对象是标尺。

>=

指示代理在进程数大于或等于 120 时发送陷阱。

hrSystem

指示该条目将 **hrSystem** 对象类用于对象状态模型。该对象类将替换为默认值。在此，“*”是无效值。

空

指示该条目将空对象实例用于对象状态模型。该对象实例将替换为默认值。在此，“*”是无效值。

注意：“//”和“/”可以用作分隔符以将实例与系统相关联：

“//hostname/instance”。对于本地系统实例，始终前置“//.”以将实例与本地系统相关联。“//hostname/*”、“//*/instance”和“//*/*”均为无效值。

hrSystemProcesses

指示该条目将 **hrSystemProcesses** 对象属性用于对象状态模型。该对象类将替换为默认值。在此，“*”是无效值。

警告

指示该条目将警告重要级别用于对象状态模型。

用于自主监控的 edgemon 命令

edgemon 命令和相关参数如下所示：

```
oid objectOID index flags interval sampleType oper value "descr" "action"
[supersededBy | objClass objInst objAttr severity]
```

```
filesystem|interface|cpu|disk name variable-name index flags interval sampleType
oper value "descr" "action" [supersededBy | objClass objInst objAttr severity]
```

```
setstatus index status
```

```
delete index
```

列表

使用 | 分隔的参数列表表示，左侧的列表或右侧的列表可以使用。

objectOID

指定要监控的 OID。您可以使用完整的点分十进制值（例如，1.3.6.1.2.1.25.1.5.0）或符号 MIB 名称（例如，hrSystemNumUsers.0）来指定 OID。不管使用哪一种方式，您都必须指定对象实例，对象实例对于非表格 MIB 变量通常为零，并且对应于属于 MIB 表的某一属性的特定实例。

示例：

- 1.3.6.2.1.25.1.5.0
- hrSystemNumUsers.0

index

指定要用于该条目的监视器表的行（索引）。表中各行由索引号唯一标识。保留行 1 至 10 以供代理内部使用，因此索引值必须大于 10 且在表中唯一。

flags

使用十六进制标志值（例如，0x00000001）指定该条目的任何其他操作说明。有关可用标志的详细信息，请参阅[自主监视器表标志](#) (p. 199)。

interval

指定应进行监控的时间间隔（秒）。例如，如果值为 30，则指示代理每隔 30 秒对该条目进行监控。该值必须至少为 30，且是 30 秒的倍数。

sampleType

指定采样类型（**absolute** 或 **delta**）。该值指示代理是对对象的绝对值进行采样，还是取连续采样之差。

oper

指定用于根据提供的阈值评估当前条目值的布尔运算符。

运算符的有效值如下：

- **nop**（无运算）
- **>**（大于）
- **<**（小于）
- **>=**（大于或等于）
- **<=**（小于或等于）
- **=**（等于）
- **!=**（不等于）

默认值： **nop**

值

使用特定运算符指定要与所监控属性的当前值比较的整数阈值。

"descr"

指定表条目的任意说明（长度为 0 至 512 个字符）。

"action"

指定当满足阈值条件且代理发送阈值违反陷阱（传统监视器）或状态更改陷阱（状态监视器）时要运行的命令（长度为 0 到 2048 个字符），包括完整路径和任何参数。对于有状态自主监视器，如果阈值违反不导致对象状态更改，则命令无法运行。如果字符串为空，则代理不为该条目执行操作。

注意：当代理执行操作时，您可以更改默认设置。有关详细信息，请参阅“代理配置”一章。

supersededBy

指定充当该条目的父伙伴或覆盖伙伴的表索引。有效值是现有的或将来的 **SystemEDGE** 监视器索引（正整数值）。默认值是零 (0)，表示默认行为。

请注意，对象状态模型（输入无以外的重要级别）的使用已过时，请禁用该功能。

objClass

指定用于状态管理模型的受监控对象的对象类。定义该属性（以及 objInst 和 objAttr）的值，以创建一个聚合具有相同值的所有条目的受管对象。代理使用 sysedge.oid 文件来分配默认值。

默认： MIB 组名称或表名称（请参阅 sysedge.oid）

objInst

指定用于状态管理模型的受监控对象的对象实例。定义该属性（以及 objClass 和 objAttr）的值，以创建一个聚合具有相同值的所有条目的受管对象。代理会使用 sysedge.oid 文件基于特定属性值为表格对象分配默认值，并分配 null 作为非表格对象的默认值。

objAttr

指定用于状态管理模型的对象属性。定义该属性（以及 objClass 和 objInst）的值，以创建一个聚合具有相同值的所有条目的受管对象。代理会使用 sysedge.oid 文件来分配对应于 MIB 属性名称的默认值。

默认： MIB 属性名称

重要级别

指定要用于对象状态管理模型的重要级别。该值定义发生阈值违反时将分配给条目的状态。有效值如下：

- 无
- 正常
- 警告
- 轻微
- 重大
- 严重
- 致命

值“无”创建传统监视器，它将条目排除在对象状态管理模型之外。

默认值： 无

name

指定要监控的文件系统、网络接口、CPU 或磁盘的名称，具体取决于您输入的命令的类型（filesystem、interface、cpu 或 disk）。例如，C: 表示文件系统。

variable-name

指定 Systems Management Empire MIB 表中的变量，以监控输入的命令（filesystem、interface、cpu 或 disk）所对应的对象。例如，您可以通过指定 devCapacity 作为 variable-name 来监控文件系统的容量。

状态

指定条目的状态，可以为以下值之一：

- active（激活行）
- notInService（停用但保留行）
- destroy（删除自主监视器表行）

edgemon 示例

本节提供的示例介绍如何将 edgemon 命令与 SNMP 版本 1、2c 和 3 结合使用。

示例：使用 edgemon 监控一分钟内的平均负载

以下示例将在代理的自主监视器表的索引 11 处创建一个条目，以便监控系统的一分钟内的平均负载（阈值为 3）。以下示例显示的是 IPv4 地址，但您也可以指定 IPv6 地址。

```
edgemon -h fe80::901:dc19 -c private -v 1 -o oid loadAverage1Min.0 11 0x00 60  
absolute '>' 300 "Monitor 1 minute load average" "" "" "" "" fatal
```

```
edgemon -h fe80::901:dc19 -c private -v 2c -o oid loadAverage1Min.0 11 0x00 60  
absolute '>' 300 "Monitor 1 minute load average" "" "" "" "" minor
```

```
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" ""
```

示例：使用 edgemon 监控硬件中断

以下示例将创建自主监视器表条目，以监控基础系统上的硬件中断数量（阈值为 1000）。以下示例显示的是 IPv4 地址，但您也可以指定 IPv6 地址。

```
edgemon -h fe80::901:dc19 -c private -v 1 -o oid numInterrupts.0 14 0x00500400  
60 delta '>' 1000 "Monitor Interrupt Rate" ""
```

```
edgemon -h fe80::901:dc19 -c private -v 2c -o oid numInterrupts.0 14 0x00500400  
60 delta '>' 1000 "Monitor Interrupt Rate" "" "" "" "" minor
```

```
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" ""
```

代理在索引 14 处创建该条目，并每隔 60 秒对 `numInterrupts` 变量进行采样。标志字段 `0x00500400` 指示代理按如下方式修改默认自主监视器表行为：

0x00000400

指示代理发送多达 *X* 个连续 `monitorEvent` 陷阱，然后不再发送。

0x00500000

包含标志值 *X=5* 以与该指令一起使用。

示例：使用 `edgemon` 监控 `/usr` 文件系统

以下示例将在索引 20 处创建自主监视器表条目，以监控系统的 `/usr` 文件系统的容量是否大于或等于 95%，并每隔两分钟（120 秒）对文件系统检查一次。以下示例显示的是 IPv4 地址，但您也可以指定 IPv6 地址。

```
edgemon -h 143.45.0.12 -c private -v 1 -o filesystem /usr devCapacity 20 0x00100500 120 absolute '>=' 95 "Monitor /usr Filesystem" "" "" "" "" critical
```

```
edgemon -h 143.45.0.12 -c private -v 2c -o filesystem /usr devCapacity 20 0x00100500 120 absolute '>=' 95 "Monitor /usr Filesystem" ""
```

```
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" "" "" "" "" critical
```

删除自主监控条目

要停止监控 MIB 变量，必须从自主监视器表和 `sysedge.cf` 文件中删除相应的条目。从 CA Virtual Assurance 用户界面使用点配置删除条目可以同时实现这两个目标。手动删除要求您分别执行操作以从两个位置删除条目。

在 CA Virtual Assurance 中删除自主监视器条目

1. 访问 CA Virtual Assurance 用户界面。
2. 单击“资源”选项卡。
此时将显示“受管资源”窗格和“数据中心”页面。
3. 在“受管资源”窗格中展开“受管”，然后选择代理驻留的服务器。
此时将显示包含服务器详细信息的页面。
4. 单击“配置”，然后单击“自主监视器”。
此时将出现自主监视器表，其中包含所有现有的自主监视器。
5. 选择要删除的监视器，然后单击“操作”>“删除”。
该条目将从该表中删除。CA Virtual Assurance 会将更改的 `sysedge.cf` 文件加载到代理。代理执行热启动来应用更改。

手动删除自主监视器条目

1. 使用以下带有 `-o` 参数的命令之一运行 `edgemon` 实用工具：

```
edgemon -o delete index  
edgemon -o setstatus index 6
```

index

指定要删除的自主监视器条目的索引值。

在使用 `setstatus` 命令时，值为 6 会将行状态设置为销毁并删除该条目。

注意：如果编辑了 `sysedge.cf`，则必须重新启动代理。

2. 删除本地 `sysedge.cf` 文件中相应条目的 `monitor` 指令，并保存文件。
3. 重新启动代理。

第 9 章： 进程和服务监控

本章介绍了如何使用 SystemEDGE 代理监控进程和服务。代理可以监控进程的许多属性，包括它是否正在运行、大小、CPU、内存和使用率，以及磁盘和网络 I/O 操作的数量。

此部分包含以下主题：

[进程和服务监控概述](#) (p. 229)

[Windows 服务监控](#) (p. 230)

[进程监视器的状态管理](#) (p. 231)

[进程监视器表](#) (p. 233)

[进程监控配置](#) (p. 252)

[用于进程监控的 edgwatch 命令](#) (p. 265)

[删除进程监控条目](#) (p. 271)

[Solaris Zone 进程监控](#) (p. 272)

[有关进程和服务监控的建议](#) (p. 273)

进程和服务监控概述

SystemEDGE 代理使您可以动态地监控在基础系统上运行的重要进程、服务和应用程序的特定属性。可以使用进程监视器表来指定进程、属性、阈值、时间间隔、重要级别、对象信息以及其他值。SystemEDGE 代理会自动监控已定义的进程，并在满足指定的条件时向管理系统发送陷阱。

SystemEDGE 代理可以维护和更新您指定的任何进程监视器的状态值，并使用对象类、实例和属性值来聚合定义了多个监视器的对象的状态。

代理也可以在受管系统上运行命令，以执行管理功能来立即更正阈值违反情况的根源。

有关监控进程组的信息，请参阅“进程组监控”一章。

Windows 服务监控

在 UNIX 系统上，服务或后台进程都是进程，您可以像监控任何其他进程一样监控它们。然而，在 Windows 系统上，服务是使用图形界面启动和停止（例如，通过“服务”控制面板）的特殊类型的进程。Windows 服务在进程内运行，但是它们之间的映射关系却不总是一对一。例如，多个 Windows 服务可能在一个进程内运行。因此，您可以采用两种方式监控 Windows 服务：

- 通过监控 Windows 服务的基础进程来监控 Windows 服务。但有时，找出运行服务的基础进程是很困难的。
- 通过设置标志或使用配置关键字 `watch ntservice` 来指导 SystemEDGE 代理监控 Windows 服务，而不是基础进程。有关详细信息，请参阅[“进程监视器表标志”](#) (p. 245)。

注意：因为 Windows 不跟踪 Windows 服务的进程属性，所以 SystemEDGE 代理仅可监控 Windows 服务的 `procAlive` 属性。要监控特定 Windows 服务的其他属性（例如，`procRSS`），必须监控实施 Windows 服务的基础进程。

进程监视器的状态管理

SystemEDGE 状态管理模型用于维护进程监视器表中的所有条目的状态值，并可定义受管对象以取得总体对象状态。状态管理模型为进程监视器表中的条目提供以下功能：

- 定义代理用于维护每个条目状态值的每个进程监视器条目的重要级别。重要级别可转换为 CIM 兼容的状态。
- 可使用对象类、实例和属性值通过三级别命名方案将一个或多个受监控的资源组为一个或多个受管对象。该机制是灵活的，这样您就可以动态地分组不同功能区域的资源。
- 可创建超级受管对象的层次结构，其中特定对象的状态将传播至层次结构的更高级别上的对象。
- 可在受管对象的总体状态更改时通过状态更改陷阱发送通知

状态管理模型集成于 CA Virtual Assurance 对象模型中。AOM 数据库中的每个受管对象都拥有一个 CIM 兼容的状态，SystemEDGE 受管对象和状态通过 SystemEDGE 平台管理模块反映在 CA Virtual Assurance 中。通过 CA Virtual Assurance 管理 SystemEDGE 可使受管对象状态信息在 CA Virtual Assurance 用户界面中可视化，并可通过 CA Virtual Assurance 事件视图查看状态更改陷阱。

有关使 SystemEDGE 受管对象在 CA Virtual Assurance 中可视化的详细信息，请参阅 CA Virtual Assurance 文档。

有关配置进程监视器来利用状态管理模型的更多信息，包括属性值和语法示例，请参阅本章的其余部分。

受管对象创建

控制使用对象类、实例和属性值从进程监视器表创建受管对象。默认情况下，SystemEDGE 将为每个唯一类、实例和属性进程监视器创建受管对象。如果多个进程监视器针对这三个属性包含相同的值，则代理组会将他们组合到单个受管对象，并且为对象获得总体状态。

为了说明使用该机制的灵活性，请考虑以下情形：

- 使用相同的类、实例和属性值定义三个进程监视器条目，用于监控进程的实际内存使用量。监视器对大于 85% 的实际内存使用量定义了警告重要级别，即实际内存使用量大于 90% 为重大，而实际内存使用量大于 95% 为严重。代理将这些监视器组合到一个受管对象中，并且将最差的当前重要级别保留为对象状态。
- 使用相同的类、实例和属性值定义三个进程监视器条目，来监控在您的企业中关联的不同功能区域的进程。例如，您可以组合基本操作系统进程监视器、应用程序进程监视器以及数据库进程监视器，将警告重要级别分配给每个进程监视器。在这种情况下，代理只有在三个进程监视器阈值都被违反时，才发送警告状态更改陷阱。
- 使用对象类的相同值或对象实例或属性的星号值定义进程监视器条目的超级聚合。该方案对大量的对象进行分组，并没特意针对每个对象定义聚合规则。例如，可以聚合监控相同进程的不同属性的所有对象，以便该进程的所有进程监视器将最差当前状态传播到顶层进程对象。

注意：还可以将代理配置为聚合更高级别的所有对象。有关详细信息，请参阅“代理配置”一章。

有关这些方案的语法示例，请参阅[“进程监控示例”](#) (p. 259)。

如果您未定义对象类、实例和属性值，SystemEDGE 将根据下列约定为这些属性分配默认值：

- 类：processEntry
- 实例：进程监视器的 pmonRegExpr 属性的值，该属性用于定义确定要监控的进程的正则表达式
- 属性：进程监视器的 pmonAttribute 属性的值，该属性用于定义要监控的进程属性

详细信息

[对象状态](#) (p. 60)

[对象合计](#) (p. 61)

[超级聚合](#) (p. 62)

[状态管理和聚合的工作原理](#) (p. 63)

聚合状态表

“聚合状态”表执行创建受管对象代表以及获取总体对象状态所必需的工作。“聚合状态”表是 SystemEDGE 自主监视器和进程监视器的已筛选视图。该表包括基于每个监视器条目提供的受管对象命名和状态信息计算的聚合状态信息。

该表使用相同的对象类、实例和属性值，为所有自主监视器和进程监视器创建“聚合状态”表条目（或在更高级别上在一起分组的对象）。自主监视器和进程监视器仅作为单独条目存在于“自主监视器”和“进程监视器”表中。“聚合状态”表维护这些监视器的分组，对表进行连续轮询，并在受管对象状态发生更改时发送状态更改陷阱。

注意：创建新的受管对象或删除现有的受管对象时，“聚合状态”表也发送陷阱。

有关“聚合状态”表如何维护受管对象及其状态的详细信息，请参阅“概念”一章。有关表列的说明，请参阅聚合状态表。

进程监视器表

进程监视器表提供有关代理当前监控的每个进程（或 Windows 服务）的信息。每行表示一个已定义的进程监视器条目。

对于每个条目，该表提供以下类型的信息：

- 索引
- 描述
- 代理轮询变量的间隔
- 示例类型（绝对值或增量值）
- 正在监控的属性
- 代理正在监控的当前值
- 阈值
- 最后示例调用
- 已经发送的陷阱数
- 最后陷阱
- 标志
- 发送陷阱时采取的操作
- 正则表达式

- 抽样的最小值和最大值
- 当前进程 ID
- 行状态
- 事件数
- 对象类
- 实例
- 属性
- 定义的重要级别
- 当前监视器状态
- 区域正则表达式
- 进程名称

进程监视器表列

以下列表描述进程监视器表的列。有关进程监视器表的完整描述，请参阅《Systems Management Empire MIB 规格》（位于代理安装的 `mib` 子目录中的 `empire.asn1`）。

pmonIndex

为该条目定义唯一的行索引（1 到 `MAXINT`）。

权限：只读

pmonDescr

定义正在监控的进程和属性的说明（长度为 0 至 512 个字符）。

权限：读写

pmonInterval

定义代理监控进程的频率（以秒为单位）。该值必须至少为 30，且是 30 秒的倍数。

权限：读写

默认值：60

pmonSampleType

定义代理是使用抽样属性的绝对值 (`absoluteValue(1)`)，还是使用连续采样之间 (`deltaValue(2)`) 之间的差值。例如，使用 `deltaValue` 监控计数器属性，因为该值可提供更改率。使用 `absoluteValue` 监控 Gauge，因为该值提供对象的精确值。

权限：读写

默认：absoluteValue(1)

pmonAttribute

指定正在监控的进程属性。有关支持的属性的完整列表，请参阅[“进程属性”](#) (p. 241)。例如，要监控进程来验证其是否处于运行状态，请指定 `procAlive` 属性。要跟踪特定应用程序或进程接收的数据包的数目，请指定 `procMsgsSent`。

权限：读写

pmonCurrVal

指定上次记录的正在监控的进程的属性值。每隔 `pmonInterval` 秒，代理将更新该字段，用于反映该属性的最近读数。

监控 `procAlive` 时，将针对 Windows 服务从 `hrSWRunStatus` 变量（位于 `hrSWRunTable`）或 `ntServiceState`（位于 `ntServiceTable`）映射该值。
`hrSWRunStatus` 的可能值：

1、2 或 3

表示进程正在运行。

4

表示进程未运行（在这种情况下，通常是僵尸进程）

`ntServiceState` 的可能值：

1

表示服务正在运行

2,3,4

表示服务未运行

权限：只读

pmonOperator

指定用于评估表达式 *currval operator value* 的布尔运算符。运算符可以为以下任何一种：

- **nop(1)**（不执行操作；监控对象的值，但不评估布尔表达式）
- **gt(2)**（大于）
- **lt(3)**（小于）
- **ge(4)**（大于或等于）
- **le(5)**（小于或等于）
- **eq(6)**（等于）
- **ne(7)**（不等于）

权限：读写

默认：nop(1)

pmonValue

定义代理在每个监控周期期间比较受监控进程属性当前值的整数值。如果比较评估为 **True**，则代理将更改监控状态（对于显示无重要级别之外的进程监视器）或发送阈值违反陷阱（对于显示无重要级别的进程监视器）。例如，如果要在进程的实际内存使用量超过 75% 时收到通知，请将代理比较属性当前值的 **pmonValue** 设为 75。

权限：读写

pmonLastCall

定义代理最后抽样（调用）其监控的进程属性的时间（基于 **sysUpTime**）。0 表示进程还没有进行抽样。

权限：只读

pmonNumTraps

定义已针对该条目发送的陷阱数目。该值将跟踪 **processStopTrap**、**processStartTrap**、**processClearTrap**、**processNotReadyTrap** 以及 **processThresholdTrap**，这些陷阱是代理在进程停止或显示无重要级别的进程监视器中发生违反阈值时发送的。该值可用于确定传统监视器发生异常情况的频率。它还提供检测错过的陷阱消息的方法。

权限：只读

注意：有关针对状态进程监视器的受管对象发送的状态更改陷阱数目的信息，请参阅“合计状态表”。

pmonLastTrap

定义代理针对该条目最后发送陷阱的时间（基于 `ysUpTime`）。0 表示尚未发送陷阱。该值将跟踪 `processStopTrap`、`processStartTrap`、`processClearTrap`、`processNotReadyTrap` 以及 `processThresholdTrap`，这些陷阱是代理在进程停止或显示无重要级别的进程监视器中发生违反阈值时发送的。

权限：只读

注意：有关针对状态进程监视器的受管对象发送的上一次状态更改陷阱的信息，请参阅“合计状态表”。

pmonFlags

定义表示其他行为语义的正整数标志，条目在其操作过程中将遵循这些语义。有关可用标志和设置标志的详细信息，请参阅[“进程监视器表标志”](#) (p. 245)。

权限：读写

pmonAction

定义发送传统陷阱（传统监视器）或状态更改陷阱（状态监视器）时运行的带有任何参数的引用命令（长度为 0 至 2048 个字符）。对于状态进程监视器，如果阈值违反未导致对象状态发生变化，该命令将不运行。如果字符串为空，则代理不为该条目执行操作。

注意：请勿使用 Windows 批处理文件进行操作；它们施加严格的编程限制，并且经常错误地使用桌面应用程序。应使用更强大且灵活的脚本语言，如 Perl 或 Visual Basic。

权限：读写

pmonRegExpr

指定当代理尝试获得要监视的应用程序的进程 ID 或进程时要应用的正则表达式。对于 Windows 服务监控，该正则表达式与要监控的 Windows 服务的名称匹配。用户需要指定进程名称或服务名称的正则表达式，而不需要指定可能发生变化的进程 ID（PID）或服务索引。代理使用该用户指定的名称来查找要监控的进程（或服务）。默认情况下，进程监视器表将继续尝试应用正则表达式，直到找到新的 PID 或服务（如果进程或服务停止运行）。

权限：读写

pmonMinValue

定义从开始轮询进程属性以来，代理观察到的最低（最小）值。

权限：只读

pmonMaxValue

定义自开始调查进程属性以来，代理观察到的最高（最大）值。

权限：只读

pmonCurrentPID

指定当前正在监控的进程的 PID 或 NT 服务索引。

权限：读写

pmonRowStatus

定义行状态，行状态可以是以下状态之一：

- 活动
- notInService
- notReady
- createAndGo
- createAndWait
- 销毁

通常，行处于 active 或 notInService 状态。这些值的含义与由 SNMPv2 SMI RowStatus 文本约定定义的值含义相同。

权限：读写

默认：createAndWait(5)

pmonNumEvents

指定该监视器条目的事件数，取决于 pmonFlags。该事件不总是暗示陷阱。

权限：读写

pmonObjClass

定义用于对象状态管理模型的受监控 MIB 对象的对象类。代理使用 sysedge.oid 文件来分配默认值。类名可用于指定被监控对象表。如果 SystemEDGE 未确定具有指定名称的现有表，SystemEDGE 会创建类别。该类别可以指定监控对象标识符 (SNMP OID)，也可出于聚合目的而被引用。

重要信息！ 只有当监控条目解析为现有 SNMP OID 时，新建的引用才有效。

pmonObjInstance

定义用于对象状态管理模型的受监控 MIB 对象的对象实例。如果不配置该值并且代理无法确定一个默认值，则该值为“未知”。非表对象的默认值为“空”。对于远程对象的无缝监视，实例名以前缀 `//hostname/` 开始。

对于本地对象，代理则会在每个实例名前面添加前缀 `//.`（包括“null”）。该属性不能设置为空字符串或“*”。任何此类请求都将被忽略。如果在解析时实例名无法与相应实例相匹配，SystemEDGE 将使用指定的实例名创建对此监控条目的 OID 的引用。

重要信息！ 只有当监控条目解析为现有 SNMP OID 时，新建的引用才有效。

pmonObjAttribute

定义用于对象状态管理模型的受监控 MIB 对象的对象实例。代理使用 `sysedge.oid` 文件为与 MIB 属性名称相应的对象分配默认值。如果解析监控条目期间属性名称未知，SystemEDGE 将使用指定的名称创建属性。

权限：读写

默认：受监控的 OID 的 ASCII MIB 属性名称

重要信息！ 只有当监控条目解析为现有 SNMP OID 时，新建的引用才有效。

pmonSeverity

定义将用于该对象状态管理模型的重要级别。该值定义发生阈值违反时将分配给条目的状态。有效值如下：

- 无(1)
- 正常(2)
- 警告(3)
- 轻微(4)
- 重大(5)
- 严重(6)
- 致命(7)

值“无”创建传统监视器，它将条目排除在对象状态管理模型之外。

默认值：无

pmonCurrState

定义条目的当前派生状态，该状态已通过重要级别和当前违反条件进行评估，同时考虑 **pmonFlags** 属性中定义的任何延迟值。有效值如下：

- 未知(1)
- 正常(2)
- 警告(3)
- 轻微(4)
- 重大(5)
- 严重(6)
- 致命(7)
- 运行(11)
- 关闭(12)

仅为传统监视器显示值“运行”和“关闭”（重要级别为“无”）。只有当该状态是“聚合状态”表中相应受管对象的所有监视器中最差状态时，该列中的状态更改才导致状态更改陷阱。

权限：只读

pmonProcName

定义由正则表达式匹配的进程名称。在 Solaris 10 和更新的版本中，如果使用区域正则表达式，则名称使用 *MatchedZoneName/MatchedProcName* 格式。

抽样进程监视器表条目

本节提供了一个抽样进程监视器表条目，该条目将指导代理监控 httpd 进程，以确保该进程启动并运行。

索引	Desc	间隔	SampleType	属性
21	“监控 httpd”	60	absoluteValue	procAlive
NumTraps	LastCall	值	Oper	CurrVal
{0}	0d 22:40:00	4	=	3
LastTrap	RowStatus	操作	标志	RegExpr
{0}	活动	""	0x00000100	httpd

ObjClass	ObjInstance	ObjAttribute	重要级别	CurrState
进程	httpd	procAlive	警告	正常

该抽样条目提供以下信息：

- 示例中的该条目位于进程监视器表的第 21 行。
- 它指导代理每隔 60 秒监控一次 Web 服务器后台进程 (httpd)。
- 正在监控的属性是进程 (procAlive) 的状态。
- 当前值 3 表示进程正在正常运行（就操作系统而言）。如果进程状态转到 4，或如果进程停止运行，则代理会将当前状态设置为警告，如果该监视器表示合计状态表中的相应受管对象的最差状态，代理将发送状态更改陷阱。
- RowStatus 列显示该条目处于活动状态。
- 操作列为空 ("")，其表示进程停止运行或再次启动时，代理未调用操作。
- 标记值 (0x00000100) 将指导代理监控父 httpd 后台进程，而不是子进程。有关标志的详细信息，请参阅 [“进程监视器表标志”](#) (p. 245)。
- 类、实例和属性值将该进程监视器条目作为合计状态表中包含这些值的受管对象的一部分进行建立。您可以使用这些相同值定义其他进程监视器条目，以针对不同重要级别进行监控，或可以将该条目与其他相关对象甚至自主监视器（如监控相同进程的其他属性）聚合在一起。

进程属性

以下列表描述 SystemEDGE 代理针对每个特定进程可监控的属性，或每个属性的服务 procAlive 和 SNMP 类型。您可以使用 SNMP 类型选择抽样类型和运算符。例如，absoluteValue 采样通常最适合整数或 gauge 属性，而 deltaValue 采样通常最适合计数器属性。

注意：代理监控特定进程属性的能力取决于基础操作系统跟踪关联的参数或度量标准的能力。

procAlive（所有 OS）

定义过程或服务是否正在运行。

类型：布尔值

procMemory（所有 OS）

定义进程使用的实际内存的百分比 (0-100)。

类型: Gauge

procSize（所有 OS）

定义文本、数据和堆栈段的大小 (KB)。

类型: Gauge

procRSS（所有 OS）

定义进程的实际内存（驻留集）大小 (KB)。

类型: Gauge

procTime（所有 OS）

定义进程的累积的 CPU 时间（以秒为单位）。

类型: 整数

procInBlks（AIX、Solaris）

定义进程输入的数据块数。

类型: 计数器

procOutBlks（AIX、Solaris）

定义进程输出的数据块数。

类型: 计数器

procMsgsSent（AIX、HP-UX、Solaris）

定义进程接收的消息数。

类型: 计数器

procMsgsRecv（AIX、HP-UX、Solaris）

定义进程发送的消息数。

类型: 计数器

procNice（所有 OS）

定义进程的优先级（nice 值）。

类型: 整数

procNumThreads（Windows、AIX、Solaris、Linux）

定义正在进程内运行的线程数。

类型: 整数

procNumSwaps (AIX、HP-UX、Solaris)

定义已交换该进程的次数。

类型：计数器

procSysCalls (Hp-ux、Solaris)

定义该进程调用的系统调用数。

类型：计数器

procMinorPgFlts (AIX、HP-UX、Linux、Solaris)

定义由进程导致的轻微页面错误数。@

类型：计数器

procMajorPgFlts (所有 OS)

定义由进程导致的主要页面错误数。

类型：计数器

procVolCtx (AIX、Solaris)

定义由进程导致的自愿上下文切换的次数。

类型：计数器

procInvolCtx (AIX、Solaris)

定义由进程导致的非自愿上下文切换的次数。

类型：计数器

procTimePermil (所有 OS)

定义乘以比例系数 1000 后的 CPU 使用率 (‰)。

类型：整数

行创建对象

进程监视器表的 pmonIndex 列充当行索引，用于区别表中的行。保留 1 到 10 行以供 SystemEDGE 代理内部使用。您可以在 11 到 MAXINT 这个范围内对行进行配置。

您可以使用下列 MIB 对象和进程监视器表来优化行创建。

pmonUnusedIndex

在您执行该对象的 SNMP Get 时，将返回进程监视器表的未用过的索引号。查询该对象来获取可用于创建行的索引号。

pmonMatchDescr

确定与 **pmonDescr** 属性的特定条目说明相对应的索引号。对包含 **pmonDescr** 属性值的 MIB 对象执行 **SNMP Set**，致使代理在进程监视器表中搜索条目，并且将最后一个匹配条目的索引值放入 **monMatchIndex** MIB 对象。

pmonMatchIndex

使用 **pmonMatchDescr** 时，将特定条目说明与索引号相匹配。

按照本地策略，您可以选择为系统管理预留行块。然后可以在保留的行块中定义条目，而无需担心行可能已经被其他用户条目使用。按照本地的策略，当所有其他用户定义用户配置的条目时，应使用在保留范围以外的行索引。

通过保留行块，您可以定义要跨所有计算机监控的一组一致的条件（行条目），以便同样的条件定义在每台计算机的相同行数中。例如，您可以使用第 11 行 (**monIndex = 11**) 定义用于监控 **swapCapacity** 变量的条目，并可将该配置分发给每个系统，以便第 11 行对每个系统中的 **swapCapacity** 变量进行监控。

在 **CA Virtual Assurance** 上配置代理可以自动执行该过程。您可以使用特定行号配置条目块，并将该配置部署到一个操作中的所有受管系统。此外，**CA Virtual Assurance** 将检测您是否定义已在使用中的行并禁止该操作。

有关在 **CA Virtual Assurance** 中定义行的详细信息，请参阅 **CA Virtual Assurance** 文档。

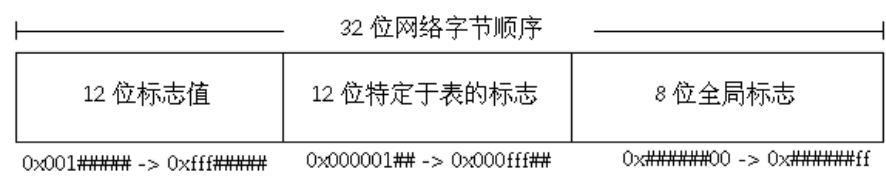
进程监视器表标志

进程监视器表的 pmonFlags 列是 32 位无符号整数字段，可用于指定相应行的其他行为语义。

默认情况下，进程监视器表行将执行以下操作：

- 尝试对自身进行重新初始化
- 发送 SNMP 陷阱（仅限传统监视器）
- 将事件记录到普通日志文件
- 调用操作（如果进行了配置）

您可以设置不同标志位来改变这些默认值。代理将以十六进制（基础 16）码解释所有的标志。下列说明列出了进程监视器表标志 (pmonFlags) 字段的组成：



标志值由三个字段组成：

- 字段 1 包含为 Systems Management Empire MIB 的监控表定义的常用表标志。该部分是标志的低 8 位。
- 字段 2 包含分别为每个监控表定义的表特定的标志。该字段定义常用表标志后面接下来的 12 个低位。
- 字段 3 包含为整数值保留的 12 个高位，以便用于表特定的标志。该字段定义了专用于进程监视器表的标志。

以下部分解释了每个标志位。您可以通过逻辑“OR”运算组合标志值。

以下列表描述进程监视器表标志：

0x00000001

禁止执行该条目的操作。

0x00000002

禁止发送该条目的 SNMP 陷阱。

注意：进程监视器表仅控制针对传统监视器发送陷阱。对于状态进程监视器，状态更改陷阱可能仍然会从合计状态表发生。

0x00000004

禁止尝试重新初始化该条目。默认情况下,如果目标进程已重新启动,代理将定期尝试通过扫描进程表对条目进行重新初始化,以确定新进程的 ID。

0x00000008

禁止在 `sysedge.log` 文件中记录该条目的陷阱。设置该位对发送陷阱没有影响。在事件频繁发生或将特定条目用作代理监控信号时,禁用事件记录会很有用。

0x00000010

每次当代理尝试重新初始化进程监控,但无法匹配进程时,将针对该条目发送连续 `processStop` 陷阱。代理的默认行为是,在进程停止时发送单个 `processStop` 陷阱,并尝试定期重新初始化该条目。如果启用该功能,则每次重新初始化失败时,代理都会发送一个额外的 `processStop` 陷阱。代理在任何情况下都不发送 `processStart` 和 `processStop` 陷阱,除非相应条目正在监控 `procAlive` 进程属性。

注意: 仅当代理正在监控 `procAlive` 属性时,该标志才有效。当您正在监控 Windows 服务时,如果该服务未运行,则条目不会进入 `notReady` 状态。如果设置该标志,则当服务未运行时, `SystemEDGE` 会生成 `processStop` 陷阱,并对条目运行任何关联的操作,即使该服务保持就绪状态也是如此。

0x00000020

禁止将默认参数传递给操作脚本或程序。`SystemEDGE` 通常传递默认操作参数(表示陷阱类型、说明字段等)。有关操作参数的详细信息,请参阅[进程监视器表操作](#) (p. 250)。

0x00000040

禁止发送该条目的 `notReady` 陷阱。这包括不记录以及不执行 `notReady` 陷阱的操作。

0x00000100

监控进程组中的父进程。许多应用程序和服务(例如, `Web` 服务器 `httpd` 后台进程)经设计,允许初始后台进程生成子进程来处理实际服务请求。这些子进程常常处理若干个请求,然后退出。在这种情况下,首选监控主父后台进程,而不是子进程。如果启用该功能,代理会搜索并监控父后台进程,而不是它发现的与进程正则表达式匹配的进程。

代理通过扫描进程监视器表,查找与进程正则表达式 (`pmonRegExpr`) 名称匹配的进程,来执行该搜索。如果进程的匹配父进程也匹配,则代理会返回父进程。该搜索算法仅适用于父子关系,无法处理派生后台进程的后台进程。该功能不适用于监控 Windows 服务。

0x00000200

禁用针对该条目发送 `processStart` 陷阱（无日志和无操作）。

0x00000400

当进程监视器表达式从 `True` 转变为 `False` 时，针对该条目发送 `processClear` 陷阱（日志和操作）。仅当正被监控的属性不是 `procAlive` 时，该功能才适用。

0x00000800

设定要监控的进程时，匹配进程名称和参数。默认情况下，代理仅根据进程名称进行匹配。如果启用该选项，则代理会将 `pmonRegExpr` 应用于进程名称和进程参数。有时在区分类似进程或同一个应用程序或二进制文件的多个调用时，这很有必要。

0xXXX01000

只有在第 *X* 次连续事件之后发送 `processThreshold` 陷阱或更改当前状态。如果启用该功能，则会指示代理等待，直到发生第 *X* 次连续事件，然后再发送 `processThreshold` 陷阱或更改进程监视器的状态。发生第 *X* 次事件后，代理将针对每个随后的连续 `True` 表达式计算发送 `processThreshold` 陷阱。如果阈值表达式从 `True` 转变回 `False`，则该行将自行重置，代理将从零开始计算事件。该标志也适用于操作执行。您可以通过标志值字段指定 *X* 的值。事件日志记录不受该标志位的影响。有关示例，请参阅[进程监控示例](#) (p. 259)。

0xXXX02000

发送最多 *X* 个连续 `processThreshold` 陷阱，然后不再发送。如果启用该功能，则会设置当进程超过阈值时可发生的连续 `processThreshold` 陷阱和操作执行数目的上限。阈值表达式从 `True` 转变为 `False` 后，该行将自行重置，代理将从零开始计算事件。该标志也适用于操作执行。您可以通过标志值字段指定 *X* 的值。事件日志记录不受该标志位的影响。有关示例，请参阅[进程监控示例](#) (p. 259)。

0xXXX04000

监控要处于运行状态的 *X* 个进程所在的进程组（仅限 `procAlive` 监控）。

0x00008000

监控与相应正则表达式匹配的 Windows 服务。如果设置该标志，则会指示 SystemEDGE 监控 Windows 服务表中匹配 Windows 服务的 `procAlive` 属性。

在 Solaris 10 上监控区域中的进程。`pmonProcRegExpr` 属性应该采用 `ZoneRegExp/ProcRegExp` 格式。

0x00010000

禁用任何 processStart 处理。如果启用该标志，则 SystemEDGE 不会调用操作、记录事件，或在 processStart 事件发生时发送陷阱（有关传统支持，另请参见 000 002 00）。

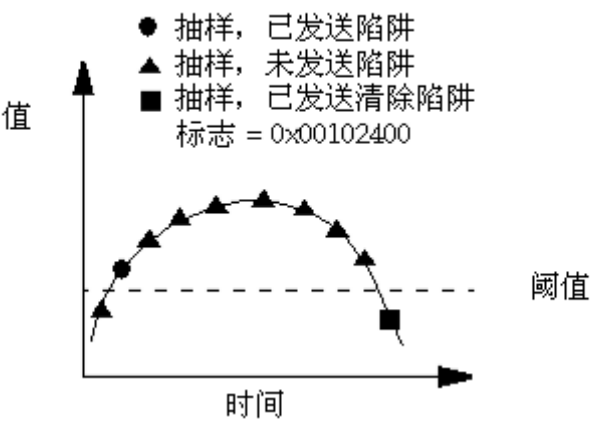
0x00020000

聚合相同对象的具有相同重要级别的监视器的状态，使其具有 AND 关系。设置有该标志且具有相同重要级别的所有连接的监视器必须具有违反条件，才能相应地更改聚合状态。

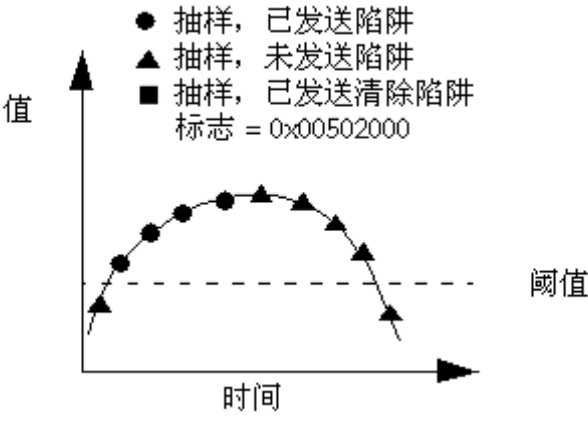
0x###00000

若干标志位使用值 X 发送陷阱和执行操作。值 X 被指定为高阶 12 位标志字段。利用该字段的标志位是互斥的。

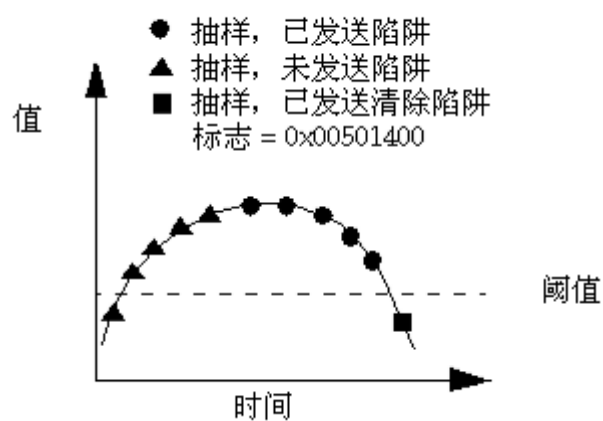
下图说明代理发送一个陷阱以指示被监控对象超过阈值，然后当值降低到低于阈值时，再发送 processClear 陷阱。



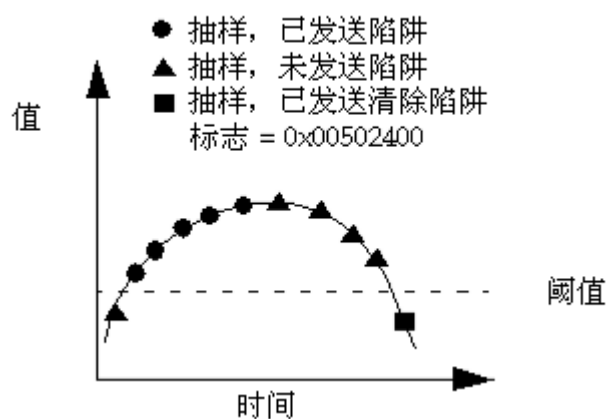
下图说明代理发送四个陷阱以指示被监控对象的值高于阈值。当值降低到低于阈值时，代理将不发送 processClear 陷阱。



下图说明代理等到事件发生若干次后，才开始发送陷阱。在被监控对象的值降低到低于阈值之前，代理不发送陷阱；在被监控对象的值降低到低于阈值时，代理将发送 processClear 陷阱。



下图说明代理仅按指定的次数发送陷阱。当被监控对象的值降低到低于阈值时，代理将发送 processClear 陷阱。



进程监视器表操作参数

SystemEDGE 代理可在操作命令被调用时向其提供若干默认参数。这些参数是对您在操作字符串中指定的任何参数的补充，并在您指定的那些参数之后通过命令行传递。默认参数与针对进程监视器表发送的 SNMP 陷阱中提供的参数一样。

以下列表介绍进程监视器表操作的默认参数：

注意：通过在表条目中设置 0x00000020 标志，可以防止代理在命令中包含这些操作参数。

trapType

定义要发送的陷阱的类型，它将是以下类型之一：

- processStop
- processStart
- processThreshold
- processClear

pmonIndex

定义分配给表条目的索引。

pmonDescr

定义表条目说明。

pmonAttribute

定义条目要监控的进程属性。

pmonCurrVal

定义针对表条目获得的当前值。

pmonOperator

定义由表条目使用的运算符。

pmonValue

定义表条目应用的比较值或阈值。

pmonFlags

定义与表条目关联的标志。

pmonRegExpr

定义代理用于查找要监控的进程的进程 ID 的正则表达式。

pmonCurrentPID

定义所监控的当前进程的进程 ID (PID)。如果您正在监控 Windows 服务（通过 `watch ntservice` 指令或通过设置 `0x8000` 标志），则此列表表示该进程或服务监控条目已经获得的 NT 服务 MIB 表的索引。

pmonObjClass

定义用于对象状态管理模型的受监控 MIB 对象的对象类。代理使用 `sysedge.oid` 文件来分配默认值。类名可用于指定被监控对象表。如果 SystemEDGE 未确定具有指定名称的现有表，SystemEDGE 会创建类别。该类别可以指定监控对象标识符 (SNMP OID)，也可出于聚合目的而被引用。

重要信息！ 只有当监控条目解析为现有 SNMP OID 时，新建的引用才有效。

pmonObjInstance

定义用于对象状态管理模型的受监控 MIB 对象的对象实例。如果不配置该值并且代理无法确定一个默认值，则该值为“未知”。非表对象的默认值为“空”。对于远程对象的无缝监视，实例名以前缀 `//hostname/` 开始。

对于本地对象，代理则会在每个实例名前面添加前缀 `///`（包括“null”）。该属性不能设置为空字符串或“*”。任何此类请求都将被忽略。如果在解析时实例名无法与相应实例相匹配，SystemEDGE 将使用指定的实例名创建对此监控条目的 OID 的引用。

重要信息！ 只有当监控条目解析为现有 SNMP OID 时，新建的引用才有效。

pmonObjAttribute

定义用于对象状态管理模型的受监控 MIB 对象的对象实例。代理使用 `sysedge.oid` 文件为与 MIB 属性名称相应的对象分配默认值。如果解析监控条目期间属性名称未知，SystemEDGE 将使用指定的名称创建属性。

权限： 读写

默认： 受监控的 OID 的 ASCII MIB 属性名称

重要信息！ 只有当监控条目解析为现有 SNMP OID 时，新建的引用才有效。

pmonCurrState

定义表条目的当前状态。

有关 SystemEDGE 代理发送的陷阱的详细信息，请参阅“概念”一章和附录“私人企业陷阱”。

查看进程监视器表

您可以使用 CA Virtual Assurance 来查看受管代理的进程监视器表的内容。

在 CA Virtual Assurance 中查看进程监视器表

1. 访问 CA Virtual Assurance 用户界面。
2. 单击“资源”选项卡。
此时将显示“受管资源”窗格和“数据中心”页面。
3. 在“受管资源”窗格中展开“受管”，然后选择代理驻留的服务器。
此时将显示包含服务器详细信息的页面。
4. 单击“配置”，然后单击“进程监视器”。
此时将出现进程监视器表，其中包含所有现有的进程监视器。

进程监控配置

您可以控制 SystemEDGE 代理监控的进程和进程属性，方法是在进程监视器表中添加、删除或修改相应的条目。

可以按下列方式配置进程监视器表：

动态使用 CA Virtual Assurance 中基于文件的配置

CA Virtual Assurance 可以配置安装在它从 CA Virtual Assurance 用户界面管理的任何系统上的代理。建议使用该配置方法，因为该方法可以利用一种用户友好且安全的机制动态地定义和传送监视器，该机制不需要代理重新启动。您可以为单个代理配置监控，也可以定义监控策略以在多个受管系统上应用。当您执行单个配置更改或应用配置策略时，CA Virtual Assurance 会将新的配置文件传送给代理系统。CA Virtual Assurance 可以在整个数据中心中充当代理配置的中心点，您可以覆盖或拒绝通过其他方式执行的更改。

有关使用 CA Virtual Assurance 配置代理的详细信息，请参阅 CA Virtual Assurance 文档。

从管理系统（例如 eHealth AdvantEDGE View、CA Spectrum 或 CA NSM）动态使用 SNMP Set 命令

各种管理系统均支持通过 SNMP Set 命令将监视器添加到 SystemEDGE 中。eHealth 通过 AdvantEDGE View 接口提供该功能，CA NSM 通过 Agent View 启用配置，CA Spectrum 通过 OneClick 启用配置。通过 CA Virtual Assurance 进行动态配置所提供的所有功能都无法通过这些方法获得。

有关 SNMP Set 配置的详细信息，请参阅相应管理系统的文档。

通过在本本地 `sysedge.cf` 文件中指定自主监视器表的条目来执行手动配置

手动配置要求您使用特定指令将条目添加到本地 `sysedge.cf` 文件中。当从 `sysedge.cf` 文件直接配置代理时，必须执行完全重新启动才能使更改生效。请使用位于代理数据目录中的 `sysedge.cf` 文件进行直接配置。

本节介绍如何在本地 `sysedge.cf` 文件中直接配置自主监控，但介绍的属性是所有形式的配置所必需的。

也可以通过 `edgwatch` 实用工具动态添加、删除或修改进程监视器表条目。有关详细信息，请参阅 `edgwatch` 实用工具—监控进程。

watch process 指令—将条目添加到进程监视器表

watch process 指令可用于通过 **sysedge.cf** 文件将进程监视器条目直接添加到进程监视器表。下列参数表示进程监视器表中的各列。

使用如下所述的语法向代理数据目录中的 **sysedge.cf** 文件添加一行，保存文件，然后重新启动代理以使更改生效。

使用该语法可监控任何可用的进程属性，包括 **procAlive**。

如果输入 **procAlive** 作为属性来监控进程是否正在运行，则会忽略某些值。**procAlive** 属性监视器基于表达式 “**hrSWRunStatus = 4**” 进行监控。如果进程停止运行或变得无效，则代理会发送 **processStop**（传统监视器）或状态更改陷阱（有状态监视器）并调用任何配置的操作。

如果指定使用 **0x00008000** 标志监控 Windows 服务，则会忽略相同的值，且代理将基于表达式 “**ntServiceState = 1**” 监控服务 **procAlive** 属性。如果服务停止运行或变得无效，则代理会发送 **processStop**（传统监视器）或状态更改陷阱（有状态监视器）并调用任何配置的操作。

使用 **watch process** 指令将条目添加到进程监视器表，如下所示：

```
watch process attribute 'regexpr' index flags interval stype oper thresh ['descr'
['action' ['objClass' 'objInst' 'objAttr' severity]]]
```

属性

指定 SystemEDGE 代理针对指定的阈值监控的进程属性。可以从[进程属性](#) (p. 241) 中指定任何进程属性，包括 **procAlive**。

'regexpr'

指定尝试获得要监控的应用程序或进程的进程 ID 时要使用的正则表达式。指定要监控的进程名称的正则表达式。

index

指定要用于该条目的监视器表的行（索引）。表中各行由索引号唯一标识。保留行 1 至 10 以供代理内部使用，因此索引值必须大于 10 且在表中唯一。

flags

使用十六进制标志值（例如，**0x00000001**）指定该条目的任何其他操作说明。有关可用标志的详细信息，请参阅[进程监视器表标志](#) (p. 245)。

interval

指定应进行监控的时间间隔（秒）。例如，如果值为 30，则指示代理每隔 30 秒对该条目进行监控。该值必须至少为 30，且是 30 秒的倍数。

stype

指定代理应对属性的绝对值 (**absoluteValue**) 进行抽样, 还是采用连续抽样之间的差 (**deltaValue**)。使用 **deltaValue** 可监控计数器变量的变化速率, 使用 **absoluteValue** 可监控计量器的精确值。

注意: 如果正在监控 **procAlive** 属性, 则代理会忽略该值。

oper

指定根据所提供的阈值评估当前条目值的布尔运算符。

运算符的有效值如下:

- **nop** (无运算)
- **>** (大于)
- **<** (小于)
- **>=** (大于或等于)
- **<=** (小于或等于)
- **=** (等于)
- **!=** (不等于)

默认值: **nop**

注意: 如果正在监控 **procAlive** 属性, 则代理会忽略该值。

thresh

使用特定运算符指定要与所监控属性的当前值比较的整数阈值。

注意: 如果正在监控 **procAlive** 属性, 则代理会忽略该值。

'descr'

指定表条目的任意说明 (长度为 0 至 512 个字符)。

'action'

指定当满足阈值条件且代理发送阈值违反陷阱 (传统监视器) 或状态更改陷阱 (状态监视器) 时要运行的命令 (长度为 0 到 2048 个字符), 包括完整路径和任何参数。对于状态进程监视器, 如果阈值违反未导致对象状态发生变化, 该命令将不运行。如果字符串为空, 则代理不为该条目执行操作。

注意: 当代理执行操作时, 您可以更改默认设置。有关详细信息, 请参阅 “代理配置” 一章。

'objClass'

指定用于状态管理模型的受监控对象的对象类。定义该属性（以及 objInst 和 objAttr）的值，以创建一个聚合具有相同值的所有条目的受管对象。

默认： MIB 组名称或表名称（请参阅 sysedge.oid）

'objInst'

指定用于状态管理模型的受监控对象的对象实例。定义该属性（以及 objClass 和 objAttr）的值，以创建一个聚合具有相同值的所有条目的受管对象。代理将指定的 regExpr 属性的值作为默认值。

'objAttr'

指定用于状态管理模型的对象属性。定义该属性（以及 objClass 和 objInst）的值，以创建一个聚合具有相同值的所有条目的受管对象。

默认： 进程属性名称

重要级别

指定要用于对象状态管理模型的重要级别。该值定义发生阈值违反时将分配给条目的状态。有效值如下：

- 无
- 正常
- 警告
- 轻微
- 重大
- 严重
- 致命

值“无”创建传统监视器，它将条目排除在对象状态管理模型之外。

默认值： 无

watch ntservice 指令—将服务监控条目添加到进程监视器表

`watch ntservice` 指令可用于将条目添加到进程监视器表中，以便基于 `procAlive` 属性监控 Windows 服务，从而跟踪其运行状态。

使用如下所述的语法向代理数据目录中的 `sysedge.cf` 文件添加一行，保存文件，然后重新启动代理以使更改生效。

使用 `watch ntservice` 指令将 Windows 服务监控条目添加到进程监视器表，如下所示：

```
watch ntservice 'regexpr' index flags interval ['descr' ['action' ['objClass' 'objInst' 'objAttr' severity]]]
```

'regexpr'

指定尝试获得要监控的服务的 ID 时要使用的正则表达式。指定要监控的服务名称的正则表达式。

index

指定要用于该条目的监视器表的行（索引）。表中各行由索引号唯一标识。保留行 1 至 10 以供代理内部使用，因此索引值必须大于 10 且在表中唯一。

flags

使用十六进制标志值（例如，`0x00000001`）指定该条目的任何其他操作说明。

interval

指定应进行监控的时间间隔（秒）。例如，如果值为 30，则指示代理每隔 30 秒对该条目进行监控。该值必须至少为 30，且是 30 秒的倍数。

'descr'

指定表条目的任意说明（长度为 0 至 512 个字符）。

'action'

指定当满足阈值条件且代理发送阈值违反陷阱（传统监视器）或状态更改陷阱（状态监视器）时要运行的命令（长度为 0 到 2048 个字符），包括完整路径和任何参数。对于状态进程监视器，如果阈值违反未导致对象状态发生变化，该命令将不运行。如果字符串为空，则代理不为该条目执行操作。

注意：当代理执行操作时，您可以更改默认设置。有关详细信息，请参阅“代理配置”一章。

'objClass'

指定用于状态管理模型的受监控对象的对象类。定义该属性（以及 objInst 和 objAttr）的值，以创建一个聚合具有相同值的所有条目的受管对象。

默认： processEntry

'objInst'

指定用于状态管理模型的受监控对象的对象实例。定义该属性（以及 objClass 和 objAttr）的值，以创建一个聚合具有相同值的所有条目的受管对象。代理将指定的 regExpr 属性的值作为默认值。

'objAttr'

指定用于状态管理模型的对象属性。定义该属性（以及 objClass 和 objInst）的值，以创建一个聚合具有相同值的所有条目的受管对象。

默认： procAlive

重要级别

指定要用于对象状态管理模型的重要级别。该值定义发生阈值违反时将分配给条目的状态。有效值如下：

- 无
- 正常
- 警告
- 轻微
- 重大
- 严重
- 致命

值“无”创建传统监视器，它将条目排除在对象状态管理模型之外。

默认值： 无

进程监控示例

本节包含用于进程监控的样例配置文件指令。每个示例说明如何定义进程监控的实例，并介绍所监控的属性或阈值。

示例：监控 Sendmail 以确保其正在运行

以下示例配置代理以监控基础系统中的 UNIX 电子邮件服务器上的 sendmail 后台进程：

```
watch process procAlive 'sendmail' 11 0x00000100 60 absolute=4 'Monitor sendmail'  
' 'processEntry' 'sendmail' 'procAlive' fatal
```

11

指示该条目将占用进程监视器表中的第 11 行 (pmonIndex=11)。

0x00000100

指示当存在多个 sendmail 后台进程并且正在运行时，代理应监控父 sendmail 进程。

60

指示代理应每隔 60 秒检查一次 sendmail 进程。

processEntry

指示该条目将 processEntry 对象类用于对象状态模型。该对象类将替换为默认值。在此，“*”是无效值。

sendmail

指示该条目将 sendmail 对象实例用于对象状态模型。该对象实例将替换为默认值。在此，“*”是无效值。

注意：“//”和“/”可以用作分隔符以将实例与系统相关联：

“//hostname/instance”。对于本地系统实例，始终前置“//.”以将实例与本地系统相关联。“//hostname/*”、“//*/instance”和“//*/*”均为无效值。

procAlive

指示该条目将 procAlive 对象属性用于对象状态模型。该属性将替换为默认值。在此，“*”是无效值。

致命

指示该条目将致命重要级别用于对象状态模型。

未指定任何操作，因此代理在发送陷阱时，不会调用命令。

示例：监控简单 TCP/IP 服务进程以确保其正在运行

以下示例配置代理以监控构成简单 TCP/IP 服务的 TCPSVCS 进程：

```
watch process procAlive 'TCPSVCS' 12 0x00000000 30 absolute=4 'Monitor NT TCP
services' '' 'processEntry' 'TCPSVCS' 'procAlive' warning
```

12

指示该条目将占用进程监视器表中的第 12 行 (pmonIndex=12)。

0x00000000

指示代理应提供默认进程监控行为。

30

指示代理应每隔 30 秒检查一次 TCPSVCS 进程。

processEntry

指示该条目将 processEntry 对象类用于对象状态模型。该对象类将替换为默认值。在此，“*”是无效值。

TCPSVCS

指示该条目将 TCPSVCS 对象实例用于对象状态模型。该对象实例将替换为默认值。在此，“*”是无效值。

注意：“//”和“/”可以用作分隔符以将实例与系统相关联：

“//hostname/instance”。对于本地系统实例，始终前置“//.”以将实例与本地系统相关联。“//hostname/*”、“//*/instance”和“//*/*”均为无效值。

procAlive

指示该条目将 procAlive 对象属性用于对象状态模型。该属性将替换为默认值。在此，“*”是无效值。

警告

指示该条目将警告重要级别用于对象状态模型。

未指定任何操作，因此代理在发送陷阱时，不会调用命令。

注意：此示例说明如何监控提供 Windows 简单 TCP/IP 服务的基础进程。以下示例说明如何监控 Windows 服务本身，而不是其基础进程。

示例：监控简单 TCP/IP 服务

以下两个示例配置代理以监控 TCPSVCS 服务本身，而不是基础进程：

```
watch process procAlive 'Simple TCP/IP Services' 13 0x08000 30 absolute=4 'Monitor
NT TCP/IP Services' '' 'processEntry' 'Simple TCP/IP Services' 'procAlive' minor
```

13

指示该条目将占用进程监视器表中的第 13 行 (pmonIndex=13)。

0x08000

指示代理应监控 Windows 服务，而不是基础进程。

30

指示代理应每隔 30 秒检查一次简单 TCP/IP 服务。

processEntry

指示该条目将 processEntry 对象类用于对象状态模型。该对象类将替换为默认值。在此，“*”是无效值。

简单 TCP/IP 服务

指示该条目将简单 TCP/IP 服务对象实例用于对象状态模型。该对象实例将替换为默认值。在此，“*”是无效值。

注意：“//”和“/”可以用作分隔符以将实例与系统相关联：

“//hostname/instance”。对于本地系统实例，始终前置“//.”以将实例与本地系统相关联。“//hostname/*”、“//*/instance”和“//*/*”均为无效值。

procAlive

指示该条目将 procAlive 对象属性用于对象状态模型。该属性将替换为默认值。在此，“*”是无效值。

轻微

指示该条目将轻微重要级别用于对象状态模型。

```
watch ntservice 'Simple TCP/IP Services' 14 0x0 30 'Monitor NT TCP/IP Services'
'' 'processEntry' 'Simple TCP/IP Services' 'procAlive' major
```

14

指示该条目将占用进程监视器表中的第 14 行 (pmonIndex=14)。

30

指示代理应每隔 30 秒检查一次简单 TCP/IP 服务。

processEntry

指示该条目将 **processEntry** 对象类用于对象状态模型。该对象类将替换为默认值。在此，“*”是无效值。

简单 TCP/IP 服务

指示该条目将简单 TCP/IP 服务对象实例用于对象状态模型。该对象实例将替换为默认值。在此，“*”是无效值。

注意：“//”和“/”可以用作分隔符以将实例与系统相关联：

“//hostname/instance”。对于本地系统实例，始终前置“//.”以将实例与本地系统相关联。“//hostname/*”、“//*/instance”和“//*/*”均为无效值。

procAlive

指示该条目将 **procAlive** 对象属性用于对象状态模型。该属性将替换为默认值。在此，“*”是无效值。

重大

指示该条目将重大重要级别用于对象状态模型。

未指定任何操作，因此代理在发送陷阱时，不会调用命令。

示例：监控 ypbind 以确保其正在运行

以下示例配置代理以监控基础系统中的 UNIX ypbind 后台进程：

```
watch process procAlive 'ypbind' 15 0x00000000 60 absolute=4 'Monitor ypbind'
'/example/pager.sh' 'processEntry' 'ypbind' 'procAlive' critical
```

15

指示该条目将占用进程监视器表中的第 15 行 (pmonIndex=15)。

60

指示代理应每隔 60 秒检查一次 ypbind 进程。

processEntry

指示该条目将 **processEntry** 对象类用于对象状态模型。该对象类将替换为默认值。在此，“*”是无效值。

ypbind

指示该条目将 **ypbind** 对象实例用于对象状态模型。该对象实例将替换为默认值。在此，“*”是无效值。

注意：“//”和“/”可以用作分隔符以将实例与系统相关联：

“//hostname/instance”。对于本地系统实例，始终前置“//.”以将实例与本地系统相关联。“//hostname/*”、“//*/instance”和“//*/*”均为无效值。

procAlive

指示该条目将 **procAlive** 对象属性用于对象状态模型。该属性将替换为默认值。在此，“*”是无效值。

严重

指示该条目将严重重要级别用于对象状态模型。

代理在每次发送陷阱时调用指定的操作脚本 `/example/pager.sh`。在这种情况下，它在每次发送 **processStop** 或 **processStart** 陷阱时调用该脚本。该脚本应检查其参数，以确定将要发送的陷阱，并将相应的消息发送到目标寻呼程序。

示例：监控进程的大小

以下示例配置代理以监控特定进程的总体大小：

```
watch process procSize 'netscape' 20 0x00a02400 60 absolute '>' 35000 'Monitor
netscape size' '' 'processEntry' 'netscape' 'procSize' none
```

procSize

指示代理正在监控的属性。它返回相应进程的文本、数据和堆栈段的大小。通过针对给定阈值监控该属性，可以确定该属性是否出现内存泄漏。

netscape

指示代理要监控的进程的名称。

20

指示该条目将占用进程监视器表中的第 20 行 (`pmonIndex=20`)。

0x00a02400

指示代理修改默认进程监视器表行为，如下所示：

0x00000400

指示代理发送 **processClear** 陷阱。

0x00002000

指示代理最多发送 10 个连续的陷阱，然后不再发送。

0x00a00000

包含标志值 10，供此示例中的指令使用。

processEntry

指示该条目将 **processEntry** 对象类用于对象状态模型。该对象类将替换为默认值。在此，“*”是无效值。

netscape

指示该条目将 **netscape** 对象实例用于对象状态模型。该对象实例将替换为默认值。在此，“*”是无效值。

注意：“//”和“/”可以用作分隔符以将实例与系统相关联：

“//hostname/instance”。对于本地系统实例，始终前置“//.”以将实例与本地系统相关联。“//hostname/*”、“//*/instance”和“//*/*”均为无效值。

procAlive

指示该条目将 **procAlive** 对象属性用于对象状态模型。该属性将替换为默认值。在此，“*”是无效值。

none

指示该条目将无重要级别用于对象状态模型。

绝对

指示代理将每个采样值与阈值进行比较，而不是测量连续采样之间的差异（增量）。

>

指示代理将采样的 **procSize** 属性与值 35000（35,000 KB 或 35 MB）进行比较，并且在超过阈值时发送 **processThreshold** 陷阱。

详细信息

[Perl 兼容的正则表达式 \(PCRE\) 支持](#) (p. 145)

[配置文本模式排除](#) (p. 147)

[正则表达式示例](#) (p. 147)

用于进程监控的 edgwatch 命令

edgwatch 进程监控命令及相关参数如下所述：

```
add procAlive processname index flags interval 'descr' 'action' [objClass objInst  
objAttr severity]
```

```
add attribute processname index flags interval sampleType oper value 'description'  
'action' [objClass objInst objAttr severity]
```

```
setStatus index status
```

```
delete index
```

列表

```
dump
```

processname

指定用于查找要监控进程的 PID 的正则表达式。如果该值包含空格或其他特殊字符，请用引号将其括起。

index

指定要用于该条目的监视器表的行（索引）。表中各行由索引号唯一标识。保留行 1 至 10 以供代理内部使用，因此索引值必须大于 10 且在表中唯一。

flags

使用十六进制标志值（例如，0x00000001）指定该条目的任何其他操作说明。

interval

指定应进行监控的时间间隔（秒）。例如，如果值为 30，则指示代理每隔 30 秒对该条目进行监控。该值必须至少为 30，且是 30 秒的倍数。

sampleType

指定采样类型（absolute 或 delta）。该值指示代理是对对象的绝对值进行采样，还是取连续采样之差。

oper

指定用于根据提供的阈值评估当前条目值的布尔运算符。

运算符的有效值如下：

- **nop** (no operation)
- **>** (大于)
- **<** (小于)
- **>=** (大于或等于)
- **<=** (小于或等于)
- **=** (等于)
- **!=** (不等于)
- **默认值：** nop

值

使用特定运算符指定要与所监控属性的当前值比较的整数阈值。

'descr'

指定表条目的任意说明（长度为 0 至 512 个字符）。

'action'

指定当满足阈值条件且代理发送阈值违反陷阱（传统监视器）或状态更改陷阱（状态监视器）时要运行的命令（长度为 0 到 2048 个字符），包括完整路径和任何参数。对于有状态自主监视器，如果阈值违反不导致对象状态更改，则命令无法运行。如果字符串为空，则代理不为该条目执行操作。

注意：当代理执行操作时，您可以更改默认设置。有关详细信息，请参阅“代理配置”一章。

objClass

指定用于状态管理模型的受监控对象的对象类。定义该属性（以及 **objInst** 和 **objAttr**）的值，以创建一个聚合具有相同值的所有条目的受管对象。

默认： processEntry

objInst

指定用于状态管理模型的受监控对象的对象实例。定义该属性（以及 **objClass** 和 **objAttr**）的值，以创建一个聚合具有相同值的所有条目的受管对象。代理将指定的 **regExpr** 属性的值作为默认值。

objAttr

指定用于状态管理模型的对象属性。定义该属性（以及 objClass 和 objInst）的值，以创建一个聚合具有相同值的所有条目的受管对象。

默认：进程属性名称

重要级别

指定要用于对象状态管理模型的重要级别。该值定义发生阈值违反时将分配给条目的状态。有效值如下：

- 无
- 正常
- 警告
- 轻微
- 重大
- 严重
- 致命

值“无”创建传统监视器，它将条目排除在对象状态管理模型之外。

默认值：无

状态

指定条目的状态，可以是下列值之一：

- active（激活行）
- notInService（停用但保留行）
- destroy（删除自主监视器表行）

属性

指定代理针对给定阈值进行监控的进程属性。您可以从[进程属性](#) (p. 241)列表中选择任何进程属性。

edgework 示例

本节包括将 edgework 命令与 SNMP 1、2c 和 3 版结合使用的示例：

示例：监控 ypbind 进程

以下示例在索引 16 创建一个“进程监视器”表条目，用于监控在 SNMPv1 的目标系统上运行的 ypbind 进程：

```
edgework -h 143.45.0.12 -c private -v 1 -o process add procAlive "ypbind" 16 0x00 60 "Monitor ypbind" "/example/pager.sh" "" "" "" minor
```

以下示例为 SNMPv2c 创建同样的配置：

```
edgework -h 143.45.0.12 -c private -v 2c -o process add procAlive "ypbind" 16 0x00 60 "Monitor ypbind" "/example/pager.sh" "" "" "" fatal
```

以下示例为 SNMPv3 创建同样的配置：

```
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" "" "" "" major
```

procAlive

指示要监控的进程属性。它指示代理监控进程，确保其处于运行状态。ypbind 是代理正在监控的进程。它负责客户端目录查找，对于运行网络信息服务 (NIS) 的计算机必不可少。

0x00

指示代理提供该表条目的默认行为。

如果进程中止，代理将发送 processStop 陷阱并运行操作脚本 /example/pager.sh。

示例：监控 firefox 进程

以下示例在索引 20 创建一个“进程监视器”表条目，用于监控在 SNMPv1 的目标计算机上运行的 firefox 进程：

```
edgework -h 143.45.0.12 -c private -v 1 -o process add procSize "firefox" 20 0x00a02400 60 absolute ">" 35000 "Monitor firefox size" "" "" "" "" critical
```

以下示例为 SNMPv2c 创建同样的配置：

```
edgework -h 143.45.0.12 -c private -v 2c -o process add procSize "firefox" 20 0x00a02400 60 absolute ">" 35000 "Monitor firefox size" "" "" "" "" none
```

以下示例为 SNMPv3 创建同样的配置：

```
edgework -h 143.45.0.12 -v 3 -u userName -s 3 -a authPassword -A MD5 -x encryptPassword -X DES -o process add procSize "firefox" 20 0x00a02400 60 absolute ">" 35000 "Monitor firefox size" "" "" "" "" ok
```

procSize

指示要监控的进程属性。它指示代理监控程序的文本、数据和堆栈段的总体大小。

netscape

指示代理正在监控的应用程序。

0x00a02400

指示代理修改默认进程监视器表行为，如下所示：

0x00000400

指示代理发送 processClear 陷阱。

0x00002000

指示代理最多发送 10 个连续的陷阱，然后不再发送。

0x00a00000

包含用于该指令的标志值 10。

>

指示在 netscape 的进程大小超过阈值 (35 MB) 时某个事件应该发生。

35,000 KB 或 35 MB

指示阈值。

示例：监控 Windows TCPSVCS 进程

以下示例在索引 15 创建一个“进程监视器”表条目，用于监控在 SNMPv1 的目标系统上运行的 TCPSVCS 进程（或服务）：

```
edgewatch -h 143.45.0.12 -c private -v 1 -o process add procAlive "TCPSVCS" 15 0x00 30 "Monitor NT TCP services" "" "" "" "" fatal
```

以下示例为 SNMPv2c 创建同样的配置：

```
edgewatch -h 143.45.0.12 -c private -v 2c -o process add procAlive "TCPSVCS" 15 0x00 30 "Monitor NT TCP services" "" "" "" "" ok
```

以下示例为 SNMPv3 创建同样的配置：

```
edgewatch -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" "" "" "" "" warning
```

procAlive

指示要监控的进程属性。它指示代理定期（每 30 秒一次）扫描“进程监视器”表，以验证该进程是否正在运行。

TCPSVCS

指示负责 Windows 系统上 TCP 相关服务的 Windows 服务。

0x00

指示代理提供该表条目的默认行为。

示例：显示某个系统上的所有进程

以下示例显示（转储）在端口 1691 运行 sysedge 的本地主机上的所有进程：

```
edgewatch -c private -p 1691 -v 1 -o process dump
```

删除进程监控条目

要停止监控某个进程，必须从“进程监视器”表和 `sysedge.cf` 文件中删除相应的条目。从 CA Virtual Assurance 用户界面使用点配置删除条目可以同时实现这两个目标。手动删除要求您分别执行操作以从两个位置删除条目。

删除 CA Virtual Assurance 中的进程监视器条目

1. 访问 CA Virtual Assurance 用户界面。
2. 单击“资源”选项卡。
此时将显示“受管资源”窗格和“数据中心”页面。
3. 在“受管资源”窗格中展开“受管”，然后选择代理驻留的服务器。
此时将显示包含服务器详细信息的页面。
4. 单击“配置”，然后单击“进程监视器”。
“进程监视器”表将显示所有的现有自主监视器。
5. 选择要删除的监视器，然后单击“操作”>“删除”。
该条目将从该表中删除。代理执行热启动来应用更改。

手动删除进程监视器条目

1. 从本地 `sysedge.cf` 文件中删除条目的 `watch` 进程指令并保存该文件。
2. 使用 `-o` 参数的下列命令之一运行 `edgewatch` 实用工具：

```
-o delete index  
-o setstatus index 6
```

index

指定要删除的进程监视器条目的索引值。

在使用 `setstatus` 命令时，值为 6 会将行状态设置为销毁并删除该条目。

3. 重新启动代理。
该条目即已删除。

Solaris Zone 进程监控

如果相应地设置标志 0x8000（Solaris Zones 进程）和变量 `pmonRegExpr`，则 SystemEDGE 将监控 Solaris Zones 中的进程。变量 `pmonRegExpr` 指定正则表达式，格式为 *ZoneRegExpr/ProcRegExpr*。由于兼容性原因，变量 `pmonZoneRegExpr` 仍然可用，但只有在未使用 0x8000 标志从 `sysedge.cf` 初始化监视器时，才允许修改。

变量 `pmonProcName` 在所有平台上均可用。如果 Solaris Zones 受监控，该变量的格式将为 *MatchedZoneName/MatchedProcessName*。如果不需要或无法监控 Solaris Zones（非 Solaris 10 OS），则该变量仅包含 *MatchedProcessName*。

如果要监控 Solaris Zones 中的进程，请使用带有标志 0x8000 的 `watch` 进程指令和格式为 *ZoneRegExpr/ProcRegExpr* 的正则表达式。

在升级过程中，所有 `watch` 区域进程指令均转换为新格式。

SystemEDGE 在所有进程监视器相关陷阱中发送 `pmonProcName` 的值。

有关进程和服务监控的建议

配置进程和服务监控时，请考虑以下事项：

- 监控以下内容：
 - 进程的状态：
 - 使用 `processState`（依赖于操作系统）或 `processStateStr`（独立于操作系统）OID 返回进程状态。有关详细信息，请参阅 `empire.asn1`。
 - 对于通过解释器（如 Perl）运行的进程，创建进程监控条目时，请设置 `0x00000800` 标志以匹配进程名和参数（仅限 UNIX）。有关详细信息，请参阅 [“进程监视器表标志”](#) (p. 245)。
 - 超过间隔的 CPU 时间（如果该值未递增，进程可能处于僵停状态）。
 - 总 CPU 时间（值较高时可能表示存在问题）
 - 泄漏内存(随时间推移的 RSS；使用警报阈值向您通知问题)
- 将 `SystemEDGE` 配置为自动重新启动失败的进程。
- 设置 UNIX 应用程序关闭文件，以禁用进程和服务监控，从而防止出现争用情况。
- 使用“进程组监视器”表监控名称相同的多个进程。有关详细信息，请参阅“进程组监控”一章。

第 10 章： 自动监测器

此部分包含以下主题：

[自动监测器的工作方式](#) (p. 275)

[edgewatch 实用工具--使用自动监测器](#) (p. 278)

[指定 sysedge.cf 中的自动监测器](#) (p. 286)

自动监测器的工作方式

自动监测器使用正则表达式作为模式运行定期发现过程，以匹配自动监测器为其创建监视器的资源的名称。自动监测器允许 SystemEDGE 在新资源处于联机状态时自动为其创建监视器。自动监测器在保留的索引范围 (1000000 - 1999999) 内创建监视器。

资源消失时，SystemEDGE 会向 CA Virtual Assurance 发送陷阱，并应用已在自动监测器中配置的“丢失操作”。如果监控的资源丢失，“丢失操作”可删除监视器，或将资源状态设置为特定状态：

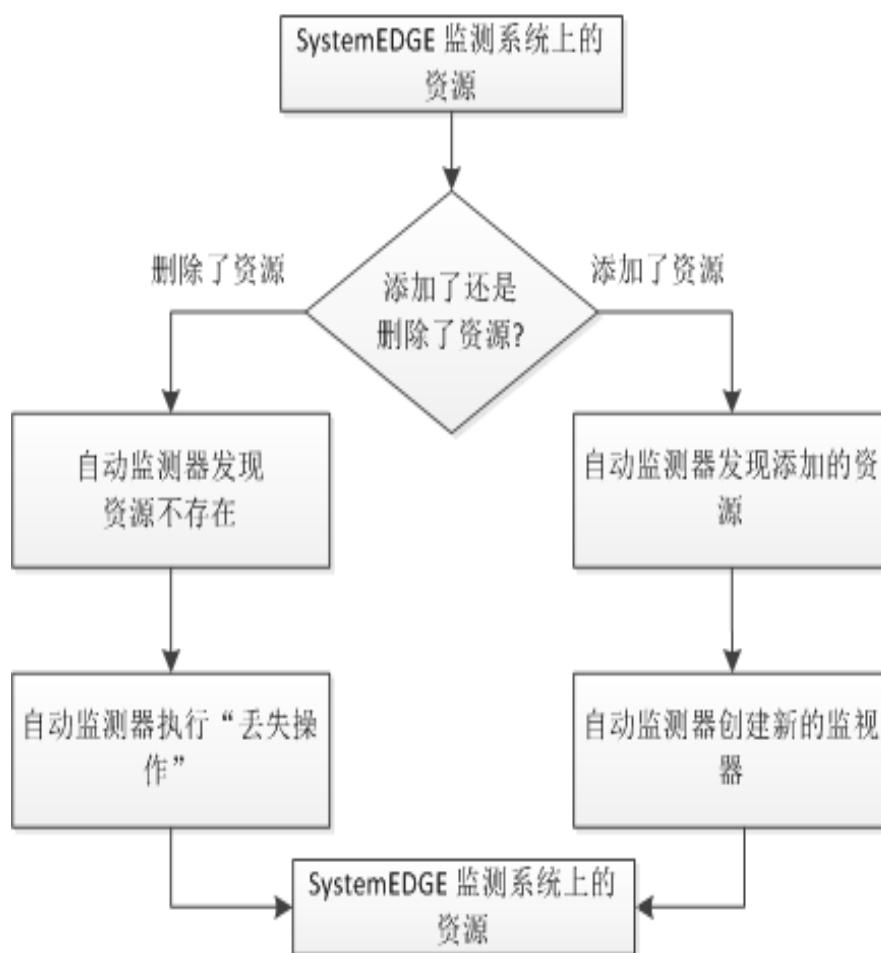
正常、警告、轻微、重大、严重、致命、运行或关闭

通过自动监测器，可在不了解受管系统上所存在资源的情况下创建灵活的策略或分层模板。资源可能是设备、服务或受管系统上运行的进程。

可使用下列自动监测器类型：

- 常规自动监测器—为受管系统上的各种资源（例如设备、接口、文件系统或文件）创建监视器。
- 进程和服务自动监测器—为受管系统上运行的进程和服务创建监视器。

自动检测器的进程 workflow



配置“丢失操作”时，可使用下列指导原则：

- 如果丢失的资源影响系统的运行状况，可配置“丢失操作”以将相应资源的状态更改为严重状态。
- 如果丢失的资源不影响系统的运行状况，可配置“丢失操作”以删除相应的监视器。

详细信息:

[常规自动监测器](#) (p. 277)

[进程和服务自动监测器](#) (p. 277)

常规自动监测器

常规自动监测器可以为受管系统上的各种资源（例如设备、接口、文件系统或文件）创建监视器。

以下列表提供了常规自动监测器的一些示例：

- 所有已发现设备的容量
- 所有已发现磁盘上的磁盘服务时间
- 所有 cmd 进程上的驻留集大小
- 所有隧道网络接口的操作状态
- 所有设备的设备状态

详细信息:

[自动监测器的工作方式](#) (p. 275)

进程和服务自动监测器

使用进程和服务自动监测器可动态地创建进程和服务监视器。

无论何时服务满足自动监测器条件（服务名称、启动类型等等）时，服务自动监测器均会在进程表中创建多个服务监视器。例如，可以监控所有启动类型为“自动”的已安装 SQL 服务。

进程自动监测器采用两种方式创建进程监视器：

- 使用进程名（默认）—进程名与自动监测器条件相匹配时。

例如，当进程符合进程名为“sql”或“svchost”的条件时，将创建进程监视器。自动监测器创建的进程监视器跟踪当前在受管系统上运行的匹配进程，而不管 PID 为何。

 - 自动监测器创建的进程监视器与手动创建的进程监视器具有相同的语义。
 - 可单独监控名称相同但参数不同的一组进程。例如，“java.exe”。
 - 可以为一组相关的进程创建监视器。

- 使用 PID—PID 符合自动监测器条件时。自动监测器允许监视器进程在用户界面中使用 PID 标记，或在 `sysedge.cf` 文件中指定 `watch` 标记 `0x1000`。

每个自动监测器创建的监视器均跟踪进程的所有匹配实例。

- 为特定的进程实例创建监视器。
- 监控不具有区分参数的进程的多个实例。

详细信息：

[自动监测器的工作方式](#) (p. 275)

edgwatch 实用工具--使用自动监测器

`edgwatch` 命令行实用工具自动将 SystemEDGE 代理配置为使用自动监测器进行资源监控。指定特定自动监测器以及关联的参数之后，`edgwatch` 实用工具会发出 SNMP Set 请求，以便在目标代理的 MIB 表中创建适当的条目。`edgwatch` 实用工具位于代理安装的 `bin` 子目录中。

自动监测器的 `edgwatch` 实用工具的用法如下：

```
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 | AES | 3DES]
          [-m FIPS_mode]
          [-r retries]
          [-t timeout] [-d logLevel] [-f logFile]
          -o autowatch command -attribute=value [-attribute=value ...]
```

-h *hostname / ipaddr*

指定要运行代理的系统的主机名或 IP 地址。接受 IPv4 和 IPv6 地址。

默认值： localhost

-p *port*

指定正在运行代理的 UDP 端口（例如，1691）。

默认： 161

-c *community*

指定代理使用的团体字符串。仅对 SNMPv1 和 SNMPv2c 有效。

注意： 为 `snmpset` 指定读/写团体字符串。

默认值： public

-v {1 | 2c | 3}

表示正在运行代理的 SNMP 版本。指定 1 表示 SNMPv1，指定 2c 表示 SNMPv2c，指定 3 表示 SNMPv3。

默认：1

-u *secName*

指定用于 SNMPv3 安全的基于用户的安全模型 (USM) 用户名。

默认值：无

-s *secLevel*

为 SNMPv3 通信指定以下安全级别之一：

- 1 - noAuthNoPriv
- 2 - AuthNoPriv
- 3 - AuthPriv (仅限 SNMPv3)

-n *contextName*

指定 MIBMuxed 代理的实例名称。

默认值：无

-a *authPassword*

指定身份验证密码（如果使用安全级别 2 (AuthNoPriv) 或 3 (AuthPriv) 为 SNMPv3 配置代理）。

注意：对于 SNMPv3 通信，该选项并非必需选项。

默认值：无

-A {MD5 | SHA}

指定要使用的身份验证协议（如果使用安全级别 2 (AuthNoPriv) 或 3 (AuthPriv) 为 SNMPv3 配置代理）。当前仅使用 MD5（消息摘要算法）和 SHA（安全哈希算法）。

默认：MD5

-x *privPassword*

指定隐私（加密）密码（如果使用安全级别 3 (AuthPriv) 为 SNMPv3 配置代理）。

默认值：无

-X {DES | AES | 3DES}

指定隐私协议（如果使用安全级别 3 (AuthPriv) 配置 SNMPv3 用户）。指定 DES（数据加密标准）、AES（使用 128 位密钥的高级加密标准 (AES128)）和 3DES（三重数据加密标准）。

默认值：无

-m *FIPS_mode*

控制操作的 FIPS 模式。可接受的值为 0、1 和 2。

{0}

表示非 FIPS 模式。

1

表示 FIPS 共存模式。

2

表示仅 FIPS 模式。

默认：1

-r *retries*

指定重试次数。

默认：10

-t *timeout*

指定 SNMP 接收器将请求视为超时之前的持续时间。

默认值：10 秒

-d *logLevel*

指定 SNMP 消息的记录级别。接受的值为 0 至 5。

{0}

记录致命消息。

1

记录严重消息。

2

记录警告消息。

3

记录通知消息。

4

记录所有消息。

5

记录所有消息，包括调试消息。

默认：0

-f logfile

指定包含错误和调试信息的日志文件的名称。

默认值: sysedge_utility.log

命令

指定命令和相关属性。支持的命令如下:

- add -index=[watchIndex] [additional attributes ...]
- setstatus -index=[watchIndex] -rowstatus=[active | notInService | destroy]
- delete -index=[watchIndex]
- 列表

注意: 有关命令的详细信息, 请参阅[用于自动监测器的 edgework 命令](#) (p. 281)。

用于自动监测器的 edgework 命令

edgework autowatch 命令及相关参数如下:

```
add -index=[watchIndex] -attribute=value [-attribute=value ...]
setstatus -index=[watchIndex] -rowstatus=[active | notInService | destroy]
delete -index=[watchIndex]
列表
```

所有属性均使用“-attribute=value”格式指定且不相互依赖。

用于添加命令的必需属性:

-name

指定自动监测器条目的名称。

-watchtype

指定以下类型之一: 常规、进程或服务。

-criteria

使用单引号指定主表标准的正则表达式, 以便选择要监控的实例。

-severity

为对象状态模型指定以下重要级别之一: 无、正常、警告、轻微、重大、严重或致命。

用于添加 `-watchtype=generic` 的其他必需属性：

-table

指定监控的表条目（例如，`devTableEntry`）。

-attribute

指定监控的表列属性（例如，`devCapacity`）。

-op

指定比较运算符：`nop`、`gt`、`lt`、`ge`、`le`、`eq` 或 `ne`。

-value

指定比较的阈值。

用于添加 `-watchtype=process` 的其他必需属性：

-attribute

指定以下值之一：`procAlive`、`procMem`、`procSize`、`procRSS`、`procTime`、`procInBlks`、`procOutBlks`、`procMsgsSent`、`procMsgsRecv`、`procNice`、`procNumThreads`、`procNumSwaps`、`procSysCalls`、`procMinorPgFlts`、`procMajorPgFlts`、`procVolCtx`、`procInvolCtx` 或 `procTimePermil`。

-op

指定比较运算符：`nop`、`gt`、`lt`、`ge`、`le`、`eq` 或 `ne`。`procAlive` 不需要上述运算符。

-value

指定比较的阈值。

可选属性：

-interval

指定生成的监视器的评估间隔（默认值：60）

-desc

使用单引号指定应用于所有生成的监视器的说明。

-monflags

指定十六进制整数（例如，0x0000）以应用于所有生成的监视器（默认：0x0000）：

0x0001—不执行操作。

0x0002—不发送陷阱。

0x0004—不重新初始化条目。

0x0008—不发送 syslog 事件。

0x0010—发送连续 notReady 陷阱。

0x0020—不发送 SystemEDGE 参数给操作脚本。

0x0040—不发送此条目的 notReady 陷阱。

-watchtype = 常规：

0x0100—发送清除陷阱。

0x0200—在 X 次违反后发送阈值陷阱。

0x0400—最多发送 X 个连续阈值陷阱 0xffff0000 = X 的值。

-watchtype = 进程或服务：

0x0200—不发送启动陷阱。

0x0400—发送清除陷阱。

0x0800—匹配进程名和参数。

0x1000—在 X 次违反后发送阈值/停止陷阱。

0x2000—最多发送 X 个连续阈值/停止陷阱。

0x8000—监控 Solaris Zone 中的进程。

0xffff0000—X 的值。

-matchtype

为主要选择标准指定以下匹配类型之一：正、负（默认：正）。

-criteria2

在受监控表中为第二选择标准指定一列。

-criteria2regex

使用单引号指定第二选择标准的正则表达式。（默认：'.*'）

-matchtype2

为第二选择标准指定以下匹配类型之一：正、负（默认：正）。

-criteria3

在受监控表中为第三选择标准指定一列。

-criteria3regex

使用单引号指定第三选择标准的正则表达式。（默认：'.*'）

-matchtype3

为第三选择标准指定以下匹配类型之一：正、负（默认：正）。

-stype

指定以下抽样类型之一以应用于所有生成的监视器：**absolute**、**delta**（默认：**absolute**）。

-scale

指定应用于所有生成的监视器的比例值（默认：1）。

-action

使用单引号指定命令（操作）的完整路径（默认：未配置操作）。

-objclass

指定对象状态模型的对象类名称。

-objattr

指定对象状态模型的对象属性名称。

-watchflags

指定十六进制整数（例如，0x0000）以修改自动监测器的行为（默认：0x0000）：

0x0002—不发送自动监测器陷阱。

0x0004—不重新初始化条目。

0x0008—不发送 syslog 事件。

0x0010—发送连续 notReady 自动监测器陷阱。

0x0040—不发送此条目的 notReady 陷阱。

0x1000—通过 PID 监控进程（仅限 -watchtype=process）。

-scaninterval

指定受监控表的发现扫描之间的间隔（默认：300）。

-limit

指定该自动监测器条目的生成的监视器的最大数目（默认：100）。

-loss

生成的监视器无法监控其对象时，请指定以下丢失操作之一：删除、正常、警告、轻微、严重、致命、运行或关闭（默认：致命）。

-starttype

指定 Windows 服务的启动类型。此属性仅应用于 **-watchtype=service**。使用以下值之一：自动、手动、禁用（默认：所有类型）。

setstatus

在使用 **setstatus** 操作时，指定用于设置 MIB 表中行的状态的状态文本约定值。有效值如下所示。值可以是分配的整数值或实际显示的状态文本：

- active(1)
- notInService(2)
- destroy(6)

edgework 自动监测器示例

本节提供的示例介绍如何将 **edgework** 实用工具与 SNMP 版本 1、2c 和 3 结合使用，以便创建自动监测器。

示例：

添加常规自动监测器条目（在表索引 13 中），该条目使用阈值 90 为所有发现的文件系统生成 **devCapacity** 监视器。

```
edgework -c private -o autowatch add -index=13 -watchtype=generic
      -name='All FileSystems - Minor' -table=devTableEntry
      -attribute=devCapacity -interval=120 -criteria='.*' -op=ge
      -value=95 -severity=minor -desc='FileSystem Usage (Percentage)'
      -objclass=FileSystem -objattr=PercentUsed
```

添加进程自动监测器条目（在表索引 15 中），该条目为所有发现的 SQL 进程生成 **procRSS** 监视器，并在进程关闭时自动删除这些监视器：

```
edgework -c private -o autowatch add -index=15 -watchtype=process
      -name='SQL Processes RSS' -criteria='.*sql.*'
      -attribute=procRSS -op=ge -value=100000 -severity=major
      -interval=60 -desc='SQL processes' -objclass='Process'
      -objattr='ResidentSet' -loss=remove
```

添加服务自动监测器条目（在表索引 17 中），该条目为自动启动配置的所有服务生成服务 **procAlive** 观察器：

```
edgewidth -c private -o autowatch add -index 17 -watchtype=service
          -name='All Automatic Services' -criteria='.*' -severity=major
          -interval=300 -limit=200 -desc='Automatic Services'
          -objclass='Service' -objattr='Running' -starttype=automatic
```

列出所有定义的自动监测条目：

```
edgewidth -c private -o autowatch list
```

删除先前指定的服务自动监测器条目：

```
edgewidth -c private -o autowatch delete -index=17
```

将先前指定的进程自动监测器的行状态设置为 **notInService**：

```
edgewidth -c private -o autowatch setstatus -index=15 -rowstatus=notInService
```

指定 sysedge.cf 中的自动监测器

针对与指定的自动监测器标准匹配的资源，自动监测器会在监视器和进程监视器 **MIB** 表中创建条目。自动监测器指令指定适用于自动生成的监视器的标准。

自动监测器的语法基于交换机，但不依赖于位置参数。所有参数均具有以下格式：

```
-parameter=value
-parameter='value'
```

可选参数可以忽略。如果值包含空格，则使用单引号。

自动监测器指令具有以下格式：

```
autowatcher -index=index -watchtype=type -table=table -attribute=attr ...
```

必需参数：

-index

指定用于存储自动监测器的表索引。

-name

指定自动监测器的名称。

-watchtype

指定要创建自动监测器的类型（常规、服务或进程）。

-table

指定监控的表条目（例如，`devTableEntry`）。对于服务或进程自动监测器，并不需要表条目。

-attribute

指定要监控的属性（例如，`devCapacity`）。对于服务自动监测器，并不需要上述属性。

-criteria

指定主要标准的正则表达式，以便选择要监控的实例。

-op

指定比较运算符（`nop`、`gt`、`lt`、`ge`、`le`、`eq`、`ne`）。

-value

指定阈值（但对于进程 `procAlive` 或服务自动监测器并不需要）。

-severity

指定用于对象状态模型的重要级别（无、正常、警告、轻微、重大、严重、致命）。

可选参数：**-interval**

指定此自动监测器生成的所有监视器的评估间隔。

默认值： 60

-desc

指定应用于此自动监测器生成的所有监视器的说明。

-monflags

指定要应用于此自动监测器生成的所有监视器的一组自主监视器标志。

-matchtype

指定主要选择标准的匹配类型（正或负）。

-criteria2

在受监控表中为第二选择标准指定一列。

-criteria2regex

为第二实例选择条件指定正则表达式。

-matchtype2

为第二选择条件指定匹配类型（positive 或 negative）。

-criteria3

为第三选择条件在受监控的表中指定列。

-criteria3regex

为第二实例选择条件指定正则表达式。

-matchtype3

为第二选择条件指定匹配类型（positive 或 negative）。

-stype

指定要应用于每个监视器的抽样类型（absolute、delta）。

-scale

指定收集的监视器值的比例系数。比例系数允许监控采用不同比例单位（变量最初以该单位进行度量）的变量。

使用比例系数 1024，您可以监控以 KB 为单位的变量的值，其最初以字节为单位进行度量。

该比例系数可以 KB(而非字节)为单位监控变量 monCurrVal、monVal、monMinValue 和 monMaxValue。

默认值：1

-action

指定与发送每个监视器的陷阱并发执行的命令的完整路径（包括参数）。

-objclass

指定要用于对象状态模型的对象类（" 将替换为默认值；'*' 是无效值）。

-objattr

指定要用于对象状态模型的对象属性（" 将替换为默认值；'*' 是无效值）。

-watchflags

指定修改自动监测器 行为的标志。

-scaninterval

指定受监控表的两次扫描之间的时间间隔（以秒为单位）。

-limit

指定此自动监测器可以生成的监视器的最大数目。

-loss

指定当监视器不再可访问受监控对象时要采取的丢失操作（删除、正常、警告、轻微、重大、严重、致命、运行、关闭）。

对所有自动监测器有效的标志

000 000 02—不发送任何陷阱，但记录陷阱。

000 000 04—不重新初始化 notReady 自动监测器。

000 000 08—不记录任何陷阱，但发送陷阱。

000 000 10—继续发送 notReady 陷阱。

000 000 40—不发送 notReady 陷阱（包括日志）。

000 000 80—AIM 插件已经创建了该条目。该标志为只读。

详细信息：

[常规自动监测器示例](#) (p. 289)

[服务自动监测器示例](#) (p. 290)

[进程自动监测器示例](#) (p. 291)

常规自动监测器示例

以下示例为 sysedge.cf 文件指定常规自动监测器：

- 监控所有发现的设备的容量。
- 监控所有发现的磁盘的磁盘服务时间。
- 监控所有 cmd 进程的驻留集大小。
- 监控所有隧道网络接口的操作状态。
- 监控所有设备的设备状态。

```
# Device Table - Monitor capacity on all discovered devices
# (minor at 50%, major at 75%)
autowatcher -watchtype=generic -index=11 -table=devTableEntry
-name='Device Table Autowatcher' -criteria='.*' -attribute=devCapacity -op=ge
-value=50 -severity=minor -desc='Device capacity (percentage)' -monflags=0x0
-limit=10 -interval=60 -objclass='Disk' -objattr='Capacity'
autowatcher -watchtype=generic -index=12 -table=devTableEntry
-name='Device Table Autowatcher' -criteria='.*' -attribute=devCapacity -op=ge
-value=75 -severity=major -desc='Device capacity (percentage)' -monflags=0x0
-limit=10 -interval=60 -objclass='Disk' -objattr='Capacity'
```

```
# Disk Statistics Table - Monitor disk service time on all discovered disks
autowatcher -watchtype=generic -index=101 -table=diskStatsEntry
-name='Disk Stats Autowatcher' -criteria='.*' -attribute=diskStatsServiceTime
-op=ge -value=100 -severity=major -desc='Disk service time' -interval=60
-objclass='DiskStats' -objattr='ServiceTime'

# Process Table - Monitor the resident set size on all cmd processes
# lossAction example
autowatcher -watchtype=generic -index=202 -table=processEntry
-name='Process Table Autowatcher' -criteria='cmd' -attribute=processRSS
-op=ge -value=10000 -severity=major -desc='cmd process RSS (KB)' -interval=30
-objclass='Process' -objattr='process RSS' -scaninterval=30

# MIB-2 Interface Table - Monitor the operating status of all tunneled network
# interfaces - multiple criteria example
autowatcher -watchtype=generic -index=303 -table=ifEntry
-name='WiFi Interface Autowatcher' -criteria='.*' -criteria2=ifType
-criteria2regex='131' -attribute=ifOperStatus -op=eq -value=2
-desc='tunnel interface operating status' -severity=critical -objclass=Network
-objattr=OperatingStatus

# Host Resource Device Table - Monitor the device status of all devices
# dual-column instance example
autowatcher -watchtype=generic -index=404 -table=hrDeviceEntry
-name='HR Device Autowatcher' -criteria='.*' -attribute=hrDeviceStatus
-op=ne -value=2 -desc='HR device status' -severity=fatal -objclass=Hardware
-objattr=OperatingStatus
```

服务自动监测器示例

以下示例为 `sysedge.cf` 文件指定服务自动监测器。自动监测器可在匹配指定条件（服务名称、启动类型等等）的进程表中创建多个服务监视器。

- 监控所有安装的启动类型为“自动”的 SQL 服务。

```
autowatch -index=12 -watchtype=service -name='All Automatic SQL Services'
-criteria='.*SQL.*' -starttype=automatic -interval=60 -severity=critical
-desc='SQL services' -objclass='Service' -objattr='Running'
```

进程自动监测器示例

以下示例为 sysedge.cf 文件指定进程自动监测器：

- 确保所有 eTrust 进程的处理器的使用率都不超过 5%。

```
autowatch -index=22 -watchtype=process -name='eTrust Processes'
-criteria='^Ino.*' -attribute=procTimePermil -op=ge -value=50000
-severity=major
-interval=60 -desc='eTrust processes' -objclass='Process' -objattr='CPU_Usage'
```

- 确保所有 SVCHOST 进程使用的物理内存都少于 10 MB。

```
autowatch -index=23 -watchtype=process -name='Svchost Processes'
-criteria='^svchost' -attribute=procRSS -op=ge -value=10000 -severity=major
-interval=60 -monflags=0x0800 -desc='Svchost processes' -objclass='Process'
-objattr='ResidentSet'
```


第 11 章： 进程组监控

本章介绍如何使用 SystemEDGE 代理监控进程组。代理使用进程组将每个进程的信息聚合为易于轮询和监控的单个值。

此部分包含以下主题：

[进程组监控概述](#) (p. 293)

[进程组监视器表](#) (p. 293)

[进程组监控配置](#) (p. 302)

[edgwatch 实用工具--监控进程组](#) (p. 308)

[删除进程组监控条目](#) (p. 313)

[Solaris Zone 进程组监控](#) (p. 314)

进程组监控概述

通过 SystemEDGE 代理，您可以动态监控在基础系统上运行的进程组。可以使用“进程组监视器”表来指定进程组、正则表达式、时间间隔以及重要级别，并且代理可使用该信息来监控与正则表达式匹配的进程组。

SystemEDGE 代理监控进程组，以确定各个组中存在的进程以及组成员身份是否发生改变。如果应用程序的组件启动或失败，或如果成员离开组或被添加到组，则 SystemEDGE 代理可通过陷阱自动通知管理系统。

代理还会使用进程组的组合度量标准值填充“进程组监视器”表的只读列。您可以创建自主监视器条目，以基于已定义的阈值监控这些属性。

注意：有关监控单个进程和 Windows 服务的信息，请参阅“进程和服务监控”一章。

进程组监视器表

“进程组监视器”表提供了有关代理当前正在监控的各个进程组的信息。每一行都表示一个已定义的进程组监视器条目。

对于每个条目，该表提供以下类型的信息：

- 组中进程的列表
- 代理检查组的时间间隔
- 使用组总数的进程属性的值
- 监控重要级别

“进程组监视器”表的各列

以下列表介绍了“进程组监视器”表的各列。有关“进程组监视器”表及其字段的完整说明，请参阅 **Systems Management Empire MIB** 规格（代理安装的 **mib** 子目录中的 **empire.asn1**）。

注意：即使某个进程离开该组之后，**SystemEDGE** 也会维护组成员身份的历史记录，并跟踪进程统计信息。

pgmonIndex

为该条目定义唯一的行索引（1 到 **MAXINT**）。行 1 至 10 保留供代理内部使用，其他行的索引必须在 11 至 **MAXINT** 范围内。

权限：只读

pgmonDescr

定义条目的说明（长度为 0 到 512 个字符）。

权限：读写

pgmonInterval

定义代理应当监控组的频率（以秒为单位）。该值必须至少为 30，且是 30 秒的倍数。

权限：读写

默认：60

pgmonProcRegExpr

定义代理尝试按名称匹配进程时要使用的正则表达式。

权限：读写

pgmonFlags

定义表示其他行为语义的正整数标志，条目在其操作过程中将遵循这些语义。有关可用标志和设置标志的详细信息，请参阅[进程组监视器表标志](#) (p. 299)。

权限：读写

pgmonNumProcs

定义条目正在跟踪的进程组中的当前进程数。如果某个进程的名称（也可能是其参数）与为该条目配置的正则表达式相匹配，则该进程属于该进程组。

权限：只读

pgmonPIDList

定义进程组中的数字 PID。各个 PID 由空格字符分隔。

权限：只读

pgmonStatusList

定义进程组中每个进程的进程状态。各个进程状态由空格字符分隔。状态列表中的条目与 PID 列表中的条目一一对应。有关进程状态的详细信息，请参阅 `processStateStr` MIB 变量。

权限：只读

pgmonAction

使用任何参数定义引用命令（长度为 0 到 2048 个字符），以便在表达式的计算结果为 `True` 且代理发送进程组更改陷阱时运行该命令。如果该字符串为空，代理将不为此条目调用任何操作。

注意：请勿使用 Windows 批处理文件进行操作；它们施加严格的编程限制，并且经常错误地使用桌面应用程序。应使用更强大且灵活的脚本语言，如 Perl 或 Visual Basic。

权限：读写

pgmonNumEvents

定义为条目生成的事件数。事件不一定暗示陷阱；可以通过标志设置关闭针对行的陷阱。

权限：只读

pgmonNumTraps

定义代理针对该条目发送的陷阱数目 (`procGroupChangeTrap`)。

权限：只读

pgmonLastTrap

定义代理针对该条目上次发送陷阱 (`procGroupChangeTrap`) 的时间（基于 `sysUpTime`）。

权限：只读

pgmonRowStatus

定义行状态，行状态可以是以下状态之一：

- 活动
- `notInService`
- `notReady`
- `createAndGo`
- `createAndWait`

通常，行处于 `active` 或 `notInService` 状态。这些值的含义与由 SNMPv2 SMI RowStatus 文本约定定义的值的含义相同。

权限：读写

默认：`createAndWait(5)`

pgmonRSS

定义进程组的驻留集合计大小 (RSS)。该组中每个进程的 RSS 将在每个时间间隔内进行一次合计，并存储在该变量中。因为 RSS 还包括共享内存，所以组的 RSS 总计可能会超过基础系统的可能总物理内存。有关详细信息，请参阅 Systems Management MIBSystems Management Empire MIB 的 processRSS 变量**任务：**只读

pgmonSize

定义进程组的文本、数据和堆栈段的合计大小。该组中每个进程的大小将在每个时间间隔内进行一次合计，并存储在该变量中。因为大小包括共享内存，所以组的总大小可能会超过基础系统的总虚拟内存。有关详细信息，请参阅 Systems Management MIBSystems Management Empire MIB 的 processSize 变量**任务：**只读

pgmonThreadCount

定义进程组的线程总数。在该组的每个进程中运行的线程数将在每个时间间隔内进行一次合计，并存储在该 MIB 变量中。有关详细信息，请参阅 Systems Management Empire MIB 的 processNumThreads 变量（在 empire.asn1 中）。

权限：只读

pgmonMEM

定义该组中进程使用的实际内存的百分比总计。使用的内存百分比将在每个时间间隔内进行一次合计，并存储在该 MIB 变量中。因为内存使用量包括共享内存（共享库和 DLL），所以百分比总计可能超过 100。

权限：只读

pgmonInBlks

定义由该组中进程输入的数据块数。

权限：只读

pgmonOutBlks

定义由该组中进程输出的数据块数。

权限：只读

pgmonMsgsSent

定义由该组中进程发送的消息数。

权限：只读

pgmonMsgsRecv

定义由该组中进程收到的消息数。

权限：只读

pgmonSysCalls

定义由该组中进程调用的系统调用数。

权限：只读

pgmonMinorPgFlts

定义由该组中进程导致的次要页面错误数。

权限：只读

pgmonMajorPgFlts

定义由该组中进程导致的主要页面错误数。

权限：只读

pgmonNumSwaps

定义已交换该组中的进程的次数。

权限：只读

pgmonVolCtx

定义由该组中进程导致的自愿上下文切换数。

权限：只读

pgmonInvolCtx

定义由该组中进程导致的非自愿上下文切换数。

权限：只读

pgmonCPUSecs

定义由该组中进程使用的 CPU 时间（秒数）。

权限：只读

pgmonMatchUser

在设置为有效用户名时，按用户名和任何进程名正则表达式匹配正在运行的进程。该变量仅在 UNIX 系统上有效。

权限：读写

pgmonMatchGroup

在设置为有效组名时，按组名、进程名正则表达式和用户名匹配正在运行的进程。该变量仅在 UNIX 系统上有效。

权限：读写

pgmonProcNameList

受监控进程的进程名列表。在 Solaris 10 或更高版本上，如果已使用区域正则表达式，则名称的格式为 ZoneName/ProcName。

权限：只读

行创建对象

“进程组监视器”表的 **pgmonIndex** 列充当键字段（或行索引），以分辨表中的行。保留 1 到 10 行以供 SystemEDGE 代理内部使用。您可以在 11 到 MAXINT 这个范围内对行进行配置。

可以将以下 MIB 对象与“进程组监视器”表一起使用来优化行创建：

pgmonUnusedIndex

对对象执行 SNMP Get 时，为“进程组监视器”表返回未用过的索引号。查询该对象来获取可用于创建行的索引号。

pgmonMatchDescr

确定与 **pgmonDescr** 属性中的特定条目说明相对应的索引号。使用 **pgmonDescr** 属性值对该 MIB 对象执行 SNMP Set，从而使代理搜索“进程组监视器”表中的条目，并且将上次匹配条目的索引值置于 **monMatchIndex** MIB 对象中。

pgmonMatchIndex

与 **pgmonMatchDescr** 一起使用时，将特定的条目说明与其索引号匹配。

按照本地策略，您可以选择为系统管理预留行块。然后可以在保留的行块中定义条目，而无需担心行可能已经被其他用户条目使用。按照本地的策略，当所有其他用户定义用户配置的条目时，应使用在保留范围以外的行索引。

通过保留行块，您可以定义要跨所有计算机监控的一组一致的条件（行条目），以便同样的条件定义在每台计算机的相同行数中。例如，您可以使用第 11 行 (`monIndex = 11`) 定义用于监控 `swapCapacity` 变量的条目，并可将该配置分发给每个系统，以便第 11 行对每个系统中的 `swapCapacity` 变量进行监控。

在 CA Virtual Assurance 上配置代理可以自动执行该过程。您可以使用特定行号配置条目块，并将该配置部署到一个操作中的所有受管系统。此外，CA Virtual Assurance 将检测您是否定义已在使用中的行并禁止该操作。

有关在 CA Virtual Assurance 中定义行的详细信息，请参阅 CA Virtual Assurance 文档。

进程组监视器表标志

“进程组监视器”表的 `pgmonFlags` 列是一个 32 位无符号整数字段，可以为相应行指定其他行为语义。

默认情况下，“进程组监视器”表的行执行以下操作：

- 尝试对自身进行重新初始化
- 发送 SNMP 陷阱
- 将事件记录到普通日志文件
- 调用操作（如果进行了配置）

您可以设置不同标志位来改变这些默认值。代理将以十六进制（基础 16）码解释所有的标志。

进程组监视器表标志 (`pgmonFlags`) 如下所示：

标志值由三个字段组成：

- 字段 1 包含为 Systems Management Empire MIB 的监控表定义的常用表标志。该部分是标志的低 8 位。
- 字段 2 包含分别为每个监控表定义的表特定的标志。该字段定义常用表标志后面接下来的 12 个低位。
- 字段 3 包含为整数值保留的 12 个高位，以便用于表特定的标志。该字段定义了特定于“进程组监视器”表的标志。

以下部分解释了每个标志位。您可以通过逻辑“OR”运算组合标志值。一个标志特定于“进程组监视器”表：**0x00100**。该标志指示 SystemEDGE 代理与该条目的进程名和参数相匹配。

以下列表描述进程监视器表标志：

0x00000001

禁止执行该条目的操作。

0x00000002

禁止发送该条目的 SNMP 陷阱。

0x00000004

禁止尝试重新初始化该条目。默认情况下，如果目标进程已重新启动，代理将定期尝试通过扫描进程表对条目进行重新初始化，以确定新进程的 ID。

0x00000008

禁止在 `sysedge.log` 文件中记录该条目的陷阱。设置该位对发送陷阱没有影响。在事件频繁发生或将特定条目用作代理监控信号时，禁用事件记录会很有用。

0x00000020

禁止将默认参数传递给操作脚本或程序。SystemEDGE 通常传递默认操作参数（表示陷阱类型、说明字段等）。有关操作参数的详细信息，请参阅[进程组监视器表操作参数](#) (p. 301)。

0x00000040

禁止发送该条目的 `notReady` 陷阱。这包括不记录以及不执行 `notReady` 陷阱的操作。

0x00000100

匹配该条目的进程名和参数。

0x00008000

匹配 Solaris Zone 中的进程。`pgmonProcRegExpr` 属性的格式应为 `ZoneRegExpr/ProcRegExpr`。

进程组监视器表操作参数

SystemEDGE 代理可在操作命令被调用时向其提供若干默认参数。这些参数是对您在操作字符串中指定的任何参数的补充，并在您指定的那些参数之后通过命令行传递。默认参数与针对“进程组监视器”表发送的 SNMP 陷阱中提供的参数相同。

注意：通过在表条目中设置 0x00000020 标志，可以防止代理在命令中包含这些操作参数。

以下列表介绍了“进程组监视器”表操作的默认参数：

pgmonIndex

定义表条目的索引。

pgmonDescr

定义表条目的说明。

pgmonFlags

定义与表条目关联的标志。传递给操作脚本的值采用以 16 为基数（十六进制）的表示法，前面的 0x 表示它是十六进制数（例如，0x00000020）。

pgmonNumProcs

定义条目正在监控的进程数。

pgmonRegExpr

定义正则表达式，以供代理用于查找要作为组监控的进程的进程 ID。

pgmonRowStatus

定义表条目的行状态。

pgmonPIDList

定义当前正受监控的进程的进程 ID (PID) 列表。

pgmonStatusList

定义当前正受监控的每个进程的状态。

pgmonSeverity

定义分配给表条目的重要级别。

有关 SystemEDGE 代理发送的陷阱的详细信息，请参阅“概念”一章和附录“私人企业陷阱”。

查看进程组监视器表

可以使用 CA Virtual Assurance 查看受管代理的“进程组监视器”表的内容。

在 CA Virtual Assurance 中查看“进程组监视器”表

1. 访问 CA Virtual Assurance 用户界面。
2. 单击“资源”选项卡。
此时将显示“受管资源”窗格和“数据中心”页面。
3. 在“受管资源”窗格中展开“受管”，然后选择代理驻留的服务器。
此时将显示包含服务器详细信息的页面。
4. 单击“配置”，然后单击“进程组监视器”。
此时将显示“进程组监视器”表，其中包含所有的现有进程组监视器。

进程组监控配置

可以通过添加、删除或修改“进程组监视器”表中的条目来控制 SystemEDGE 代理监控的进程和进程属性。

可通过以下一种方法配置“进程组监视器”表：

动态使用 CA Virtual Assurance 中基于文件的配置

CA Virtual Assurance 可以配置安装在它从 CA Virtual Assurance 用户界面管理的任何系统上的代理。建议使用该配置方法，因为该方法可以利用一种用户友好且安全的机制动态地定义和传送监视器，该机制不需要代理重新启动。您可以为单个代理配置监控，也可以定义监控策略以在多个受管系统上应用。当您执行单个配置更改或应用配置策略时，CA Virtual Assurance 会将新的配置文件传送给代理系统。CA Virtual Assurance 可以在整个数据中心中充当代理配置的中心点，您可以覆盖或拒绝通过其他方式执行的更改。

有关使用 CA Virtual Assurance 配置代理的详细信息，请参阅 CA Virtual Assurance 文档。

从管理系统（例如 eHealth AdvantEDGE View、CA Spectrum 或 CA NSM）动态使用 SNMP Set 命令

各种管理系统均支持通过 SNMP Set 命令将监视器添加到 SystemEDGE 中。eHealth 通过 AdvantEDGE View 接口提供该功能，CA NSM 通过 Agent View 启用配置，CA Spectrum 通过 OneClick 启用配置。通过 CA Virtual Assurance 进行动态配置所提供的所有功能都无法通过这些方法获得。

有关 SNMP Set 配置的详细信息，请参阅相应管理系统的文档。

通过在本本地 `sysedge.cf` 文件中指定自主监视器表的条目来执行手动配置

手动配置要求您使用特定指令将条目添加到本地 `sysedge.cf` 文件中。当从 `sysedge.cf` 文件直接配置代理时，必须执行完全重新启动才能使更改生效。请使用位于代理数据目录中的 `sysedge.cf` 文件进行直接配置。

本节介绍如何在本地 `sysedge.cf` 文件中直接配置自主监控，但介绍的属性是所有形式的配置所必需的。

watch procgroup 指令--将条目添加到“进程组监视器”表中

通过 `watch procgroup` 指令，您可以直接在 `sysedge.cf` 文件中将进程组监视器条目添加到“进程组监视器”表中。参数表示“进程组监视器”表中的列。

使用如下所述的语法向代理数据目录中的 `sysedge.cf` 文件添加一行，保存文件，然后重新启动代理以使更改生效。

使用 `watch procgroup` 指令将条目添加到“进程组监视器”表中，如下所示：

```
watch procgroup 'regexpr' index flags interval ['descr'] ['action'] [severity]
'regexpr'
```

指定尝试获取要监控的组的进程 ID 时所使用的正则表达式。

index

指定要用于该条目的监视器表的行（索引）。表中各行由索引号唯一标识。保留行 1 至 10 以供代理内部使用，因此索引值必须大于 10 且在表中唯一。

flags

使用十六进制标志值（例如，0x00000001）指定该条目的任何其他操作说明。有关可用标志的详细信息，请参阅[进程组监视器表标志](#) (p. 299)。

interval

指定应进行监控的时间间隔（秒）。例如，如果值为 30，则指示代理每隔 30 秒对该条目进行监控。该值必须至少为 30，且是 30 秒的倍数。

'descr'

指定表条目的任意说明（长度为 0 至 512 个字符）。

'action'

指定找到进程组的匹配项时要运行的命令（长度为 0 到 2048 个字符），其中包括完整路径和所有参数。如果字符串为空，则代理不为该条目执行操作。

注意：当代理执行操作时，您可以更改默认设置。有关详细信息，请参阅“代理配置”一章。

重要级别

指定在组改变并发送陷阱时要分配给条目的重要级别。陷阱中包含重要级别。有效值如下：

- 无
- 正常
- 警告
- 轻微
- 重大
- 严重
- 致命

请注意，重要级别仅指定监视器的重要性，并不用于计算状态。

默认值：无

watch procgroupex 指令--将条目添加到“进程组监视器”表中（仅 UNIX）

通过 `watch procgroupex` 指令，您可以直接在 `sysedge.cf` 文件中将进程组监视器条目添加到“进程组监视器”表中。参数表示“进程组监视器”表中的列。

使用如下所述的语法向代理数据目录中的 `sysedge.cf` 文件添加一行，保存文件，然后重新启动代理以使更改生效。

使用 `watch procgroupex` 指令将条目添加到“进程组监视器”表中，如下所示：

```
watch procgroupex 'regexpr' user group index flags interval ['descr'] ['action']  
[severity]
```

'regexpr'

指定尝试获取要监控的组的进程 ID 时所使用的正则表达式。

user

指定除了任何进程名称正则表达式之外还存在匹配的用户名（特定于 UNIX）。

组

指定除了任何进程名称正则表达式之外还存在匹配的组名（特定于 UNIX）。

index

指定要用于该条目的监视器表的行（索引）。表中各行由索引号唯一标识。保留行 1 至 10 以供代理内部使用，因此索引值必须大于 10 且在表中唯一。

flags

使用十六进制标志值（例如，`0x00000001`）指定该条目的任何其他操作说明。有关可用标志的详细信息，请参阅[进程组监视器表标志](#)（p. 299）。

interval

指定应进行监控的时间间隔（秒）。例如，如果值为 30，则指示代理每隔 30 秒对该条目进行监控。该值必须至少为 30，且是 30 秒的倍数。

'descr'

指定表条目的任意说明（长度为 0 至 512 个字符）。

'action'

指定找到进程组的匹配项时要运行的命令（长度为 0 到 2048 个字符），其中包括完整路径和所有参数。如果字符串为空，则代理不为该条目执行操作。

注意：当代理执行操作时，您可以更改默认设置。有关详细信息，请参阅“代理配置”一章。

重要级别

指定在组改变并发送陷阱时要分配给条目的重要级别。陷阱中包含重要级别。有效值如下：

- 无
- 正常
- 警告
- 轻微
- 重大
- 严重
- 致命

请注意，重要级别仅指定监视器的重要性，并不用于计算状态。

默认值：无

进程组监控示例

本节包含进程组监控的配置文件指令示例。可以使用 `watch procgroup` 在 `sysedge.cf` 文件中添加这些条目。

示例：监控 `xterm` 进程组

以下示例将代理配置为监控 `xterm` 进程组：

```
watch procgroup 'xterm.*' 101 0x00 60 'Watch xterms' '' warning
```

101

表示该条目将占用“进程组监视器”表中的第 101 行 (pgmonIndex=101)。

0x00

表示默认行为。

60

表示代理应每 60 秒检查一次 `xterm` 进程组。

警告

表示进程组更改陷阱将包含警告重要级别。

因为没有指定操作，所以代理在发送陷阱时不会调用任何命令。

示例：监控 `emacs` 进程组

以下示例将代理配置为监控 `emacs` 进程组：

```
watch procgroup 'emacs.*|xmibmgr' 12 0x00 60 'Watch emacs' '' major
```

12

表示该条目将占用“进程组监视器”表中的第 12 行 (pgmonIndex=12)。

0x00

表示默认行为。

60

表示代理应每 60 秒检查一次 `emacs` 进程组。

重大

表示进程组更改陷阱将包含重大重要级别。

因为没有指定操作，所以代理在发送陷阱时不会调用任何命令。

示例：监控 DT 进程组

以下示例将代理配置为监控 DT 进程组：

```
watch procgroup 'dt/bin' 13 0x00 60 'Watch DT stuff' ''
```

13

表示该条目将占用“进程组监视器”表中的第 13 行 (pgmonIndex=13)。

0x00

表示默认行为。

60

表示代理应每 60 秒检查一次 DT 进程组。

因为没有指定操作，所以代理在发送陷阱时不会调用任何命令。

详细信息

[Perl 兼容的正则表达式 \(PCRE\) 支持](#) (p. 145)

[配置文本模式排除](#) (p. 147)

[正则表达式示例](#) (p. 147)

edgework 实用工具--监控进程组

edgework 命令行实用工具可自动配置 SystemEDGE 代理，以监控进程、进程组、日志文件和 Windows 事件日志。指定特定进程、进程组、日志文件或 Windows 事件日志以及关联的参数之后，edgework 实用工具会发出 SNMP Set 请求，以便在目标代理的相应监控表中创建适当的条目。edgework 实用工具位于代理安装的 bin 子目录中。

使用 edgework 实用工具进行进程组监控，如下所示：

```
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 procgroup command
```

-h *hostname* / *ipaddr*

指定要运行代理的系统的主机名或 IP 地址。接受 IPv4 和 IPv6 地址。

默认值: localhost

-p *port*

指定正在运行代理的 UDP 端口（例如，1691）。

默认: 161

-c *community*

指定 edgework 在发送给代理的 SNMP 请求中使用的团体字符串。对于 SNMPv1 和 SNMPv2c 有效。

默认值: public

-v {1 | 2c | 3}

表示正在运行代理的 SNMP 版本。指定 1 表示 SNMPv1，指定 2c 表示 SNMPv2c，指定 3 表示 SNMPv3。

默认: 1

-u *secName*

指定用于 SNMPv3 安全的基于用户的安全模型 (USM) 用户名。

默认值: 无

-s *secLevel*

为 SNMPv3 通信指定以下安全级别之一：

- 1 - noAuthNoPriv
- 2 - AuthNoPriv

3—AuthPriv（仅限 SNMPv3）

-n *contextName*

指定 MIBMuxed 代理的实例名称。

默认值: 无

-a *authPassword*

指定身份验证密码（如果使用安全级别 2 (AuthNoPriv) 或 3 (AuthPriv) 为 SNMPv3 配置代理）。

注意: 对于 SNMPv3 通信，该选项并非必需选项。

默认值: 无

-A {MD5 | SHA}

指定要使用的身份验证协议（如果使用安全级别 2 (AuthNoPriv) 或 3 (AuthPriv) 为 SNMPv3 配置代理）。当前仅使用 MD5（消息摘要算法）和 SHA（安全哈希算法）。

默认: MD5

-x *privPassword*

指定隐私（加密）密码（如果使用安全级别 3 (AuthPriv) 为 SNMPv3 配置代理）。

默认值: 无

-X {DES | AES | 3DES}

指定隐私协议（如果使用安全级别 3 (AuthPriv) 配置 SNMPv3 用户）。指定 DES（数据加密标准）、AES（使用 128 位密钥的高级加密标准 (AES128)）和 3DES（三重数据加密标准）。

默认值: 无

-m *FIPS_mode*

控制操作的 FIPS 模式。可接受的值为 0、1 和 2。

{0}

表示非 FIPS 模式。

1

表示 FIPS 共存模式。

2

表示仅 FIPS 模式。

默认: 1

-r *retries*

指定重试次数。

默认: 10

-t *timeout*

指定 SNMP 接收器将请求视为超时之前的持续时间。

默认值: 10 秒

-d logLevel

指定 SNMP 消息的记录级别。接受的值为 0 至 5。

{0}

记录致命消息。

1

记录严重消息。

2

记录警告消息。

3

记录通知消息。

4

记录所有消息。

5

记录所有消息，包括调试消息。

默认：0

-f logfile

指定包含错误和调试信息的日志文件的名称。

默认值：sysedge_utility.log

命令

指定命令和相关参数。支持的命令包括：

- add
- setstatus
- 删除
- 列表

注意：有关这些命令的详细信息，请参阅“用于进程组监控的 edgework 命令”。

用于进程组监控的 edgwatch 命令

用于进程组监控的 edgwatch 命令和相关参数如下所示：

```
add attribute regexpr index flags interval 'description' 'action'
```

```
setstatus index status
```

```
delete index
```

列表

注意：以粗体显示的参数是必须完全按显示内容键入的字符串文本。其他参数是必须输入的值。

regexpr

指定用于查找要监控进程的 PID 的正则表达式。如果该值包含空格或其他特殊字符，请用引号将其括起。

index

指定要用于该条目的监视器表的行（索引）。表中各行由索引号唯一标识。保留行 1 至 10 以供代理内部使用，因此索引值必须大于 10 且在表中唯一。

flags

使用十六进制标志值（例如，0x00000001）指定该条目的任何其他操作说明。有关可用标志的列表，请参阅[进程组监视器表标志](#) (p. 299)。

interval

指定应进行监控的时间间隔（秒）。例如，如果值为 30，则指示代理每隔 30 秒对该条目进行监控。该值必须至少为 30，且是 30 秒的倍数。

'descr'

指定表条目的任意说明（长度为 0 至 512 个字符）。

'action'

指定找到进程组的匹配项时要运行的命令（长度为 0 到 2048 个字符），其中包括完整路径和所有参数。如果字符串为空，则代理不为该条目执行操作。

注意：当代理执行操作时，您可以更改默认设置。有关详细信息，请参阅“代理配置”一章。

状态

指定条目的状态，可以是下列值之一：

- active（激活行）
- notInService（停用但保留行）
- destroy（删除行）

edgwatch 示例

以下命令定义了索引为 10、11、12 和 13 的四个监视器。索引为 10 和 11 的监视器未指定对象信息。索引为 12 和 13 的监视器已指定对象信息（类、实例、属性）。

```
edgwatch -c admin -o process add procAlive "lsold1" 10 0x00 60 "monitor oldls2" ""
edgwatch -c admin -o process add procSize "lsold2" 11 0x00 60 absolute "!=" 100 "monitor oldls1" ""
edgwatch -c admin -o process add procAlive "ls" 12 0x00 60 "monitor ls1" ""
testClass testInst testAttr none
edgwatch -c admin -o process add procSize "lsold2" 13 0x00 60 absolute "!=" 100 "monitor oldls1" "" testClass testInst testAttr none
edgwatch -c admin -o process list
```

删除进程组监控条目

要停止监控进程组，必须从“进程组监视器”表和 `sysedge.cf` 文件中删除相应的条目。从 CA Virtual Assurance 用户界面使用点配置删除条目可以同时实现这两个目标。手动删除要求您分别执行操作以从两个位置删除条目。

删除 CA Virtual Assurance 中的进程组监视器条目

1. 访问 CA Virtual Assurance 用户界面。
2. 单击“资源”选项卡。
此时将显示“受管资源”窗格和“数据中心”页面。
3. 在“受管资源”窗格中展开“受管”，然后选择代理驻留的服务器。
此时将显示包含服务器详细信息的页面。
4. 单击“配置”，然后单击“进程组监视器”。
此时将显示“进程组监视器”表，其中包含所有的现有自主监视器。
5. 选择要删除的监视器，然后单击“操作”>“删除”。
该条目将从该表中删除。代理执行热启动来应用更改。

手动删除进程组监视器条目

1. 从本地 `sysedge.cf` 文件中删除针对条目的 `watch procgroup` 指令，然后保存文件。

2. 使用 `-o` 参数的下列命令之一运行 `edgewatch` 实用工具：

```
-o delete index  
-o setstatus index 6
```

index

指定要删除的进程组监视器条目的索引值。

在使用 `setstatus` 命令时，值为 6 会将行状态设置为销毁并删除该条目。

3. 重新启动代理。

该条目即已删除。

Solaris Zone 进程组监控

如果相应设置了标志 `0x8000`（Solaris Zone 进程）和变量 `pgmonRegExpr`，则 `SystemEDGE` 会监控 Solaris Zone 中的进程组。变量 `pgmonRegExpr` 以格式 `ZoneRegExpr/ProcRegExpr` 指定了正则表达式。出于兼容性考虑，变量 `pgmonZoneRegExpr` 仍然可用，但是，仅当不从标志为 `0x8000` 的 `sysedge.cf` 初始化监视器时，才允许进行修改。

变量 `pgmonProcNameList` 在所有平台上均可用。如果监控 Solaris Zone，则该变量将包含一个由空格分隔的值的列表，值格式为 `MatchedZoneName/MatchedProcessName`。如果不必或不可能监控 Solaris Zone（非 Solaris 10 操作系统），则该变量将包含一个由空格分隔的匹配进程名的列表。

在希望监控 Solaris Zone 中的进程组时，请使用标志为 `0x8000` 的 `watch process` 指令和格式为 `ZoneRegExpr/ProcRegExpr` 的正则表达式。

在升级过程中，所有的 `watch zone procgroup` 指令都将转换为新格式。

`SystemEDGE` 在与进程监视器相关的所有陷阱中发送 `pgmonProcNameList` 的值。

第 12 章： 日志文件监控

本章介绍如何使用 SystemEDGE 代理来监控日志文件的正则表达式和目录的大小及内容。

此部分包含以下主题：

[日志文件监控概述](#) (p. 315)

[目录监控](#) (p. 316)

[日志监视器表](#) (p. 316)

[日志监控配置](#) (p. 325)

[edgemonitor 实用工具—监控日志文件](#) (p. 329)

[删除日志监控条目](#) (p. 335)

[日志文件监控建议](#) (p. 336)

日志文件监控概述

通过 SystemEDGE 代理，您可以连续监控采用 UTF-8 编码的文本文件中是否存在用户指定的正则表达式。日志文件监控提供了一种灵活的解决方案，用于通过监控应用程序记录的消息来监控应用程序。该功能还有助于安全管理；例如，您可以配置代理来监控系统日志文件，以获取用于通知您可能存在的安全违规的消息。使用“日志监视器”表可指定要监控的文件、要匹配的正则表达式、间隔、操作、重要级别及其他值。代理会自动监控定义的日志文件，并在检测到正则表达式匹配项时向管理系统发送陷阱。

日志文件规格可以是通配符表达式，这将导致代理监控最近更新的匹配该表达式的单个日志文件。

在代理启动时（或在向“日志监视器”表添加多行之后），代理会评估日志文件表达式，识别最近更新的日志文件的当前长度和上次访问时间。之后，代理将定期统计与日志文件表达式匹配的各个日志文件自上次状态检查以来新增或修改的内容。该行为允许单个监视器条目跟踪更改名称（也可能包括日期或修订信息）的日志文件，而无需手动修改给定的条目。

如果受监控的日志文件已经更改，代理将仅扫描更改内容而不是整个日志文件，以查看是否存在指定正则表达式的匹配项。如果代理找到匹配项，您可以对其配置以发送企业特定的 logMonMatch SNMP 陷阱，并且针对该行运行指定的操作（前提是操作字段不为空）。

目录监控

您可以使用“日志监视器”表来监控文件目录的内容和大小。可以指定目录路径和名称，代理将跟踪该目录的大小和该目录中文件的数量。然后，您可以针对这些属性定义自主监视器，从而设置关于目录大小和内容的阈值。

有关详细信息，请参阅[日志监视器表列](#) (p. 316)和[watch logfile 指令](#) (p. 326)。

日志监视器表

“日志监视器”表提供 SystemEDGE 代理当前正在监控的各个日志文件或目录的相关信息。每行代表一个已定义的日志监视器条目。

对于每个条目，“日志监视器”表提供以下类型的信息：

- 代理正在监控以获取特定正则表达式的日志文件的名称
- 日志文件匹配的正则表达式
- 由于找到匹配项已发送陷阱的次数
- 上次发送陷阱的时间
- 导致上次匹配的日志条目
- 重要级别

日志监视器表列

以下列表介绍“日志监视器”表中的各列。有关“日志监视器”表的详细信息，请参阅“Systems Management Empire MIB empire.asn1”，位于代理安装的 `mib` 子目录中。

logMonitorIndex

为该条目定义唯一的行索引（1 到 MAXINT）。行 1 至 10 保留供代理内部使用，其他行的索引必须在 11 至 MAXINT 范围内。

权限：只读

logMonitorLogFile

定义要监控的日志文件或目录的完整路径及文件名。该值可以是通配符表达式，代理会在每次扫描时评估该值以识别最近更新的日志文件。

权限：读写

logMonitorRegularExpression

定义扫描日志文件以查找匹配项时要搜索的正则表达式。
SystemEDGE 支持最多为 512 个字符的正则表达式。

权限：读写

logMonitorNumberTraps

定义由于将与正则表达式匹配的字符串记录到文件中而导致代理发送陷阱 (logMonMatchTrap) 的次数。

权限：只读

logMonitorLastTrap

定义代理上次针对该条目发送陷阱 (logMonMatchTrap) 的时间（基于 sysUpTime）。

权限：只读

logMonitorLastMatch

定义上一个与正则表达式匹配的日志文件条目。该变量会在每次出现匹配项时更新。

权限：读写

logMonitorStatus

定义行状态，行状态可以是以下状态之一：

- 活动
- notInService
- notReady
- createAndGo
- createAndWait

通常，行处于 active 或 notInService 状态。这些值的含义与由 SNMPv2 SMI RowStatus 文本约定定义的值的含义相同。

权限：读写

默认：createAndWait(5)

logMonitorLogFileSize

定义受监控文件的当前大小（字节）或受监控目录的当前大小（千字节）。

权限：只读

logMonitorLogFileLastUpdate

定义文件或目录上次更新的时间。

权限：只读

logMonitorDescr

定义受监控文件或目录的说明（长度介于 0 至 512 个字符）。

权限：读写

logMonitorAction

定义带有任意参数的引用命令（长度介于 0 至 2048 个字符），用于在匹配正则表达式并且代理发送陷阱时运行。如果该字符串为空，代理将不为此条目调用任何操作。

注意：请勿使用 Windows 批处理文件进行操作；它们施加严格的编程限制，并且经常错误地使用桌面应用程序。应使用更强大且灵活的脚本语言，如 Perl 或 Visual Basic。

权限：读写

logMonitorFlags

定义表示其他行为语义的正整数标志，条目在其操作过程中将遵循这些语义。有关可用标志和设置标志的详细信息，请参阅[日志监视器表标志](#) (p. 321)。

权限：读写

logMonitorMatches

定义与正则表达式匹配的日志文件条目的数量。无论是否针对该条目发送陷阱，该值均会递增。通过轮询该变量，您可以确定随着时间的推移匹配项的比率或数量。

权限：只读

logMonitorInterval

定义连续扫描日志文件或目录的最有效时间间隔，以分钟为单位。

权限：读写

logMonitorLogFileName

定义要监控的实际文件名或目录名，以防 logMonitorLogFile 属性包含通配符。

权限：只读

logMonitorLogFileCount

定义受监控目录中的文件和子目录的数量。监控日志文件时，该值为 1 或 0，具体取决于文件是否存在。

权限：只读

logMonitorSeverity

定义发送陷阱时分配给条目的重要级别。有效值如下：

- none
- 确定
- 警告
- 轻微
- 重大
- 严重
- 致命

默认值：无

行创建对象

“日志监视器”表的 `logMonIndex` 列充当关键字段（或行索引），用于区分表中的行。保留 1 到 10 行以供 `SystemEDGE` 代理内部使用。您可以在 11 至 `MAXINT` 范围内配置行。可以将以下 MIB 对象和“日志监视器”表结合使用来优化行创建：

logmonUnusedIndex

针对对象执行 `SNMP Get` 时，将返回“日志监视器”表中未使用的索引号。查询该对象来获取可用于创建行的索引号。

logmonMatchDescr

确定对应于 `logmonDescr` 属性中特定条目说明的索引号。针对此具有 `logmonDescr` 属性值的 MIB 对象执行 `SNMP Set`，使代理搜索“日志监视器”表中的所有条目，并将最后一个匹配条目的索引值设置为 `monMatchIndex` MIB 对象的索引值。

logmonMatchIndex

与 `logmonMatchDescr` 结合使用时，使用索引号匹配特定条目说明。

按照本地策略，您可以选择为系统管理预留行块。然后可以在保留的行块中定义条目，而无需担心行可能已经被其他用户条目使用。按照本地的策略，当所有其他用户定义用户配置的条目时，应使用在保留范围以外的行索引。

通过保留行块，您可以定义要跨所有计算机监控的一组一致的条件（行条目），以便同样的条件定义在每台计算机的相同行数中。

在 `CA Virtual Assurance` 上配置代理可以自动执行该过程。您可以使用特定行号配置条目块，并将该配置部署到一个操作中的所有受管系统。此外，`CA Virtual Assurance` 将检测您是否定义已在使用中的行并禁止该操作。

有关在 `CA Virtual Assurance` 中定义行的详细信息，请参阅 `CA Virtual Assurance` 文档。

日志监视器表标志

“日志监视器”表的 `logMonitorFlags` 列是 32 位无符号整数，可指定“日志监视器”表中相应行的其他行为。默认情况下，“日志监视器”表中的行执行以下操作：

- 尝试对自身进行重新初始化
- 发送 SNMP 陷阱
- 记录常见日志文件的匹配项
- 调用操作（如果进行了配置）

您可以指定不同的标志位来更改这些默认值。SystemEDGE 代理以十六进制（基数为 16）表示法解释所有标志。

日志监视器表标志 (`logMonitorFlags`) 如下所示：

标志值由三个字段组成：

- 字段 1 包含为 Systems Management Empire MIB 的监控表定义的常用表标志。该部分是标志的低 8 位。
- 字段 2 包含分别为每个监控表定义的表特定的标志。该字段定义常用表标志后面接下来的 12 个低位。

有关如何为“日志监视器”表定义此 12 位的详细信息，请参阅[日志监视器表](#) (p. 316) 中的说明。

- 字段 3 包含为整数值保留的 12 个高位，以便用于表特定的标志。该字段包括特定于“日志监视器”表的标志。

以下部分定义各个标志位。您可以通过逻辑“OR”运算组合标志值。

以下列表介绍“日志监视器”表的标志：

0x00000001

禁止执行该条目的操作。

0x00000002

禁止发送该条目的 SNMP 陷阱。

0x00000004

禁止尝试重新初始化该条目。默认情况下，如果受监控的日志文件始终不可用，代理会定期尝试重新初始化该表条目。

0x00000008

禁止在 `sysedge.log` 文件中记录该条目的陷阱。设置该位对发送陷阱没有影响。在事件频繁发生或将特定条目用作代理监控信号时，禁用事件记录会很有用。

0x00000010

在代理每次尝试重新初始化日志文件监控并失败时，针对该条目发送连续的 `logMonEntryNotReady` 陷阱。

默认行为是，当受监控的日志文件不复存在或访问该日志文件出错时，发送单个 `logMonEntryNotReady` 陷阱。

代理会定期尝试重新初始化条目。启用该功能可使代理在每次重新初始化失败时发送其他的 `logMonEntryNotReady` 陷阱。

0x00000020

禁止将默认参数传递给操作脚本或程序。`SystemEDGE` 通常传递默认操作参数（表示陷阱类型、说明字段等）。有关操作参数的详细信息，请参阅[日志监视器表操作参数](#) (p. 323)。

0x00000040

禁止发送该条目的 `notReady` 陷阱。这包括不记录以及不执行 `notReady` 陷阱的操作。

0x00000100

将逻辑运算符 `NOT` 应用于正则表达式评估。启用该标志时，会将匹配的正则表达式评估为 `false`，因而不会发送陷阱或执行其他操作。将与正则表达式不匹配的所有条目评估为 `true`，因而会发送陷阱和执行任何指定操作。

0x00000200

跟踪日志文件大小，但不通过文件解析。该标志在使用自主监控跟踪日志文件是否存在和文件大小时很有用。

0xXXX00400

指定直到发生 `X` 个匹配之后，监视器才触发事件。该值根据每次扫描进行计算。如果设置该标志，将会抑制陷阱、事件日志记录和操作，直到匹配项数超过 `0x###00000` 中存储的 `X` 值；此时，将触发单个事件。此标志有助于最大程度减少经常发生的事件的陷阱流量。

0x00000800

指定此标志以监控目录，而不是特定日志文件。

0x00001000

以递归方式监控目录。

0x00002000

指定此标志以在目录监控期间不打开符号链接。请仅将此标志与 `0x00000800` 标志一起使用（仅限目录监控）。

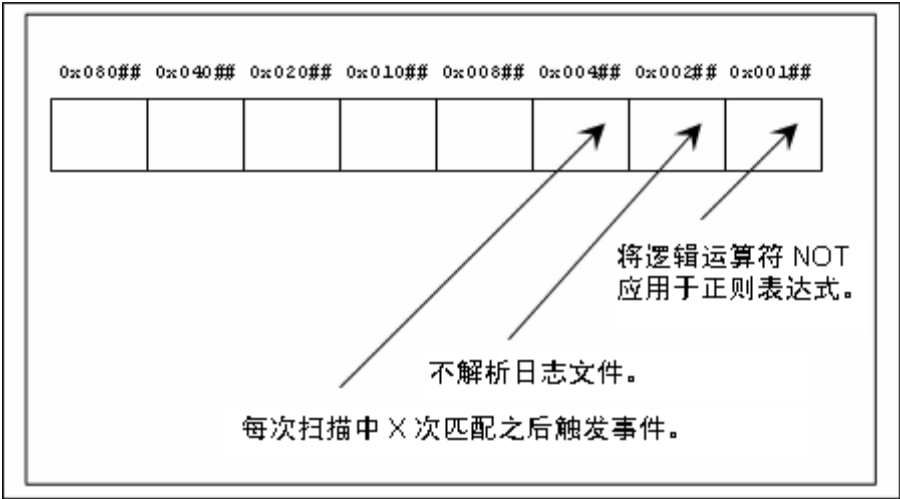
0x00004000

以千字节而不是字节为单位报告 `logMonitorLogFileSize` 变量。

0x###00000

此 X 值与 0x00000400 一起使用，可指定触发事件之前所需的事件数目。

下图显示特定于日志监视器表的标志位：



日志监视器表操作参数

SystemEDGE 代理可在操作命令被调用时向其提供若干默认参数。这些参数是对您在操作字符串中指定的任何参数的补充，并在您指定的那些参数之后通过命令行传递。默认参数与针对日志监视器表发送的 SNMP 陷阱中提供的参数一样。

注意：通过在表条目中设置 0x00000020 标志，可以防止代理在命令中包含这些操作参数。

以下列表介绍日志监视器表操作的默认参数：

trapType

定义所发送陷阱的类型，例如 logMonMatchEvent 或 logMonNotReadyEvent。

logMonitorLogFile

定义代理要监控的文件的名称。

logMonitorRegularExpression

定义代理尝试匹配条目的正则表达式。

logMonitorLastTrap

定义发送此陷阱的时间。

logMonitorLastMatch

定义触发此陷阱的日志文件中的行。

logMonitorDescr

定义表条目说明。

logMonitorIndex

定义表条目的索引。

logMonitorFlags

以十六进制表示法（例如，0x0000）定义表条目的标志字段。

logMonitorLogFileName

定义受监控文件的实际文件名，以防 logMonitorLogFile 参数使用通配符。

logMonitorSeverity

定义分配给表条目的重要级别。

查看日志监视器表

您可以使用 CA Virtual Assurance 来查看受管代理的日志监视器表的内容。

在 CA Virtual Assurance 中查看日志监视器表

1. 访问 CA Virtual Assurance 用户界面。
2. 单击“资源”选项卡。
此时将显示“浏览”窗格和“数据中心”页面。
3. 在“浏览”窗格中展开“受管”，并选择代理驻留的服务器。
此时将显示包含服务器详细信息的页面。
4. 单击“配置”，然后单击“日志监视器”。
此时将出现日志监视器表，其中包含所有现有的日志监视器。

日志监控配置

您可以控制 SystemEDGE 代理监控的日志文件，方法是在日志监视器表中添加、删除或修改相应的条目。

可以按下列方式配置日志监视器表：

动态使用 CA Virtual Assurance 中基于文件的配置

CA Virtual Assurance 可以配置安装在它从 CA Virtual Assurance 用户界面管理的任何系统上的代理。建议使用该配置方法，因为该方法可以利用一种用户友好且安全的机制动态地定义和传送监视器，该机制不需要代理重新启动。您可以为单个代理配置监控，也可以定义监控策略以在多个受管系统上应用。当您执行单个配置更改或应用配置策略时，CA Virtual Assurance 会将新的配置文件传送给代理系统。CA Virtual Assurance 可以在整个数据中心中充当代理配置的中心点，您可以覆盖或拒绝通过其他方式执行的更改。

有关使用 CA Virtual Assurance 配置代理的详细信息，请参阅 CA Virtual Assurance 文档。

从管理系统（例如 eHealth AdvantEDGE View、CA Spectrum 或 CA NSM）

动态使用 SNMP Set 命令

各种管理系统均支持通过 SNMP Set 命令将监视器添加到 SystemEDGE 中。eHealth 通过 AdvantEDGE View 接口提供该功能，CA NSM 通过 Agent View 启用配置，CA Spectrum 通过 OneClick 启用配置。通过 CA Virtual Assurance 进行动态配置所提供的所有功能都无法通过这些方法获得。

有关 SNMP Set 配置的详细信息，请参阅相应管理系统的文档。

通过在本地 sysedge.cf 文件中指定自主监视器表的条目来执行手动配置

手动配置要求您使用特定指令将条目添加到本地 sysedge.cf 文件中。当从 sysedge.cf 文件直接配置代理时，必须执行完全重新启动才能使更改生效。请使用位于代理数据目录中的 sysedge.cf 文件进行直接配置。

本节介绍如何在本地 sysedge.cf 文件中直接配置自主监控，但介绍的属性是所有形式的配置所必需的。

watch logfile 指令—将条目添加到日志监视器表

watch logfile 指令可用于通过 **sysedge.cf** 文件将日志监视器条目直接添加到日志监视器表。下列参数表示日志监视器表中的各列。

使用如下所述的语法向代理数据目录中的 **sysedge.cf** 文件添加一行，保存文件，然后重新启动代理以使更改生效。

使用 **watch logfile** 指令将条目添加到日志监视器表，如下所示：

```
watch logfile index flags 'file' 'regexpr'
['descr']['action'[interval[severity]]]
```

index

指定要用于该条目的监视器表的行（索引）。表中各行由索引号唯一标识。保留行 1 至 10 以供代理内部使用，因此索引值必须大于 10 且在表中唯一。

flags

使用十六进制标志值（例如，0x00000001）指定该条目的任何其他操作说明。有关可用标志的详细信息，请参阅[日志监视器表标志](#) (p. 321)。

'file'

指定要监控的日志文件的完整路径和文件名，或要监控的目录的完整路径。指定此参数时可以包含通配符（例如 * 和 ?）。

您监控的文件必须是基于 ASCII 的文本文件。**SystemEDGE** 不支持对其他字符集（如 Unicode）进行监控。

可以通过在文本编辑器中打开文件并选择“另存为”来确定文件的编码方式。编码方式将在“保存类型”字段中列出。

'regexpr'

指定扫描日志文件以获取匹配项时要使用的正则表达式。

SystemEDGE 支持的日志文件正则表达式不能超过 512 个字符。当监控目录时，将忽略此参数。

'descr'

指定表条目的任意说明（长度为 0 至 512 个字符）。

'action'

指定当正则表达式匹配且发送陷阱时要运行的命令（长度为 0 到 2048 个字符），包括完整路径和任何参数。如果字符串为空，则代理不为该条目执行操作。

注意：当代理执行操作时，您可以更改默认设置。有关详细信息，请参阅“代理配置”一章。

interval

指定监控此条目的频率（分钟）。有效时间间隔范围为 1 到 1440 分钟。

默认值：1

重要级别

指定当出现匹配项且发送陷阱时要分配给条目的重要级别。陷阱中包含重要级别。有效值如下：

- 无
- 正常
- 警告
- 轻微
- 重大
- 严重
- 致命

请注意，重要级别仅指定监视器的重要性，并不用于计算状态。

默认值：无

日志文件监控示例

本节提供的示例介绍如何在 `sysedge.cf` 文件中使用 `watch logfile` 指令将日志监视器条目添加到日志监视器表。

示例：搜索 pop 连接尝试

以下示例指示代理将条目添加到日志监视器表中表索引 15 处，以搜索系统上的 pop 连接尝试，并在出现匹配项时发送重要级别为警告的陷阱：

```
watch logfile 15 0x00 /var/log/syslog 'popper' 'NOTICE - pop connection' '' 1
warning
```

示例：搜索 su 尝试

以下示例指示代理将条目添加到日志监视器表中表索引 16 处，以搜索系统上的 su 尝试，并在出现匹配项时发送重要级别为严重的陷阱：

```
watch logfile 16 0x02 /var/adm/messages 'su.*fail' 'WARNING - su attempt'
'/local/bin/mail2admin' 5 critical
```

0x02

指定代理不应发送陷阱。相反，代理应该调用指定的操作命令。

示例：监控目录大小和内容

以下示例指示代理将条目添加到日志监视器表中表索引 17 处，以监控 /var/tmp 目录的大小和文件数。

```
watch logfile 17 0x00 /var/tmp 'Temporary directory' '' 5
```

可以创建自主监视器条目，以设置大小 (logMonitorLogFileSize) 和内容 (logMonitorLogFileCount) 属性的阈值。

以下示例指示代理将条目添加到日志监视器表中表索引 232 处，以监控目录大小和文件数。

```
watch logfile 232 0x1800 'C:\testdir23' '' Monitor for dir testdir23' '' 1 warning
```

标志 0x0800 可用于以非递归方式监控目录。此外，还创建自主监视器条目，以设置大小 (logMonitorLogFileSize) 和内容 (logMonitorLogFileCount) 属性的阈值。

```
monitor oid 1.3.6.1.4.1.546.11.1.1.8.232 2323232 0x8 30 absolute >= 10 'Directory  
Size is more than 10 bytes' '' 0 (or)  
monitor oid logMonitorLogFileSize.232 2323232 0x8 30 absolute >= 10 'Directory  
Size is more than 10 bytes' '' 0  
monitor oid logMonitorLogFileCount.232 2323231 0x8 30 absolute >= 10 'FileCount  
is more than 10' '' 0 (or)  
monitor oid 1.3.6.1.4.1.546.11.1.1.16.232 2323231 0x8 30 absolute >= 10 'File  
Count is more than 10' '' 0
```

示例：在多行上搜索模式

以下示例指定一个正则表达式，以便指示代理将条目添加到日志监视器表中表索引 20 处。该条目用于在受监控的日志文件中搜索在多行上出现的文本。在此示例中，文本 “WARNING:” 作为未指定的行数出现在文本 “Invalid login attempt” 前面。

```
watch logfile 20 0x00 /var/log/syslog '/^WARNING:(.*\n)*Invalid login attempt/m'  
'NOTICE - Invalid login attempt' '' 1 warning
```

注意：此示例要求在使用策略配置控制设置或使用 SystemEDGE 配置文件（如果在传统模式下）的代理上激活 Perl 兼容的正则表达式 (PCRE)。

详细信息

[Perl 兼容的正则表达式 \(PCRE\) 支持](#) (p. 145)

[配置文本模式排除](#) (p. 147)

[正则表达式示例](#) (p. 147)

edgework 实用工具—监控日志文件

edgework 命令行实用工具可自动配置 SystemEDGE 代理，以监控进程、进程组、日志文件和 Windows 事件日志。指定特定进程、进程组、日志文件或 Windows 事件日志以及关联的参数之后，edgework 实用工具会发出 SNMP Set 请求，以便在目标代理的相应监控表中创建适当的条目。edgework 实用工具位于代理安装的 bin 子目录中。

使用 edgework 实用工具监控日志文件，如下所示：

```
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 logfile command
```

-h *hostname* / *ipaddr*

指定要运行代理的系统的主机名或 IP 地址。接受 IPv4 和 IPv6 地址。

默认值：localhost

-p *port*

指定正在运行代理的 UDP 端口（例如，1691）。

默认：161

-c *community*

指定代理使用的团体字符串。仅对 SNMPv1 和 SNMPv2c 有效。

注意：为 snmpset 指定读/写团体字符串。

默认值：public

-v {*1* | *2c* | *3*}

表示正在运行代理的 SNMP 版本。指定 1 表示 SNMPv1，指定 2c 表示 SNMPv2c，指定 3 表示 SNMPv3。

默认：1

-u *secName*

指定用于 SNMPv3 安全的基于用户的安全模型 (USM) 用户名。

默认值：无

-s *secLevel*

为 SNMPv3 通信指定以下安全级别之一：

- 1 - noAuthNoPriv
- 2 - AuthNoPriv

3—AuthPriv（仅限 SNMPv3）

-n *contextName*

指定 MIBMuxed 代理的实例名称。

默认值：无

-a *authPassword*

指定身份验证密码（如果使用安全级别 2 (AuthNoPriv) 或 3 (AuthPriv) 为 SNMPv3 配置代理）。

注意：对于 SNMPv3 通信，该选项并非是必需选项。

默认值：无

-A {MD5 | SHA}

指定要使用的身份验证协议（如果使用安全级别 2 (AuthNoPriv) 或 3 (AuthPriv) 为 SNMPv3 配置代理）。当前仅使用 MD5（消息摘要算法）和 SHA（安全哈希算法）。

默认：MD5

-x *privPassword*

指定隐私（加密）密码（如果使用安全级别 3 (AuthPriv) 为 SNMPv3 配置代理）。

默认值：无

-X {DES | AES | 3DES}

指定隐私协议（如果使用安全级别 3 (AuthPriv) 配置 SNMPv3 用户）。指定 DES（数据加密标准）、AES（使用 128 位密钥的高级加密标准 (AES128)）和 3DES（三重数据加密标准）。

默认值：无

-m *FIPS_mode*

控制操作的 FIPS 模式。可接受的值为 0、1 和 2。

{0}

表示非 FIPS 模式。

1

表示 FIPS 共存模式。

2

表示仅 FIPS 模式。

默认：1

-r *retries*

指定重试次数。

默认：10

-t *timeout*

指定 SNMP 接收器将请求视为超时之前的持续时间。

默认值：10 秒

-d *logLevel*

指定 SNMP 消息的记录级别。接受的值为 0 至 5。

{0}

记录致命消息。

1

记录严重消息。

2

记录警告消息。

3

记录通知消息。

4

记录所有消息。

5

记录所有消息，包括调试消息。

默认：0

-f logfile

指定包含错误和调试信息的日志文件的名称。

默认值: sysedge_utility.log

命令

指定命令和相关参数。支持的命令如下:

- add
- setstatus
- 删除
- 列表

注意: 有关命令的详细信息, 请参阅[用于日志文件监控的 edgewatch 命令](#) (p. 332)。

用于日志文件监控的 edgewatch 命令

用于日志文件监控的 edgewatch 命令及相关参数如下所示:

```
add index flags "file" "regexpr" "descr" "action" interval
```

```
setstatus index status
```

```
delete index
```

列表

index

指定要用于该条目的监视器表的行(索引)。表中各行由索引号唯一标识。保留行 1 至 10 以供代理内部使用, 因此索引值必须大于 10 且在表中唯一。

flags

使用十六进制标志值(例如, 0x00000001)指定该条目的任何其他操作说明。有关可用标志的详细信息, 请参阅[日志监视器表标志](#) (p. 321)。

"file"

指定要监控的日志文件的完整路径和文件名, 或要监控的目录的完整路径。指定此参数时可以包含通配符(例如 * 和 ?)。

您监控的文件必须是基于 ASCII 的文本文件。SystemEDGE 不支持对其他字符集(如 Unicode)进行监控。

可以通过在文本编辑器中打开文件并选择“另存为”来确定文件的编码方式。编码方式将在“保存类型”字段中列出。

'regexpr'

指定扫描日志文件以获取匹配项时要使用的正则表达式。
SystemEDGE 支持的日志文件正则表达式不能超过 512 个字符。当监控目录时，将忽略此参数。

"descr"

指定表条目的任意说明（长度为 0 至 512 个字符）。

"action"

指定当正则表达式匹配且发送陷阱时要运行的命令（长度为 0 到 2048 个字符），包括完整路径和任何参数。如果字符串为空，则代理不为该条目执行操作。

注意：当代理执行操作时，您可以更改默认设置。有关详细信息，请参阅“代理配置”一章。

interval

指定监控此条目的频率（分钟）。

默认： 1

状态

指定设置日志监视器表中行的状态时要使用的 RowStatus 文本约定值。setstatus 操作使用此参数。

有效值如下所示。值可以是分配的整数值或实际显示的状态文本：

- active(1)
- notInService(2)
- destroy(6)

edgework 示例

本节提供的示例介绍如何将 edgework 实用工具与 SNMP 版本 1、2c 和 3 结合使用。

示例：列出日志监视器表中的条目

以下示例显示日志监视器表的内容：

```
edgework -v 1 -h 127.0.0.1 -c public -o logfile list
```

```
edgework -v 2c -h fe80::2367:1 -c public -o logfile list
```

```
edgework -v 3 -s 3 -u userName -x privPassword -X encryptProtocol -A authProtocol  
-a authPassword -o logfile list
```

添加日志监视器条目

以下示例指示代理将条目添加到日志监视器表中表索引 5 处，以搜索 HP-UX 系统上的 su 失败。代理在找到匹配项时会运行脚本 `/local/bin/mail2admin`。

```
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
```

示例：删除日志监视器条目

以下示例从日志监视器表中删除表索引 5 处的条目：

```
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
```

示例：禁用日志监视器条目

以下示例通过将条目状态设置为 `notInService(2)` 来禁用表索引 5 处的日志监视器表条目。该条目仍保留在表中，但代理不会扫描日志文件来查找正则表达式，除非状态恢复为 `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
```

删除日志监控条目

要停止监控日志文件或目录，必须从日志监视器表和 `sysedge.cf` 文件中删除相应条目。从 CA Virtual Assurance 用户界面使用点配置删除条目可以同时实现这两个目标。手动删除要求您分别执行操作以从两个位置删除条目。

在 CA Virtual Assurance 中删除日志监视器条目

1. 访问 CA Virtual Assurance 用户界面。
2. 单击“资源”选项卡。
此时将显示“受管资源”窗格和“数据中心”页面。
3. 在“受管资源”窗格中展开“受管”，然后选择代理驻留的服务器。
此时将显示包含服务器详细信息的页面。
4. 单击“配置”，然后单击“日志监视器”。
此时将出现日志监视器表，其中包含所有现有的自主监视器。
5. 选择要删除的监视器，然后单击“操作”>“删除”。
该条目将从该表中删除。代理执行热启动来应用更改。

手动删除日志监视器条目

1. 删除本地 `sysedge.cf` 文件中相应条目的 `watch logfile` 指令，并保存文件。
2. 使用 `-o` 参数的下列命令之一运行 `edgework` 实用工具：

```
-o delete index
```

```
-o setstatus index 6
```

index

指定要删除的日志监视器条目的索引值。

在使用 `setstatus` 命令时，值为 6 会将行状态设置为销毁并删除该条目。

3. 重新启动代理。
该条目即已删除。

日志文件监控建议

可以监控系统 and 应用程序日志，以获取有关用户、系统和应用程序行为的详细信息。

下表给出了可监控的日志文件以及可搜索的正则表达式（以便监控安全、设备故障、系统容量、Windows 安全以及应用程序和系统）的建议：

描述	要监控的日志文件	正则表达式
警告—后台进程核心	/var/log/daemon-log	core dumped
警告—后台进程核心	/var/log/syslog	core dumped
监控 su 尝试	/var/adm/messages	su.*fail
监控 rlogin	/var/log/syslog	in.rlogin
监控 telnet	/var/log/syslog	in.telnet
监控 rsh	/var/log/syslog	in.rsh
警告—非法指令，后台进程	/var/log/daemon-log	.*illegal.*nstruction
垃圾邮件中继尝试	/var/log/syslog	Relaying denied
监控 Linux 防火墙中的 DENY 数据包	/var/log/messages	DENY

注意：本节中介绍的日志文件针对不同的操作系统会有所不同。提供的这些示例仅供参考。您应该针对自己的操作系统改变日志文件。

下表给出了可在系统日志 (/var/adm/messages) 中搜索的正则表达式（以便监控设备故障）的建议：

描述	正则表达式
严重: Badtrap 错误	.*BAD TRAP.*
错误: SCSI 错误	.*SCSI.*[E,e]rror.*
错误: SCSI 错误	.*SCSI.*failed.*
错误: SCSI 错误	.*SCSI.*hung.*
严重: badsimms 错误	.*SIMM.*
严重: badsimms 错误	.*BAD.*SIMM.*
严重: 内存错误	.*[M,m]emory [E,e]rror.*

描述	正则表达式
错误: 磁盘错误	.*disk not responding.*
错误: 磁盘错误	.*[D,d]isk.*[E,e]rror.*
“警告: 磁盘碎片错误”	.*optimization changed.*
错误: 磁盘错误	.*corrupt label.*
错误: I/O 错误	.*I/O.*[E,e]rror.*
错误: 磁盘读取/写入错误	.*Error for Command:.*[read,write].*
错误: 介质错误	.*Media Error.*
信息: 串行端口错误	.*zs[0,1,2]: silo overflow.*
警告: 载波错误	.*no carrier.*
警告: 链接已关闭	.*Link Down - cable problem?.*
错误: SDS 错误	.*[NOTICE,WARNING,PANIC]: md:.*

下表给出了可在 `/var/adm/messages` 中搜索的正则表达式（以便监控系统容量）的建议：

描述	正则表达式
严重: 内存错误	*[O,o]ut of [M,m]emory.*
严重: 内存错误	.*[F,f]ile system full.*
错误: 磁盘空间错误	.*No space left on device.*

下表给出了可在 Windows 事件日志中监控的日志和表达式（以便监控 Windows 安全）的建议：

描述	事件日志	安全类型	正则表达式
随机密码黑客	安全	全部	.*bad
误用权限	安全	全部	.*[user rights,group management,security change, restart,shutdown]
不正确的文件访问	安全	失败	.*[read,write]
不正确的打印机访问	安全	失败	.*print
病毒爆发警告: 程序文件已更新	安全	全部	.*write.*[exe,dll,com]

描述	事件日志	安全类型	正则表达式
安全策略更改	应用程序	信息	.*[S,s]ecurity policy

下表给出了可在 Windows 事件日志中监控的日志和表达式（以便监控应用程序和系统）的建议：

描述	事件日志	安全类型	正则表达式
应用程序错误或失败	应用程序	全部	.*[F,f]ail . *[E,e]rror
应用程序加载问题	应用程序	全部	.*[L,l]oad. *[P,p]roblem
安装了新软件	应用程序	全部	.*[I,NSTALL,Install,instal l]
初始化过程中服务器 进程失败	应用程序	全部	.*4131
磁盘故障和错误	全部	全部	.*[D,d]isk
网络适配器错误	全部	错误	.*[N,n]etwork [A,a]dapter

轮转日志文件

通过使用相应通配符条目来指定日志文件条目，可以监控轮转日志文件。通配符语法依赖于平台，并允许您使用星号 (*) 表示任意数目的字符，使用问号 (?) 表示任何单个字符。代理会监控最新的匹配日志文件，并在创建和更新下一个滚动日志文件时自动切换到该日志文件。

第 13 章： Windows 事件监控

本章介绍如何使用 SystemEDGE 代理根据正则表达式监控 Windows 事件日志。

重要信息！ 本章仅与 Windows 版本的 SystemEDGE 代理相关。

此部分包含以下主题：

[Windows 事件监控概述](#) (p. 339)

[NT 事件监视器表](#) (p. 341)

[Windows 事件监控配置](#) (p. 350)

[edgewatch 实用工具—监控 Windows 事件](#) (p. 354)

[删除 Windows 事件监控条目](#) (p. 360)

Windows 事件监控概述

SystemEDGE 代理可用于根据正则表达式连续监控 Windows 事件日志。此功能类似于“日志文件监控”一章中介绍的标准日志文件监控。使用 NT 事件监视器表来指定事件日志、事件类型、匹配源和说明的正则表达式、重要级别以及其他值。代理会自动监控定义的日志，并在检测到正则表达式匹配项时将陷阱发送到管理系统。

代理还可以运行操作命令，以立即处理事件。由于 Windows 事件除文本消息之外还包含多个识别特征，因此，就可以指定的匹配项类型而言，此监控功能比标准日志文件监控更复杂。

当 SystemEDGE 代理启动时（或向 NT 事件监视器表中添加行之后），它会检查每个 Windows 事件日志的状态，以获取其当前长度以及上次更新时间。此后，代理会定期扫描每个事件日志，以获取自上次更新之后进行的添加或修改。如果事件日志文件已更改，代理将仅扫描这些更改（而不是整个事件日志），以确定是否存在指定筛选的匹配项。

Windows 事件搜索条件

每个 Windows 事件监视器表条目指示代理根据下表中所述的条件搜索匹配项：

事件日志

指定事件日志的名称。该值可以是以下任一项：

- 应用程序
- 系统
- 安全
- DirService（对于目录服务）
- DnsServer（对于 DNS 服务）
- FileRepService（对于文件复制服务）

事件类型

指定事件的类型。类型 1 至 5 由 Windows 定义，如下所示：

- 错误(1)
- 警告(2)
- 信息(3)
- 成功(4)
- 失败(5)

类型“所有(6)”指示代理应该匹配所有事件类型。

事件源

指定生成事件的程序或模块的名称。代理使用正则表达式匹配该字段。

事件说明

说明事件。代理使用正则表达式匹配该字段。

当 SystemEDGE 代理根据全部四个条件找到匹配项时，该代理会生成 SNMP 陷阱消息。这种匹配类似于布尔 AND 运算。

NT 事件监视器表

NT 事件监视器表提供代理当前根据特定正则表达式模式监控的每个事件日志的信息。每行表示一个已定义的 Windows 事件监视器条目。

对于 NT 事件监视器表中的每个条目，表提供以下类型的信息：

- 代理监控的事件日志
- 监控日志所依据的正则表达式
- 因找到匹配而发送陷阱的次数
- 上次发送陷阱的时间
- 导致上次匹配的日志条目
- 重要级别

NT 事件监视器表列

以下列表描述 NT 事件监视器表的列。有关 NT 事件监视器表的详细信息，请参阅代理安装的 `mib` 子目录中的 `Systems Management Empire MIB empire.asn1`。

ntEventMonIndex

为该条目定义唯一的行索引（1 到 MAXINT）。行 1 至 10 保留供代理内部使用，其他行的索引必须在 11 至 MAXINT 范围内。

权限：只读

ntEventMonLog

定义要监控的事件日志。有效值如下：

- Application(1)
- Security(2)
- System(3)
- Directory Service(4)
- DNS Service(5)
- File Replication Service(6)

权限：读写

ntEventMonTime

根据 `sysUpTime` 定义事件发生的时间。

权限：只读

ntEventMonTraps

定义该条目发生匹配且代理发送陷阱的次数 (ntEventMonMatchTrap)。

权限：只读

ntEventMonTypeLastMatch

定义匹配搜索条件的最后一个事件的事件类型。类型 1 到 5 由 Windows 定义，如下所示：

- 错误(1)
- 警告(2)
- 信息(3)
- 成功(4)
- 失败(5)

类型“未匹配(6)”表示对于该监控条目还没有匹配事件。

权限：只读

ntEventMonTypeFilter

定义要匹配该条目的事件类型。类型 1 到 5 由 Windows 定义，如下所示：

- 错误(1)
- 警告(2)
- 信息(3)
- 成功(4)
- 失败(5)

类型“所有(6)”指示代理应该匹配所有事件类型。

权限：读写

ntEventMonSrcLastMatch

定义匹配该监控条目的最后一个事件日志条目的事件源。事件源通常是生成事件的程序的名称。每当发生匹配时，该变量都将更新。

权限：只读

ntEventMonSrcFilter

定义扫描事件中的匹配时要应用于事件源的正则表达式。

权限：读写

ntEventMonDescLastMatch

定义匹配该监控条目的最后一个事件日志条目。每当发生匹配时，该变量都将更新。

权限：只读

ntEventMonDescFilter

定义扫描事件中的匹配时要应用于事件说明的正则表达式。

权限：读写

ntEventMonStatus

定义行状态，行状态可以是以下状态之一：

- 活动
- notInService
- notReady
- createAndGo
- createAndWait

通常，行处于 active 或 notInService 状态。这些值的含义与由 SNMPv2 SMI RowStatus 文本约定定义的值的含义相同。

权限：读写

默认：createAndWait(5)

ntEventMonDescr

定义受监控事件的说明（长度为 0 到 512 个字符）。

权限：读写

ntEventMonAction

定义找到匹配并发送陷阱时要运行的带有任何参数的引用命令（长度为 0 到 2048 个字符）。如果该字符串为空，代理将不为此条目调用任何操作。

注意：请勿使用 Windows 批处理文件进行操作；它们施加严格的编程限制，并且经常错误地使用桌面应用程序。应使用更强大且灵活的脚本语言，如 Perl 或 Visual Basic。

权限：读写

ntEventMonFlags

定义表示其他行为语义的正整数标志，条目在其操作过程中将遵循这些语义。有关可用标志和设置标志的详细信息，请参阅 [NT 事件监视器表标志](#) (p. 346)。

ntEventMonSeverity

定义发送陷阱时分配给条目的重要级别。有效值如下：

- 无
- 正常
- 警告
- 轻微
- 重大
- 严重
- 致命

默认值：无

行创建对象

NT 事件监视器表的 **ntEventMonIndex** 列充当关键字字段（或行索引）以区分表中的行。保留 1 到 10 行以供 SystemEDGE 代理内部使用。您可以配置范围 11 到 MAXINT 的行。

您可以使用具有 NT 事件监视器表的下列 MIB 对象来优化行创建：

ntEventUnusedIndex

在对象上执行 **SNMP Get** 时，返回 NT 事件监视器表的未使用的索引号。查询该对象来获取可用于创建行的索引号。

ntEventMatchDescr

确定与 **ntEventMonDescr** 属性中特定的条目说明对应的索引号。在具有 **ntEventMonDescr** 属性值的该 MIB 对象上执行 **SNMP Set**，从而导致代理在“自主监视器”表中搜索条目，并且将最后一个匹配条目的索引值放入 **monMatchIndex** MIB 对象。

ntEventMatchIndex

用于 **ntEventMatchDescr** 时将特定的条目说明与其索引号相对应。

按照本地策略，您可以选择为系统管理预留行块。然后可以在保留的行块中定义条目，而无需担心行可能已经被其他用户条目使用。按照本地的策略，当所有其他用户定义用户配置的条目时，应使用在保留范围以外的行索引。

通过保留行块，您可以定义要跨所有计算机监控的一组一致的条件（行条目），以便同样的条件定义在每台计算机的相同行数中。例如，您可以使用第 11 行 (**monIndex = 11**) 定义用于监控 **swapCapacity** 变量的条目，并可将该配置分发给每个系统，以便第 11 行对每个系统中的 **swapCapacity** 变量进行监控。

在 CA Virtual Assurance 上配置代理可以自动执行该过程。您可以使用特定行号配置条目块，并将该配置部署到一个操作中的所有受管系统。此外，CA Virtual Assurance 将检测您是否定义已在使用中的行并禁止该操作。

有关在 CA Virtual Assurance 中定义行的详细信息，请参阅 CA Virtual Assurance 文档。

NT 事件监视器表标志

可以使用 NT 事件监视器表中的 `ntEventMonFlags` 列来指定相应 NT 事件监视器表行的其他行为语义。默认情况下，NT 事件监视器表行执行以下操作：

- 尝试对自身进行重新初始化
- 发送 SNMP 陷阱
- 将事件记录到普通日志文件
- 调用操作（如果进行了配置）

您可以设置不同标志位来改变这些默认值。SystemEDGE 代理以十六进制（基数为 16）表示法解释所有标志。

NT 事件监视器表标志字段 (`ntEventmonFlags`) 如下所示：

标志值由三个字段组成：

- 字段 1 包含为 Systems Management Empire MIB 的监控表定义的常用表标志。该部分是标志的低 8 位。
- 字段 2 包含分别为每个监控表定义的表特定的标志。该字段定义常用表标志后面接下来的 12 个低位。对于 Windows 事件监控，当前没有定义特定于表的标志。
- 字段 3 包含为整数值保留的 12 个高位，以便用于表特定的标志。对于 Windows 事件监控，当前没有定义特定于表的标志。

以下部分定义各个标志位。可以通过逻辑 OR 操作组合标志值。

以下列表描述监控表的通用表标志：

0x00000001

禁止执行该条目的操作。

0x00000002

禁止发送该条目的 SNMP 陷阱。

0x00000004

禁止尝试重新初始化该条目。默认情况下，如果监控的事件日志不可用，代理将定期尝试重新初始化该表条目。

0x00000008

禁止在 `sysedge.log` 文件中记录该条目的陷阱。设置该位对发送陷阱没有影响。在事件频繁发生或将特定条目用作代理监控信号时，禁用事件记录会很有用。

0x00000010

每次代理尝试重新初始化事件日志监控时为该条目发送连续 `ntEventMonEntryNotReady` 陷阱并失败。代理监控的事件日志不再存在时或访问该事件日志发生错误时，将代理的默认行为发送至单个 `ntEventMonEntryNotReady` 陷阱。代理会定期尝试重新初始化条目。每次重新启动失败时，启用此功能将导致代理发送其他 `ntEventMonEntryNotReady` 陷阱。

0x00000020

禁止将默认参数传递给操作脚本或程序。`SystemEDGE` 通常传递默认操作参数（表示陷阱类型、说明字段等）。有关操作参数的详细信息，请参阅 [NT 事件监视器表操作参数](#) (p. 348)。

0x00000040

禁止发送该条目的 `notReady` 陷阱。这包括不记录以及不执行 `notReady` 陷阱的操作。

0x00000100

将事件 ID 号预附加到事件说明字段。添加 `[#]` 格式的标识符，其中 `#` 为标识符。该标志的用途是搜索特定事件 ID。例如，要指定仅事件 ID 528 匹配，请指定该标志并将其 `ntEventMonDescFilter` 设置为 `[528]`。

0x00000200

将逻辑 NOT 运算符应用于事件源、类型和说明评估。启用该标志时，匹配指定源、类型和说明的事件将评估为 **false** 并导致没有任何陷阱或操作。不匹配一个或多个事件源、类型和说明的事件将评估为 **true** 并导致陷阱和指定操作。

注意： 设置该标志时要谨慎。

0x####0000

NT 事件监视器表不使用该标记值字段；设置该字段对 NT 事件监视器表操作或任何支持的标志位没有任何影响。

NT 事件监视器表操作参数

SystemEDGE 代理向操作命令提供多个默认参数。除操作字符串中指定的参数以外，还包含这些参数，它们将接着您指定的参数在命令行上传递。默认参数与针对 NT 事件监视器表发送的 SNMP 陷阱中提供的参数一样。

注意： 通过在表条目中设置 0x00000020 标志，可以防止代理在命令中包含这些操作参数。

以下列表描述 NT 事件监视器表操作的默认参数：

trapType

定义发送的陷阱的类型，如 ntEventMonMatchEvent 或 ntEventMonNotReadyEvent。

ntEventMonLog

定义代理监控的事件日志。

ntEventMonTypeLastMatch

定义生成该陷阱的事件的类型。

ntEventMonTime

定义生成该事件的时间。

ntEventMonSrcLastMatch

定义生成该陷阱的事件的源。

ntEventMonDescLastMatch

定义生成该陷阱的事件的说明。

ntEventMonDesc

定义表条目的说明。

ntEventMonIndex

定义表条目的索引。

ntEventMonFlags

以十六进制表示法（例如，0x0000）定义表条目的标志字段。

ntEventMonSeverity

定义分配给表条目的重要级别。

有关 SystemEDGE 代理发送的陷阱的详细信息，请参阅“概念”一章和附录“私人企业陷阱”。

查看 NT 事件监视器表

可以使用 CA Virtual Assurance 查看受管代理的 NT 事件监视器表的内容。

查看 CA Virtual Assurance 中的 NT 事件监视器表

1. 访问 CA Virtual Assurance 用户界面。
2. 单击“资源”选项卡。
此时将显示“受管资源”窗格和“数据中心”页面。
3. 在“受管资源”窗格中展开“受管”，然后选择代理驻留的服务器。
此时将显示包含服务器详细信息的页面。
4. 单击“配置”，然后单击“Windows 事件监视器”。
此时将显示 Windows 事件监视器表以及现有的所有 Windows 事件监视器。

Windows 事件监控配置

可以通过添加、删除或修改 NT 事件监视器表中的条目来控制 SystemEDGE 代理监控的事件日志。

可以通过下列方法配置 NT 事件监视器表：

动态使用 CA Virtual Assurance 中基于文件的配置

CA Virtual Assurance 可以配置安装在它从 CA Virtual Assurance 用户界面管理的任何系统上的代理。建议使用该配置方法，因为该方法可以利用一种用户友好且安全的机制动态地定义和传送监视器，该机制不需要代理重新启动。您可以为单个代理配置监控，也可以定义监控策略以在多个受管系统上应用。当您执行单个配置更改或应用配置策略时，CA Virtual Assurance 会将新的配置文件传送给代理系统。CA Virtual Assurance 可以在整个数据中心中充当代理配置的中心点，您可以覆盖或拒绝通过其他方式执行的更改。

有关使用 CA Virtual Assurance 配置代理的详细信息，请参阅 CA Virtual Assurance 文档。

从管理系统（例如 eHealth AdvantEDGE View、CA Spectrum 或 CA NSM）动态使用 SNMP Set 命令

各种管理系统均支持通过 SNMP Set 命令将监视器添加到 SystemEDGE 中。eHealth 通过 AdvantEDGE View 接口提供该功能，CA NSM 通过 Agent View 启用配置，CA Spectrum 通过 OneClick 启用配置。通过 CA Virtual Assurance 进行动态配置所提供的所有功能都无法通过这些方法获得。

有关 SNMP Set 配置的详细信息，请参阅相应管理系统的文档。

通过在本本地 sysedge.cf 文件中指定自主监视器表的条目来执行手动配置

手动配置要求您使用特定指令将条目添加到本地 sysedge.cf 文件中。当从 sysedge.cf 文件直接配置代理时，必须执行完全重新启动才能使更改生效。请使用位于代理数据目录中的 sysedge.cf 文件进行直接配置。

本节介绍如何在本地 sysedge.cf 文件中直接配置自主监控，但介绍的属性是所有形式的配置所必需的。

watch ntevent 指令—将条目添加到 NT 事件监视器表中

通过 `watch ntevent` 指令可以在 `sysedge.cf` 文件中直接将 Windows 事件监视器条目添加到 NT 事件监视器表中。参数表示 NT 事件监视器表中的列。

使用如下所述的语法向代理数据目录中的 `sysedge.cf` 文件添加一行，保存文件，然后重新启动代理以使更改生效。

使用 `watch ntevent` 指令将条目添加到 NT 事件监视器表中，如下所示：

```
watch ntevent index flags 'evLog' 'evType' 'evSrc' 'evDescr' ['descr'] ['action']  
[severity]
```

index

指定要用于该条目的监视器表的行（索引）。表中各行由索引号唯一标识。保留行 1 至 10 以供代理内部使用，因此索引值必须大于 10 且在表中唯一。

flags

使用十六进制标志值（例如，0x00000001）指定该条目的任何其他操作说明。有关可用标志的详细信息，请参阅 [NT 事件监视器表标志](#) (p. 346)。

'evLog'

指定要监控的事件日志。该值可以是以下任何一种：

- application
- 系统
- 安全性
- dnsServer
- dirService
- fileRepService

'evType'

指定要匹配该条目的事件类型。有效类型如下：

- 错误
- 警告
- 信息
- 成功
- 失败
- all

输入“all”表示所有事件类型都应匹配。

'evSrc'

指定扫描每个事件中的事件源属性时要使用的正则表达式。

'evDescr'

指定扫描每个事件中的事件说明属性时要使用的正则表达式。

'descr'

指定表条目的任意说明（长度为 0 至 512 个字符）。

'action'

指定条目匹配且发送陷阱时要运行的命令（长度为 0 到 2048 个字符），包括完整路径和所有参数。如果字符串为空，则代理不为该条目执行操作。

注意：当代理执行操作时，您可以更改默认设置。有关详细信息，请参阅“代理配置”一章。

重要级别

指定当出现匹配项且发送陷阱时要分配给条目的重要级别。陷阱中包含重要级别。有效值如下：

- 无
- 正常
- 警告
- 轻微
- 重大
- 严重
- 致命

请注意，重要级别仅指定监视器的重要性，并不用于计算状态。

默认值：无

Windows 事件监控示例

本节为 NT 事件监视器表提供示例条目，以使用 `watch ntevent` 指令监控 Windows 事件日志。可以将这些条目添加到 `sysedge.cf` 文件中。

示例：在应用程序日志中搜索 Web 服务器消息

以下示例将新条目添加到表索引为 11 的代理 NT 事件监视器表中，以在应用程序日志中搜索 http Web 服务器应用程序消息并在发生匹配时发送重要级别为警告的陷阱：

```
watch ntevent 11 0x00 Application All 'http' '.*' 'Web Server messages' " warning
```

示例：在安全日志中搜索失败事件

以下示例将新条目添加到表索引为 12 的代理 NT 事件监视器表中，以在安全日志搜索表示登录失败的失败事件并在发生匹配时发送重要级别为严重的陷阱：

```
watch ntevent 12 0x00 Security Failure '.*' '.*' 'Access Failure - WARNING' "
critical
```

示例：在应用程序日志中搜索特定事件

以下示例将新条目添加到表索引为 3 的代理 NT 事件监视器表中，以在应用程序日志搜索事件 ID 为 277 的事件：

```
watch ntevent 3 0x0100 Application All '.*' '\[277\]' 'Event ID 277' "
```

0x0100

将事件 ID 添加到说明中。`[277\]` 是代理将尝试匹配的说明字段。

需要使用反斜杠字符 (`\`)，因为方括号 (`[]`) 是正则表达式匹配的特殊字符。

详细信息

[Perl 兼容的正则表达式 \(PCRE\) 支持](#) (p. 145)

[配置文本模式排除](#) (p. 147)

[正则表达式示例](#) (p. 147)

edgework 实用工具—监控 Windows 事件

edgework 命令行实用工具可自动配置 SystemEDGE 代理，以监控进程、进程组、日志文件和 Windows 事件日志。指定特定进程、进程组、日志文件或 Windows 事件日志以及关联的参数之后，edgework 实用工具会发出 SNMP Set 请求，以便在目标代理的相应监控表中创建适当的条目。edgework 实用工具位于代理安装的 bin 子目录中。

虽然仅在 Windows 上提供 Windows 事件监控功能，但可以使用 edgework 实用工具从任何支持的平台配置它。

使用 edgework 实用工具监控 Windows 事件，如下所示：

```
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 ntevent command
```

-h *hostname / ipaddr*

指定要运行代理的系统的主机名或 IP 地址。接受 IPv4 和 IPv6 地址。

默认值：localhost

-p *port*

指定正在运行代理的 UDP 端口（例如，1691）。

默认：161

-c *community*

指定代理使用的团体字符串。仅对 SNMPv1 和 SNMPv2c 有效。

注意：为 snmpset 指定读/写团体字符串。

默认值：public

-v {1 | 2c | 3}

表示正在运行代理的 SNMP 版本。指定 1 表示 SNMPv1，指定 2c 表示 SNMPv2c，指定 3 表示 SNMPv3。

默认：1

-u *secName*

指定用于 SNMPv3 安全的基于用户的安全模型 (USM) 用户名。

默认值：无

-s *secLevel*

为 SNMPv3 通信指定以下安全级别之一：

- 1 - noAuthNoPriv
- 2 - AuthNoPriv
- 3 - AuthPriv (仅限 SNMPv3)

-n *contextName*

指定 MIBMuxed 代理的实例名称。

默认值：无

-a *authPassword*

指定身份验证密码（如果使用安全级别 2 (AuthNoPriv) 或 3 (AuthPriv) 为 SNMPv3 配置代理）。

注意：对于 SNMPv3 通信，该选项并非必需选项。

默认值：无

-A {MD5 | SHA}

指定要使用的身份验证协议（如果使用安全级别 2 (AuthNoPriv) 或 3 (AuthPriv) 为 SNMPv3 配置代理）。当前仅使用 MD5（消息摘要算法）和 SHA（安全哈希算法）。

默认：MD5

-x *privPassword*

指定隐私（加密）密码（如果使用安全级别 3 (AuthPriv) 为 SNMPv3 配置代理）。

默认值：无

-X {DES | AES | 3DES}

指定隐私协议（如果使用安全级别 3 (AuthPriv) 配置 SNMPv3 用户）。指定 DES（数据加密标准）、AES（使用 128 位密钥的高级加密标准 (AES128)）和 3DES（三重数据加密标准）。

默认值：无

-m *FIPS_mode*

控制操作的 FIPS 模式。可接受的值为 0、1 和 2。

{0}

表示非 FIPS 模式。

1

表示 FIPS 共存模式。

2

表示仅 FIPS 模式。

默认：1

-r *retries*

指定重试次数。

默认：10

-t *timeout*

指定 SNMP 接收器将请求视为超时之前的持续时间。

默认值：10 秒

-d *logLevel*

指定 SNMP 消息的记录级别。接受的值为 0 至 5。

{0}

记录致命消息。

1

记录严重消息。

2

记录警告消息。

3

记录通知消息。

4

记录所有消息。

5

记录所有消息，包括调试消息。

默认：0

-f logfile

指定包含错误和调试信息的日志文件的名称。

默认值: sysedge_utility.log

命令

指定命令和相关参数。支持的命令如下:

- add
- setstatus
- 删除
- 列表

注意: 有关命令的详细信息, 请参阅[用于 Windows 事件监控的 edgework 命令](#) (p. 357)。

用于 Windows 事件监控的 edgework 命令

edgework Windows 事件监控命令及相关参数如下:

```
add index flags evLog evType "evSrc" "evDesc" "descr" "action"
```

```
setstatus index status
```

```
delete index
```

列表

index

指定要用于该条目的监视器表的行(索引)。表中各行由索引号唯一标识。保留行 1 至 10 以供代理内部使用, 因此索引值必须大于 10 且在表中唯一。

flags

使用十六进制标志值(例如, 0x00000001)指定该条目的任何其他操作说明。

evLog

指定要监控的事件日志。该值可以是以下任何一种:

- application
- 系统
- 安全性
- dnsServer
- dirService
- fileRepService

evType

指定要匹配该条目的事件类型。有效类型如下：

- 错误
- 警告
- 信息
- 成功
- 失败
- all

输入“all”表示所有事件类型都应匹配。

"evSrc"

指定扫描每个事件中的事件源属性时要使用的正则表达式。

"evDescr"

指定扫描每个事件中的事件说明属性时要使用的正则表达式。

"descr"

指定表条目的任意说明（长度为 0 至 512 个字符）。

"action"

指定条目匹配且发送陷阱时要运行的命令（长度为 0 到 2048 个字符），包括完整路径和所有参数。如果字符串为空，则代理不为该条目执行操作。

注意：当代理执行操作时，您可以更改默认设置。有关详细信息，请参阅“代理配置”一章。

状态

指定设置与 **setstatus** 操作一起使用的 NT 事件监视器表中的行状态时要使用的状态文本约定值。有效值如下所示。值可以是分配的整数值或实际显示的状态文本：

- active(1)
- notInService(2)
- destroy(6)

edgewatch 示例

本节提供将 edgewatch 实用工具与 SNMP 版本 1、2c 和 3 结合使用的示例，以监控 Windows 事件。

示例：列出 NT 事件监视器表中的条目

以下示例列出代理的 NT 事件监视器表的内容：

```
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
```

示例：添加 NT 事件监控条目

以下示例将新条目添加到在表索引为 5 的代理 NT 事件监视器表中，以搜索 Windows 系统上的登录失败。

```
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"
```

此示例还指示当代理找到匹配时代理运行 \\local\\bin\\mail2admin.exe 脚本。

示例：删除 NT 事件监控条目

以下示例从表索引为 5 的 NT 事件监视器表中删除条目：

```
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
```

示例：禁用 NT 事件监控条目

以下示例通过将条目状态设置为 `notInService(2)` 禁用表索引为 5 的 NT 事件监视器表条目。条目仍保留在表中，但代理不会扫描事件日志中的匹配项，除非条目状态返回到 `active(1)`。

```
edgework -v 1 -h 127.0.0.1 -c private ntevent setstatus 5 2
```

```
edgework -v 2c -h fe80:ab01::901:bdef -c private ntevent setstatus 5 2
```

```
edgework -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

对应于行状态文本约定值 `notInService(2)`。

删除 Windows 事件监控条目

要停止监控 Windows 事件，必须从 NT 事件监视器表和 `sysedge.cf` 文件中删除相应条目。从 CA Virtual Assurance 用户界面使用点配置删除条目可以同时实现这两个目标。手动删除要求您分别执行操作以从两个位置删除条目。

删除 CA Virtual Assurance 中的 Windows 事件监控条目

1. 访问 CA Virtual Assurance 用户界面。
2. 单击“资源”选项卡。
此时将显示“受管资源”窗格和“数据中心”页面。
3. 在“受管资源”窗格中展开“受管”，然后选择代理驻留的服务器。
此时将显示包含服务器详细信息的页面。
4. 单击“配置”，然后单击“NT 事件监控”。
此时将显示 NT 事件监视器表以及现有的所有自主监视器。
5. 选择要删除的监视器，然后单击“操作”>“删除”。
该条目将从该表中删除。代理执行热启动来应用更改。

手动删除 Windows 事件监控条目

1. 删除本地 `sysedge.cf` 文件中的 `watch ntevent` 指令并保存文件。
2. 使用 `-o` 参数的下列命令之一运行 `edgework` 实用工具：

```
-o delete index
```

```
-o setstatus index 6
```

index

指定要删除的 Windows 事件监控条目的索引值。

在使用 `setstatus` 命令时，值为 6 会将行状态设置为销毁并删除该条目。

3. 重新启动代理。

该条目即已删除。

第 14 章：历史记录收集

本章介绍了 SystemEDGE 代理的历史记录采样功能，以及如何随着时间推移指示代理监控和存储 MIB 变量的值以供管理员在将来检索。

此部分包含以下主题：

[历史记录收集概述](#) (p. 363)

[历史记录控制表和历史记录数据表](#) (p. 364)

[历史记录采样示例](#) (p. 367)

[查看历史记录控制表](#) (p. 369)

[历史记录控制表配置](#) (p. 369)

历史记录收集概述

SystemEDGE 可以随着时间的推移跟踪各种基于整数的 MIB 对象（计数器、标尺等）的值，并存储这些值供以后检索。该功能（通常称为历史记录采样）可大大减少网络中管理站轮询的数量。

无需随着时间的推移通过持续不断的轮询来收集 MIB 变量的值，管理员只需指示代理抽样并存储这些值即可。管理系统可以定期联系代理，上载完整的抽样历史记录。甚至在网络断开期间，当管理系统无法与代理进行通信时，代理还是可以继续抽样并存储指定的 MIB 值。

历史记录抽样

代理使用以下两个 SNMP MIB 表提供历史记录功能：

- 历史记录控制表，用于定义数据收集功能。
- 历史记录表，用于存储实际的数据抽样。

通过控制表，您可以动态配置代理，使其抽样并存储受其控制的任何基于整数的 MIB 变量的值。数据表存储这些值供将来的检索使用。

要执行基准和趋势分析，可以将 SystemEDGE 配置为监控和存储 swapCapacity 的值，例如，通过配置代理使其每 5 分钟对值抽样一次并存储最近的 144 个抽样。然后，每 12 个小时，管理系统可以将全部 144 个抽样上载一次，以获取在上一个 12 小时期间收集的 swapCapacity 值。

历史记录控制表和历史记录数据表

历史记录控制表包含一些参数，这些参数描述代理将抽样并存储在历史记录表中的数据。历史记录控制表的每一行都会将值分配给该表的参数（列式对象），进而定义特定的数据收集功能。“历史记录”表中的一行或多行（存储的采样）与该单个控制行关联。

控制表每一行分配有唯一的 **empireHistoryCtrlIndex** 值。行通过指定要抽样的对象实例、抽样的频率（30 秒的倍数）以及要保留的抽样数（存储段）来定义数据收集功能。与每个数据收集功能（控制表的行）关联的是历史记录表中的一组行。历史记录表中的每一行（也称为一个存储段）都保存有在一个采样时间间隔内收集的指定 MIB 对象的值。

在每个采样时间间隔发生时，代理会根据该数据收集功能将其 **empireHistoryIndex** 与其他行相同的新行添加到历史记录表中。每个新行与历史记录控制表中的单个行相对应，并且其 **empireHistorySampleIndex** 大于上一个抽样的 **SampleIndex**（差为一）。

历史记录控制表的列

以下列表介绍了历史记录控制表的各列：

empireHistoryCtrlIndex

指定可唯一标识表中条目的整数（1 到 MAXINT）。

empireHistoryCtrlDescr

描述由该条目定义的数据收集功能。

empireHistoryCtrlInterval

指定一个整数值，该值表示代理应对 MIB 变量进行抽样的频率（秒）。

注意：时间间隔必须是 30 秒的倍数。

empireHistoryCtrlObjID

指定要进行抽样的 MIB 变量的完整对象实例标识符。

注意：必须包括对象标识符的实例部分（例如，表示标量的 .0）。对象实例必须存在并包含在 Systems Management Empire MIB 中。

例如，MIB-II、主机资源 MIB 或 Systems Management Empire MIB 中的任何受支持对象（基于 INTEGER）都有效。对象应为整型（包括“counter”、“gauge”、“integer”或“enumerated integer”）。

empireHistoryCtrlObjType

指定代理正在进行采样的 MIB 变量的 ASN1/SNMP 类型。

empireHistoryCtrlBucketsReq

指定请求保存在历史记录表中的离散抽样的数量。根据可用资源，代理将 empireHistoryBucketsGrant 列设置为尽可能接近于该值。

empireHistoryCtrlBucketsGrant

指定代理将保存到历史记录表中的离散抽样的实际数量，用于由该条目定义的数据收集功能。代理保留最新的 BucketsGrant（抽样数）。

empireHistoryCtrlLastCall

指定上次代表该条目获取抽样的时间（基于 sysUptime）。

empireHistoryCtrlCreateTime

指定创建该历史记录采样函数的时间（基于 sysUptime）。

empireHistoryCtrlStatus

指定条目的状态，可以是下列值之一：

- 活动
- notInService
- notReady
- createAndGo
- createAndWait
- 销毁

这些值是在 SNMPv2 SMI RowStatus 文本约定中定义的。

将状态设置为 destroy(6) 将导致代理终止该条目的历史记录采样，并且删除该行以及历史记录表中相应的数据抽样行。

empireHistoryCtrlFlags

表示位字段。设置一个或多个位会更改该条目的行为。

000 000 80—条目由 AIM 插件创建（该标志为只读）

000 001 00—收集性能多维数据

注意：有关历史记录控制表的详细信息，请参阅代理安装的 mib 子目录下的规格 empire.asn1，以及“Systems Management Empire MIB”一章。

历史记录表的列

以下是历史记录表的各列：

注意：有关历史记录表各列的详细信息，请参阅 MIB 规格（代理安装的 mib 子目录下的 empire.asn1）以及“Systems Management Empire MIB”一章。

empireHistoryIndex

标识该抽样所属的历史记录控制表中的行。它与控制表中相应条目的 empireHistoryCtrlIndex 具有相同的值。

empireHistorySampleIndex

指定在与同一历史记录控制条目相关联的所有抽样中唯一标识特定抽样的索引。该索引从 1 开始，并且每存储一个新的抽样就会加 1。

empireHistoryStartTime

指定获取第一个抽样的时间（基于 sysUptime）。

empireHistorySampleTime

指定获取该特定抽样的时间（基于 sysUptime）。

empireHistoryValue

指定在该抽样中获取的 MIB 变量的当前值。

行创建对象

可以将以下 MIB 对象与历史记录表一起使用来优化行创建：

histCtrlUnusedIndex

对变量执行 SNMP Get 时，为历史记录表返回未用过的索引号。

权限：只读

histCtrlMatchDescr

确定与特定条目说明相对应的索引号。执行该 MIB 对象的 SNMP Set，使代理在历史记录表中搜索条目，并放入其说明与 histCtrlMatchIndex MIB 对象相匹配的最后一个条目的索引值。

权限：读写

histCtrlMatchIndex

与 histCtrlMatchDescr 一起使用时，将特定的条目说明与其索引号匹配。

权限：只读

历史记录采样示例

下表显示了历史记录控制表中的条目示例：

索引	Desc	Int	ObjID	类型	存储段 请求	存储段 授予	上次 调用	创建 时间	状态
1	1 分钟 CPU 负载 平均值。	60	loadAverage1Min .0	2	5	5	0d 6:30:0 0	0d 0:0:0	活动
2	内存 使用中	30	memInUse.0	2	10	10	0d 6:30:3 0	0d 0:0:0	活动
3	交换 容量	900	swapCapacity.0	2	48	48	0d 6:30:0 0	0d 2:00:00	活动

历史记录控制表使用 SNMPv2 管理信息结构 (SMI) 行状态文本约定来添加和删除行。

在该表中，行定义如下：

- 控制行 1 使代理每分钟轮询一次 1 分钟负载平均值，并存储最近的 5 个抽样（5 分钟）。
- 控制行 2 使代理每 30 秒轮询一次 memoryInUse MIB 变量，并存储最近的 10 个抽样（5 分钟）。
- 控制行 3 使代理每 15 分钟轮询一次 swapCapacity 变量（使用中的交换百分比），并存储最近的 48 个抽样。通过存储 48 个抽样，管理系统仅需每 12 小时上载一次这些抽样（即，时间间隔为 15 分钟的 48 个抽样涵盖了 12 小时）。

下表显示了作为采样历史记录控制表中所定义的采样配置的结果而存储在历史记录表中的历史记录采样：

索引	SampleIndex	StartTime	SampleTime	值
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

有关使用历史记录控制表的详细信息，请参阅“历史记录收集”一章。

查看历史记录控制表

如果使用 CA eHealth AdvantEDGE View，则可以向系统查询历史记录信息，方法是：从“系统”列表中选择要监控的系统，从“配置”列表中选择“历史记录收集”，然后单击“配置”图标。

注意：有关详细信息，请参阅《CA eHealth AdvantEDGE View Web 帮助》。

历史记录控制表配置

您可以通过在历史记录控制表中添加、删除或修改条目来控制 SystemEDGE 在历史记录表中存储哪些 MIB 对象。

可通过以下一种方法配置历史记录控制表：

- 动态地使用以下两种方法之一：
 - 从管理系统（如 CA eHealth AdvantEDGE View）使用 SNMP 命令来修改表。
 - 在 CA Virtual Assurance 中使用基于文件的配置。

注意：有关详细信息，请参阅“操作期间的动态配置”。

通过在 SystemEDGE sysedge.cf 配置文件中为历史记录控制表指定条目来实现启动时初始化。

注意：有关详细信息，请参阅“启动期间的初始配置”。

启动期间的初始配置

在启动时，代理读取 sysedge.cf 配置文件。您可以使用该文件来指定代理要监控的 MIB 对象，方法是使用 emphistory 指令将相应的条目添加到历史记录控制表中。

操作期间的动态配置

您可以使用 emphistory 命令行实用工具来配置历史记录控制表中的条目，并从历史记录控制表中动态地检索数据抽样。emphistory 实用工具位于代理安装的 bin 子目录中。您也可以手动将历史记录表条目添加到代理配置文件中。

注意：有关如何将历史记录表条目添加到代理配置文件中的详细信息，请参阅“emphistory 实用工具--管理历史记录控制表中的条目”。

emphistory 指令--将条目添加到历史记录控制表中

使用 **emphistory** 指令可以将条目添加到历史记录控制表中。在 **sysedge.cf** 文件中使用该指令及其参数（参数表示历史记录控制表中的列）定义要为历史记录采样收集的 MIB 对象。

在 **sysedge.cf** 文件中使用 **emphistory** 指令将条目添加到历史记录控制表中，如下所示：

```
emphistory index interval objid buckets 'description' flags
```

index

指定代理将在其中创建条目的行号。

interval

指定代理将对对象值进行抽样的时间间隔（以秒为单位）。间隔必须是 30 秒的倍数。

objid

指定代理 MIB 内的应对其值进行抽样的对象实例。您可以使用完整的点分十进制值（例如，1.3.6.1.2.1.25.1.5.0）或符号 MIB 名称（例如，hrSystemNumUsers.0）来指定 OID。在这两种情况下，都必须指定对于非表格 MIB 变量通常为零的对象实例。

存储段

指定代理将在内部存储的存储段数或抽样数。代理存储最近的存储段（抽样数）。当代理获取每个新的抽样时，它会删除最早的抽样。

'description'

指定一个带引号的字符串（长度为 0 到 128 个字符），该字符串表示该条目的说明字段。

empireHistoryCtrlFlags

表示位字段。设置一个或多个位会更改该条目的行为。

000 000 80—条目由 AIM 插件创建（该标志为只读）

000 001 00—收集性能多维数据

flags

表示位字段。设置一个或多个位会更改该条目的行为。

000 000 80—条目由 AIM 插件创建（该标志为只读）

000 001 00—收集性能多维数据

历史记录收集示例

以下条目指示代理每 120 秒通过表索引 15 对对象 MemInUse 的值进行一次抽样，并存储最近的 60 个抽样：

```
emphistory 15 120 memInUse.0 60 'MemInUse history'
```

用于管理历史记录控制表中的条目的 **emphistory** 命令

emphistory 命令指定要从可用的命令列表中执行的命令的类型：**add**、**setstatus**、**delete**、**dump** 和 **list**。

根据命令的类型，可能要求您指定其他参数以完成命令。命令的语法如下所示：

```
add index objid interval buckets "description" flags
```

```
setstatus index status
```

```
delete index
```

```
dump index
```

列表

index

指定代理的历史记录控制表中该条目的行号。

行的索引从 1 开始。当指定转储操作时，代理从与历史记录控制表中索引所标识的条目相对应的历史记录表中检索所有数据抽样。如果为转储操作指定的索引值为 -1，则代理检索历史记录表中的所有内容。

要指定一个未使用的行的特定 MIB 表索引以用于创建行，您必须为 **add** 操作指定一个索引值。

objid

指定代理将在历史记录控制表中抽样和存储的 MIB 变量的完整对象实例标识符。您可以使用完整的点分十进制值（例如，1.3.6.1.2.1.25.1.5.0）或符号 MIB 名称（例如，hrSystemNumUsers.0）来指定 OID。必须提供 OID 的实例部分（即，表示标量的 .0）。对象实例必须存在，且必须为整型（包括“counter”、“gauge”、“integer”或“enumerated integer”）。

interval

指定连续抽样之间的时间间隔（以秒为单位）；该值必须是 30 秒的倍数。

存储段

指定代表该控制条目在历史记录表中保存的离散数据抽样的数量。每个抽样（名为“存储段”）中包含 MIB 变量的快照值、创建抽样的时间、抽样索引，以及与历史记录控制表（该表定义数据收集功能）中的条目相对应的索引。历史记录表存储最近的存储段（抽样数）。

状态

在用于 `setstatus` 操作时，指定用于设置历史记录控制表中行的状态的 `RowStatus` 文本约定值。`Status` 参数可以为以下值之一：

- (1)active
- (2)notInService
- (6)destroy

将行状态设置为 `destroy(6)` 将导致代理中断该条目的历史记录采样，并且在历史记录控制表中删除行，在历史记录表中删除对应的数据采样行。

'description'

（可选）。指定行的说明。如果指定值，则 `emphistory` 使用提供的字符串，而不是默认的历史记录控制条目说明字符串。如果包括了说明，则必须使用单引号 (') 将其括起来。说明的长度必须小于 128 个字符（不包括引号）。过长的字符串将被截短。

标志

表示位字段。设置一个或多个位会更改该条目的行为。

000 000 80—条目由 AIM 插件创建（该标志为只读）

000 001 00—收集性能多维数据

重要信息！ `emphistory` 实用工具的以下用法已被弃用。我们强烈建议使用新的参数，因为将来不会支持以下格式。

```
emphistory add ipaddr[:port] commstr ctrlIndex objid interval buckets  
['description']
```

emphistory 示例

本节提供了一些将 emphistory 实用工具与 SNMP 版本 1 和 3 一起使用的示例。假定代理使用默认端口 161 运行。

示例：列出历史记录控制表中的条目

以下示例列出代理的历史记录控制表的内容：

```
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
```

示例：添加历史记录控制条目

以下示例将表索引 5 处的新控制条目添加到代理的历史记录控制表中。该控制条目指示代理每 60 秒对 ifInOctets.1 MIB 对象实例进行一次抽样，并存储最近的 10 个抽样：

```
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
```

示例：删除历史记录控制条目

以下条目删除表索引 3 处的历史记录控制表条目：

```
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
```

注意：这些条目还指示代理从历史记录表中删除与该控制条目相对应的已存储数据抽样。

示例：设置控制条目的行状态

以下示例禁用历史记录控制表中表索引 5 处的控制条目，但在历史记录表中保存相应的已存储抽样。

```
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

与 RowStatus 文本约定值 notInService(2) 相对应。

示例：检索已存储的数据抽样

以下示例检索与历史记录控制表的第 5 行中定义的数据收集功能相对应的所有已存储数据抽样：

```
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
```

以下示例使用 -1 来检索所有控制条目的所有已存储抽样。该命令检索历史记录表的整个内容：

```
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
```

第 15 章： 自定义 MIB 对象

本章说明如何通过添加自定义 MIB 对象来扩展 Systems Management Empire MIB。

此部分包含以下主题：

[Systems Management Empire MIB 扩展组](#) (p. 375)

[扩展组功能](#) (p. 376)

[扩展变量配置](#) (p. 376)

[扩展示例](#) (p. 378)

[扩展脚本](#) (p. 380)

[在管理软件中使用扩展变量](#) (p. 382)

[针对使用扩展的建议](#) (p. 384)

Systems Management Empire MIB 扩展组

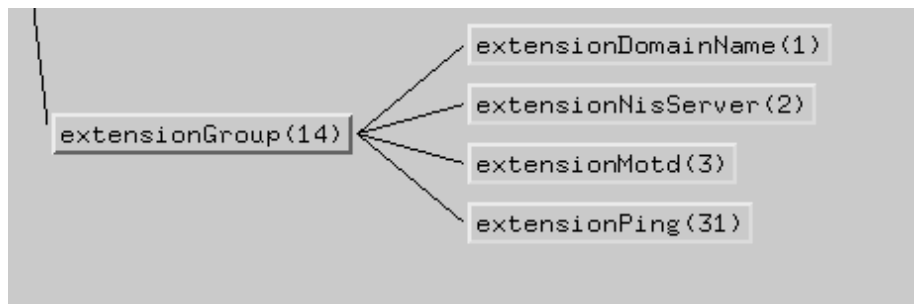
SystemEDGE 代理提供了一个机制，用于扩展 Systems Management MIB 以包括有关系统和应用程序的信息。使用该功能，可以扩展代理来更有效地管理系统环境。您还可以使用 SNMP 设计特定于应用程序的 MIB 变量来管理应用程序，而无需在应用程序源代码内部实现 SNMP 支持。

SystemEDGE 代理通过 Systems Management MIB 的扩展组 (extensionGroup) 来提供该扩展支持。该组中包含未指定的标量 MIB 变量，您可以对其进行配置。为了响应这些变量之一的 SNMP GetRequest，SystemEDGE 代理调用您为变量指定的命令，并返回该命令返回的值。使用 SNMP Set 操作，还可以将参数传递给命令。

在 UNIX 和 Windows 系统上支持扩展组。在 Windows 上，您可能将代理配置为报告 Windows 注册表中可用的性能和配置数据。为了支持该报告功能，SystemEDGE 代理还提供了 Windows 注册表扩展组。

注意：有关该组的详细信息，请参阅“Windows 注册表和性能 MIB 对象”一章。

下图显示了随代理分发的四个扩展变量示例：



扩展组功能

Systems Management MIB 的扩展组位于 OID 1.3.6.1.4.1.546.14。该组为用户定义的 2^{32} 个标量 MIB 变量提供空间。这些新变量的编号为 1 到 2^{32} ，并引用为任何其他标量 MIB 对象。例如，在 Get 和 Set 请求消息中将扩展对象 1 引用为 1.3.6.1.4.1.546.14.1.0。在所有情况下，扩展变量所使用的对象实例标识符均为 .0。

可以指定变量是只读的或可读写的。扩展变量可以是任何有效的 基本 SNMP 类型，包括以下类型：

- 整型
- 计数器
- 标尺
- Octetstring
- TimeTicks
- ObjectId
- IpAddress

注意：有关在启用调试选项的情况下启动代理的详细信息，请参阅“启动代理”一章。

扩展变量配置

可以在 SystemEDGE 代理配置文件 `sysedge.cf` 中配置扩展变量。启动时，代理配置这些值并验证与这些变量关联的程序或命令是否可执行。在配置文件内，关键字 `extension` 定义扩展变量。下一节介绍如何使用该关键字。

extension 关键字--向扩展组中添加条目

可以使用 **extension** 关键字向 Systems Management Empire MIB 扩展组中添加条目。在 `sysedge.cf` 文件中使用该关键字可指定自定义 MIB 对象，使它们可供监控。

使用 **extension** 关键字向扩展组中添加条目，如下所示：

```
extension LeafNumber Type Access 'Command'
```

LeafNumber

指定该条目定义的扩展变量编号（范围是 1 到 2³²）。

类型

指定该条目的 SNMP 类型。支持以下类型：

- 整型
- 计数器
- 标尺
- Octetstring
- TimeTicks
- Objectid
- IPAddress

访问权

指定访问类型（可以为只读或读写）。

'Command'

指定一个长度为 0 到 256 个字符的带引号字符串，该字符串指定应运行的命令（带有任何参数）的完整路径。该命令可以是与平台无关的一个扩展名称，也可以是通过 **Get**、**GetNext** 或 **Set** 请求访问该变量时运行的命令的完整路径。如果指定的是与平台无关的扩展，将在 `CASYSEDGE/extension/<ExtensionName>` 中搜索名为 `<ExtensionName>` 的可执行文件。如果命令文件当前不可访问，将导致该变量的配置失败。

其他参数

除配置文件中作为扩展命令的一部分指定的参数之外，SystemEDGE 代理还传递一些其他参数来帮助您决定如何处理请求。这些参数在您在引号中指定的任何参数之后传递，作为扩展语句的一部分。

以下列表介绍了可以传递给扩展命令的其他参数：

Leafnumber

指定表示该变量的叶编号的整数值（1 到 232）。如果有一个脚本支持多个扩展值，则可以使用该参数来确定正在请求的是哪个变量。

Request-Type

表示 SNMP 请求的类型，可以是以下项之一：

- Get
- GetNext
- 设置

请求类型在被传递时其所有字母均为大写。

Set-Value

指定包含在 Set 请求中传递的值的字符串。在扩展程序中使用该字符串，可以通过将来的 Get 或 GetNext 返回该值的方法来修改该变量的当前值。

扩展示例

SystemEDGE 代理包括几个扩展变量示例。这些示例在 `sysedge.cf` 文件示例中进行定义。实现这些示例的脚本包括在 SystemEDGE 代理安装的 `contrib` 子目录中。也在 Systems Management Empire MIB（代理安装的 `mib` 子目录中的 `empire.asn1`）中对这些变量进行了定义。

重要信息！ 在添加自己的扩展之前，请仔细查看本章和“Systems Management Empire MIB”中的示例。为了清晰起见，这些示例包括一些适当的配置文件扩展命令。

您可以将这些扩展添加到 `sysedge.cf` 文件中，以便它们可供 SystemEDGE 代理使用。

示例：与平台无关的扩展

以下扩展对象在 Unix 上返回命令 `/opt/CA/SystemEDGE/extension/myextension/myextension` 的输出，或在 Windows 上返回 “`C:\Program Files\CA\SystemEDGE\extesnion\myextension\myextension.<EXT>`” 的输出（其中 `<EXT>` 是一个来自环境变量 `%PATHEXT%` 的可执行文件扩展名）：

```
extension 1 OctetString ReadOnly myextension
```

该对象的实例标识符是 1.3.6.1.4.1.546.14.1.0。

示例：远程 ping（UNIX 和 Windows）

使用以下扩展对象可指示 SystemEDGE 代理从正在运行该代理的主机 ping 远程主机。

对于 UNIX 系统，在 `sysedge.cf` 文件中输入以下内容：

```
extension 31 OctetString ReadOnly "/bin/ping www.ca.com"
```

对于 Windows 系统，在 `%SystemRoot%\System32\sysedge.cf` 文件中输入以下内容：

```
extension 31 OctetString ReadOnly "%WINDIR%\system32\ping.exe www.ca.com"
```

该对象在两个示例中的实例标识符都是 1.3.6.1.4.1.546.14.31.0。

扩展脚本

对于扩展脚本的操作，SystemEDGE 代理几乎没有什么约束。但是，它具有以下要求：

- 所有操作（Set、Get 或 GetNext）必须有输出。Set 调用的输出应当回应实际设置的值，而 Get 和 GetNext 的输出应当是对象的值。如果没有输出，SNMP 查询将失败。
- 扩展脚本的输出仅通过第一个换行字符进行解析。如果第一个字符就是换行字符，则认为输出为空，这将导致 SNMP 查询失败并返回错误。

注意：由于 SystemEDGE 代理以 root 或管理员身份运行，请确保所有命令和脚本都使用绝对路径名，经过了充分地调试，且不包含不明确的代码或不必要的选项。

在 Windows 系统上，可以使用批处理文件来编写扩展脚本。代理可以直接运行这些批处理文件。然而，批处理文件的功能非常有限。对于 Windows，可以改用 Perl 和其他脚本语言。

示例脚本测试

创建扩展脚本后，可以在命令行上对其进行测试。以下示例在 **UNIX** 系统上创建一个 **OctetString** 扩展，测试其输出，然后使用 **SNMP** 从脚本返回值。

测试脚本

1. 创建一个名为 `/opt/CA/SystemEDGE/debugext.sh` 的可执行文件，其中包含以下信息：

```
#!/bin/sh
Script=$0
LeafNumber=$1
Operation=$2
SetValue=$3

if [ "$Operation" = "SET" ]; then
echo $Script      > /tmp/debugext_set.txt
echo `date`      >> /tmp/debugext_set.txt
echo $LeafNumber >> /tmp/debugext_set.txt
echo $Operation  >> /tmp/debugext_set.txt
echo $SetValue   >> /tmp/debugext_set.txt
echo $SetValue   > /tmp/debugext_setvalue.txt #SET VALUE HERE
echo $SetValue   #You must echo back what was set
fi

# Override GETNEXT and just do GET
if [ "$Operation" = "GETNEXT" ]; then
Operation="GET"
fi

if [ "$Operation" = "GET" ]; then
echo $Script      > /tmp/debugext_get.txt
echo `date`      >> /tmp/debugext_get.txt
echo $LeafNumber >> /tmp/debugext_get.txt
echo $Operation  >> /tmp/debugext_get.txt
if [ ! -f /tmp/debugext_setvalue.txt ]; then
echo "ERROR: do set first... Script:$Script
LeafNumber:$LeafNumber Operation:$Operation"
exit 0
fi
cat /tmp/debugext_setvalue.txt #return SET VALUE
fi
```

注意：该示例测试为 **OctetString** 的扩展脚本。它对于 **UNIX** 操作系统有效。在 **Windows** 系统上，必须在操作脚本命令中调用解释器。例如，输入 `perl.exe myscript.pl`。

2. 在 `sysedge.cf` 中添加以下行：
`extension 1 OctetString ReadWrite /opt/CA/SystemEDGE/debugext.sh`

注意：该示例测试为 `OctetString` 的扩展脚本。它对于 `UNIX` 操作系统有效。在 `Windows` 系统上，必须在操作脚本命令中调用解释器。例如，输入 `perl.exe myscript.pl`。

3. 在命令行上输入以下内容来创建名为 `myset.txt` 的文件：

```
echo "1.3.6.1.4.1.546.14.1.0 04 debugSetString" > myset.txt
```

您现在有了名为 `myset.txt` 的文件。希望将 `OctetString(04)` `"debugSetString"` 设置为 `OID 1.3.6.1.4.1.546.14.1.0`。

4. 通过输入以下内容，发出 `SNMP Set`：

```
./snmpset -c private -h 127.0.0.1 < myset.txt
```

5. 通过输入以下内容，检索 `debugSetString` 的值：

```
./snmpget -c public -h 127.0.0.1 -o 1.3.6.1.4.1.546.14.1.0
```

6. 通过在命令行上输入以下内容，测试设置：

```
./debugext.sh 1 SET debugSetString2
```

该命令应返回设置为输出的值。

```
./debugext.sh 1 GET
```

该命令应返回在前面的 `SET` 调用中设置的值。

在管理软件中使用扩展变量

将扩展变量并入您的管理系统软件的所有方法都要求编辑和导入 `MIB` 规格文件。本指南不讨论有关如何将 `MIB` 规格文件导入管理系统软件的详细信息，但介绍了并入扩展变量的两个总体策略：

- 编辑 `empire.asn1`，使其包括为您的站点定义的扩展变量。
- 编辑单独的 `MIB` 文件，使其包括为您的站点定义的扩展变量。

如何为扩展变量编辑 `empire.asn1`

按照以下步骤，将自己的扩展变量添加到系统管理 MIB (`empire.asn1`) 中：

1. 创建并调试相关的扩展脚本，然后在代理的配置文件 `sysedge.cf` 中进行配置，以包括这些扩展脚本。
2. 编辑 `empire.asn1`，使其包括 `extensionGroup` 下新增的扩展 MIB 变量。
3. 对 `empire.asn1` 执行编译测试，以确保没有语法错误。该过程特定于您的管理系统和 MIB 编译器。
4. 将新的 `empire.asn1` 文件导入到您的管理系统软件中。该过程特定于您的管理系统和 MIB 编译器。

注意：如果不重新导入 MIB 文件，则您的管理系统软件将无法访问新的扩展 MIB 对象。

如何为扩展变量编辑单独的 MIB 规格

可以将扩展变量放置在单独的 MIB 规格文件中，以便于更新。通过对 MIB 规格文件进行更改可以节省时间和精力，该文件比 `empire.asn1` 文件小得多，从而避免了重新编译和重新加载整个 `empire.asn1` 文件。

向单独的 MIB 规格文件中添加扩展变量

1. 创建并调试相关的扩展脚本，然后在代理的配置文件 `sysedge.cf` 中进行配置，以包括这些扩展脚本。
2. 编辑自己的 MIB 规格文件（例如 `extensions.asn1`），使其包括系统管理扩展组下新增的扩展 MIB 变量。
3. 对 `extensions.asn1` 执行编译测试，以确认没有语法错误。该过程特定于您的管理系统和 MIB 编译器。
4. 将 `extensions.asn1` 文件导入到管理软件中。该过程特定于您的管理系统和 MIB 编译器。

注意：如果不重新导入 MIB 文件，则您的管理系统软件将无法访问新的扩展 MIB 对象。

针对使用扩展的建议

创建 SystemEDGE 扩展时，请遵循以下准则：

- 不要编辑 `empire.asn1`。使用单独的 MIB 规格。
- 使用 SystemEDGE 调试（日志文件监控）来查明输出、插件和扩展方面的问题。
- 扩展条目的长度不应超过 128 个字符，以满足 Windows 对命令行长度的限制。请记住，路径、命令和参数都会计入长度。
- 不要使用批处理文件。改用 Windows 脚本、VBScript、Perl、C、Shell 等。
- 观察新扩展的 `fork/exec` 时间。扩展是同步的，并可能阻止其他操作。

第 16 章：Windows 注册表和性能 MIB 对象

本章说明如何向 Systems Management Empire MIB 中添加 Windows 注册表和性能数据对象。

此部分包含以下主题：

[Windows 注册表和性能功能](#) (p. 385)

[Systems Management Empire MIB ntRegPerf 组](#) (p. 386)

[Windows 注册表和性能变量的配置](#) (p. 389)

[Windows 注册表和性能示例](#) (p. 391)

[在管理软件中使用 Windows 注册表和性能变量](#) (p. 392)

Windows 注册表和性能功能

SystemEDGE 提供强大的机制，用于从 Windows 注册表和性能计数器扩展 Systems Management Empire MIB。该信息包括配置数据（通常使用 Windows 程序 regedit 进行查看）和性能数据（通常使用 Windows 程序 perfmon 进行查看）。

使用该功能，您可以自定义代理，以便返回系统和应用程序的其他配置和性能数据。例如，您可以使用代理来使许多应用程序注册表项可通过 SNMP 提供。此外，您可以访问许多应用程序的性能统计信息。

Systems Management Empire MIB ntRegPerf 组

SystemEDGE 代理在 Systems Management Empire MIB ntRegPerf 组中提供对其他 Windows 注册表和性能参数的支持。Systems Management Empire MIB 的 ntRegPerf 组位于 OID 1.3.6.1.4.1.546.5.7。该组为最多 128 个用户定义的标量 MIB 变量提供空间。这些新变量的编号为 1 到 128，就像任何其他标量 MIB 对象一样被引用。例如，ntRegPerf 对象 1 在 Get 请求消息中引用为 1.3.6.1.4.1.546.5.7.1.0。在所有情况下，ntRegPerf 变量所使用的对象实例标识符均为 .0。作为对这些变量之一的 SNMP Get 请求的响应，代理将读取 Windows 注册表并返回所获取的值。

ntRegPerf 变量可以是任何有效的基本 SNMP 类型，包括以下类型：

- 整型
- 计数器
- 标尺
- OctetString
- TimeTicks
- ObjectId
- IpAddress

注册表数据

SystemEDGE 提供来自标准配置注册表的数据。通过键名和值为该数据编制索引。例如，键 SYSTEM\CurrentControlSet\Control\CrashControl 和值 DumpFile 识别的文本字符串描述了系统转储文件的位置。仅支持 LOCAL_MACHINE 注册表数据。

下表列出了 SystemEDGE 支持的注册表数据类型与首选的 SNMP 类型的对应关系：

REG_DWORD

整型

REG_SZ

OctetString

REG_EXPAND_SZ

OctetString

REG_MULT_SZ

OctetString；仅返回第一个字符串

性能数据

SystemEDGE 通过指定对象和计数器名称，允许访问性能注册表中的性能计数器。例如，对象 NWLink NetBIOS 和计数器 Bytes Total/sec 标识 NetBIOS 字节计数器统计信息。

使用性能计数器时，必须仔细地解释数据。在 Windows 中至少有 27 种不同的计数器类型。大多数计数器实际上并不按照您依据计数器名称推断的格式来报告数据。相反，大多数值都是需要您进行后期处理的原始计数器；例如，将两个抽样之差除以所用时间。如果您看到类似于 Bytes Total/sec 的计数器名称，您真正获得的数据是可以用来计算该比率的原始计数器。

使用该功能之前，应熟悉 Windows 性能计数器。有关详细信息，请参阅 Windows Resource Kit 中有关优化 Windows 的部分。

下表列出了 SystemEDGE 支持的 4 字节性能数据类型与首选的 SNMP 类型的对应关系：

PERF_COUNTER_COUNTER

Counter 或 integer

PERF_COUNTER_RAWCOUNT

Counter、integer 或 gauge

PERF_COUNTER_RAWCOUNT_HEX

Counter、integer 或 gauge

PERF_SAMPLE_COUNTER

Counter、integer 或 gauge

下表列出了 SystemEDGE 支持的 8 字节性能数据类型与首选的 SNMP 类型的对应关系。因为 SNMPv1 仅支持 4 字节值，所以代理仅返回数据中最不重要的 4 个字节。

PERF_COUNTER_TIMER

Counter、integer 或 gauge

PERF_COUNTER_BULK_COUNT

Counter、integer 或 gauge

PERF_COUNTER_LARGE_RAWCOUNT

Counter、integer 或 gauge

PERF_COUNTER_LARGE_RAWCOUNT_HEX

Counter、integer 或 gauge

PERF_COUNTER_TIMER_INV

Counter、integer 或 gauge

PERF_AVERAGE_BULK

Counter、integer 或 gauge

PERF_100NSEC_TIMER

Counter 或 integer

PERF_100NSEC_TIMER_INV

Counter 或 integer

PERF_COUNTER_MULTI_TIMER

Counter 或 integer

PERF_COUNTER_MULTI_TIMER_INV

Counter 或 integer

PERF_100NSEC_MULTI_TIMER

Counter 或 integer

PERF_100NSEC_MULTI_TIMER_INV

Counter、integer 或 gauge

PERF_ELAPSED_TIME

整型

PERF_RAW_FRACTION

Integer 或 gauge

不支持的性能数据类型

带有多个抽样和内部数据的几个计数器很难用单个值表示，需要大量后期处理。因此，CA SystemEDGE 不支持以下计数器类型：

- 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

Windows 注册表和性能变量的配置

在 `sysedge.cf` 文件中配置 Windows 注册表和性能变量。SystemEDGE 在启动时配置这些值，并验证与每个变量关联的值是否可访问。如果不可访问，它会将错误消息输出到 `sysedge.log`，且不会创建 MIB 对象。

ntregperf 关键字—添加 Windows 注册表和性能 MIB 对象

在 `sysedge.cf` 文件中，使用 `ntregperf` 关键字为 `ntRegPerf` 组定义 Windows 注册表和性能 MIB 对象。

可以使用 `ntregperf` 关键字在 `ntRegPerf` 组中添加条目，如下所示：

注册表数据：

```
ntregperf LeafNumber Type Registry 'Key' 'Value'
```

性能数据：

```
ntregperf LeafNumber Type Performance 'Object' 'Counter' 'PerfInstance'
```

LeafNumber

定义 `ntRegPerf` 变量编号，范围为 1 到 128。

类型

指定该条目的 SNMP 类型，可以是以下项之一：

- 整型
- 计数器
- 标尺
- Octetstring
- TimeTicks
- Objectid
- IPAddress

注册表

选择配置注册表条目。

性能

选择配置性能条目。

'Key'

指定长度为 0 到 512 个字符的带引号字符串，该字符串指定要访问的注册表项。

'Value'

指定长度为 0 到 128 个字符的带引号字符串，该字符串指定要访问的注册表项。

'Object'

指定长度为 0 到 512 个字符的带引号字符串，该字符串指定要访问的性能对象。

'Counter'

指定长度为 0 到 128 个字符的带引号字符串，该字符串指定要访问的对象性能计数器值。

'PerfInstance'

指定要访问的性能计数器实例（应等效于 Windows perfmon 实用工具中列出的实例）。

Windows 注册表和性能示例

SystemEDGE 代理包括几个 ntRegPerf 变量示例。这些示例在 sysedge.cf 文件示例中进行定义。在添加自己的 ntRegPerf 扩展之前，请仔细阅读 Systems Management Empire MIB（代理安装的 mib 子目录中的 empire.asn1）中的这些示例及其定义。为了清晰起见，这些示例包括一些适当的配置文件 ntRegPerf 命令。

示例：CrashControl DumpFile

以下 ntRegPerf 对象返回转储文件的路径：

```
ntregperf 1 OctetString Registry  
'SYSTEM\CurrentControlSet\Control\CrashControl' 'DumpFile'
```

该对象的对象实例标识符是 1.3.6.1.4.1.546.5.7.1.0。

示例：Total Number of Threads

下面的 ntRegPerf 对象返回系统中当前可用的线程的总数：

```
ntregperf 2 Gauge Performance 'Objects' 'Threads' '1'
```

该对象的对象实例标识符是 1.3.6.1.4.1.546.5.7.2.0。

示例：TCP Segments Sent/Sec

下面的 ntRegPerf 对象返回系统传输的 TCP 段的总数：

```
ntregperf 3 Counter Performance 'TCP' 'Segments Sent/sec' '1'
```

该对象的对象实例标识符是 1.3.6.1.4.1.546.5.7.3.0。

在管理软件中使用 Windows 注册表和性能变量

有几种方法可以将 Windows 注册表和性能变量并入您的管理系统软件，所有这些方法都要求编辑和导入 MIB 规格文件。有关如何将 MIB 规格文件导入到管理系统软件中的详细信息超出了本指南的范围，但本指南介绍了并入 Windows 注册表和性能变量的两个总体策略：

- 编辑 `empire.asn1`，使其包括为您的站点定义的变量。
- 编辑单独的 MIB 文件，使其包括为您的站点定义的变量。

如何为 ntRegPerf 变量编辑 empire.asn1

按照以下过程，可以将自己的 ntRegPerf 变量添加到 Systems Management Empire MIB 中：

1. 在 `sysedge.cf` 文件中创建并调试相关的 ntRegPerf 条目以包括它们。
2. 编辑 `empire.asn1`，以包括新的 ntRegPerf MIB 变量。
3. 对 `empire.asn1` 执行编译测试，以确保没有语法错误。该过程特定于您的管理系统及其相应的 MIB 编译器。
4. 将新的 `empire.asn1` 文件导入到您的管理系统软件中。该过程特定于您的管理系统和 MIB 编译器。

注意：如果不重新导入 MIB 文件，则您的管理系统软件将无法访问新的 MIB 对象。

如何为 ntRegPerf 变量添加单独的 MIB 规格

可以将 ntRegPerf 变量放在单独的 MIB 规格文件中，以避免直接处理 MIB 文件。通过对 MIB 规格文件进行更改可以节省时间和精力，该文件比 `empire.asn1` 文件小得多。这样就可以避免重新编译和重新加载整个 `empire.asn1` 文件。

要为 ntRegPerf 变量添加单独的 MIB 规格，请完成以下过程：

1. 编辑自己的 MIB 规格文件（例如 `ntregperf.asn1`），使其包括 Systems Management Empire MIB `ntregperfGroup` 下新增的 ntRegPerf MIB 变量。
2. 对 `ntregperf.asn1` 文件执行编译测试，以确保没有语法错误。该过程特定于您的管理系统及其相应的 MIB 编译器。

3. 将 ntregperf.asn1 文件导入到您的管理系统软件中。该过程特定于您的网络管理站及其相应的 MIB 编译器。

注意：如果不重新导入 MIB 文件，则您的管理系统软件将无法访问新的 MIB 对象。

第 17 章： SystemEDGE AIM

AIM 是将其他监控功能添加到基本 SystemEDGE 中的插件模块。本章介绍如何配置与代理一起提供的 AIM。有关 AIM 体系结构的详细信息，请参阅“概念”一章。

此部分包含以下主题：

[核心 AIM](#) (p. 395)

[顶级进程 AIM](#) (p. 395)

[监控维护 Windows AIM](#) (p. 395)

[性能多维数据集 AIM](#) (p. 397)

核心 AIM

monwin 和 perfcube AIM 是 SystemEDGE 核心 AIM。使用 SystemEDGE 核心软件包安装核心 AIM。它们不需要 sysedge.cf 文件中的 sysedge_plugin 语句。如果需要此功能，SystemEDGE 将自动加载 SystemEDGE 核心 AIM 库。

- 如果在 sysedge.cf 文件中找到 monwin 条目，则代理将加载监控维护 Windows AIM。
- 如果 sysedge.cf 文件中存在具有收集标志 (0x100) 的 emphistory 条目，则代理将加载性能多维数据集 AIM。

CA SystemEDGE 5.0 之前的版本包括名为“顶级进程 AIM”的 AIM。在 CA SystemEDGE 5.0 中，顶级进程 AIM 的功能集成到了核心代理中。因此，该 AIM 不再是可加载的插件。

顶级进程 AIM

上一个顶级进程 AIM 的功能包含在 SystemEDGE 核心中。在 sysedge.cf 文件中激活“no_topprocs_table”指令时，禁用顶级进程功能。

监控维护 Windows AIM

使用受监控的 Windows AIM 对 SystemEDGE 监控的活动进行基于时间的控制。此 AIM 应用于自主监控、进程监控、进程组监控、日志文件监控、历史记录以及 Windows 事件监控（仅 Windows）。当不亲自停止监视器不重要时，使用受监控的 Windows AIM 停止从这些监视器生成陷阱（警报）。

启用监视器维护窗口 AIM

要启动监视器维护窗口 AIM，必须在 `sysedge.cf` 文件中指定相应的条目。
SystemEDGE 执行热启动并自动启动 AIM。

以下条目指定监视器的监控（维护）时间段：

语法：

```
monwin monitor index timeOff timeOn status
```

monitor

指定监视器表（selfmon、processmon、procgrouppmon、nteventmon、logfilemon、emphistory）。

index

指定监视器的索引。

timeOff

以 24 小时格式 (hhmm) 指定关闭监视器（notInService）的时间。维护时间开始。

timeOn

以 24 小时格式 (hhmm) 指定打开监视器 (active) 的时间。维护时间结束。

状态

指定该条目为 active (1) 还是 notInService (2)。

示例

```
monwin processmon 12 2200 2330 1
```

监视器维护窗口 AIM 的限制

受监控的 Windows AIM 每天仅运行一次。不支持一周内某天、一月中某天或其他非每天一次的监控控制。

忽略 AIM 配置文件中已指定，但监视器表中实际不存在的任何监视器，直至该监视器存在。从监视器表中删除监视器索引时，忽略 AIM 用于运行定时控制的监视器。

AIM 尝试在指定的具体时间运行监视器激活和取消激活。但是，在具有重要负载的系统上，AIM 可能无法准时体验循环。因此，所有行的结束时间和开始时间至少应该相隔 1 分钟。

任何单个监视器都可以成为 AIM 配置文件中多个行的目标。也就是说，一天之内可以多次打开和关闭监视器。由您来避免此案例中因关闭和打开时间重叠而导致的逻辑问题。

性能多维数据集 AIM

性能多维数据集 AIM 读取历史记录表条目，计算多维数据集收集间隔（默认 = 20 分钟）内的平均度量标准，并将平均度量标准值写入多维数据集文件（性能数据文件）。AIM 使用现有 SystemEDGE 历史记录表定义收集哪些度量标准，还计算度量标准平均值。

启用性能多维数据集 AIM

如果 sysedge.cf 文件中至少存在一个具有收集标志 (0x100) 的 emphistory 条目，则 SystemEDGE 将隐式加载性能多维数据集 AIM。AIM 不要求 sysedge.cf 中存在 sysedge_plugin 条目。

语法

```
emphistory index interval oid number ['descr' [flags]]
```

index

指定要使用的表索引。

interval

指定收集时间间隔（30 秒的倍数）。

oid

指定要收集的对象的 OID（采用圆点表示法的完整 OID 或 *attrName.instOID*）。

number

指定要收集的抽样数。

descr

指定任意说明。

flags

如果指定为 0x100，将启用性能多维数据集 AIM。

示例

```
emphistory 11 60 loadAverage1Min.0 60 '1-min load average history' 0x100
```

第 18 章： 命令行实用工具

本章介绍了 SystemEDGE 命令行实用工具的用法。

此部分包含以下主题：

[SystemEDGE 命令行实用工具](#) (p. 399)

[SystemEDGE 其他命令行实用工具](#) (p. 400)

[配置 SystemEDGE 命令行实用工具](#) (p. 400)

SystemEDGE 命令行实用工具

如果在未受管模式下运行 SystemEDGE，则可以使用 SystemEDGE 命令行实用工具来配置参数，以便操作 CA SystemEDGE 代理。

SystemEDGE 命令行实用工具在 IPv4 和 IPv6 环境中均可使用 SNMPv1、SNMPv2c 和 SNMPv3 进行通信。这些实用工具驻留在 SystemEDGE 安装目录的 bin 子目录中。

这些实用工具传递的信息应与代理的配置文件（sysedge.cf 和 sysedgeV3.cf）中的信息相匹配。有关 SystemEDGE 配置文件的详细信息，请参阅“代理配置”一章。

代理支持以下 SNMP 命令行实用工具：

- diagsysedge
- edgemon
- edgewatch
- emphistory
- se_enc
- sendtrap
- snmpget
- snmpset
- sysvariable
- walktree
- edgetrapmon

SystemEDGE 其他命令行实用工具

SystemEDGE 提供不基于 SNMP 的其他命令行实用工具。这些实用工具与 SystemEDGE 监视器一起充当操作项。当监视器表条目求值为 true 时操作运行，或者，对于有状态的监视器，在阈值违反导致相关受管对象的状态更改时运行操作。

代理支持以下其他命令行实用工具：

- bounce (Windows)
- checkfile
- 电子邮件
- getver (Windows)
- restartproc.exe (Windows)
- restartproc.sh (UNIX/Linux)
- restartsvc (Windows)

配置 SystemEDGE 命令行实用工具

通过访问 SystemEDGE 命令行实用工具来配置参数，以便操作 CA SystemEDGE 代理。

遵循这些步骤：

1. 使用管理员权限登录到受管节点。
2. 打开命令提示符窗口并转到 Install_Dir/SystemEDGE/bin 目录。

现在您可以在命令提示符中输入需要的 SystemEDGE 命令。

注意：有关 SystemEDGE 命令行实用工具的详细信息，请使用控制台帮助选项 (-h)。例如，要了解 edgemon，请在命令提示符中输入以下命令：
edgemon -h

第 19 章： 使用状态管理的示例

以下示例使用 SystemEDGE 的典型状态管理功能。

此部分包含以下主题：

[监控 Windows 任务排定程序](#) (p. 401)

[监控隔离文件是否存在](#) (p. 403)

[监控文件系统](#) (p. 403)

[通过 SNMP 重新配置扩展脚本](#) (p. 405)

[监控日志文件并通过电子邮件发送通知](#) (p. 409)

[监控进程](#) (p. 410)

[创建受监控的对象](#) (p. 410)

监控 Windows 任务排定程序

此示例监控 Windows 上失败或过度任务的任务排定程序。它由 sysedge.cf 文件的一组条目以及关联的 schedlgu.cmd 脚本组成。schedlgu.cmd 脚本对 *.job 文件进行计数，分析 %SystemRoot%\Tasks 目录中的 *.txt 文件，并将结果写入 %CASYSEDGE_DATA%\schedlgu.log 文件。SystemEDGE 启动脚本并监控 schedlgu.log 文件以评估当前状态。

将 schedlgu.cmd 脚本复制到 %CASYSEDGE%\bin 目录，或将其添加到将其发送到 %CASYSEDGE%\bin\managedscripts 目录的配置部署软件包中。

sysedge.cf 的条目

```
# Windows sample configuration for monitoring the Task Scheduler for failed or
excessive tasks
extension 55 integer readonly 'schedlgu.cmd %CASYSEDGE_DATA%\schedlgu.log'
watch logfile 55 0x0 '%CASYSEDGE_DATA%\schedlgu.log' 'The task completed with an
exit code of [()][0]*[^\0].*[]]' 'Windows Task Scheduler Job Execution Errors' ''
10 warning
watch logfile 56 0x0 '%CASYSEDGE_DATA%\schedlgu.log' 'WARNING' 'Windows Task
Scheduler Warnings' '' 10 major
watch logfile 57 0x0 '%CASYSEDGE_DATA%\schedlgu.log' 'ERROR' 'Windows Task
Scheduler Errors' '' 10 critical
monitor oid extensionGroup.55.0 55 0x100 600 absolute > 15 'Number of Scheduled
Tasks > 15' '' 'SysHealth' 'Scheduled Tasks' 'Count' 'minor'
```

扩展脚本 schedlgu.cmd

```
@echo off

REM SystemEDGE extension script for monitoring Windows Task Scheduler

setlocal

set SLOG=.\schedlgu.log
if "%1" NEQ "" set SLOG=%1

set SDIR=%SystemRoot%\Tasks

set SREC=
set DONE=N

REM Logical records in the Task Scheduler log file span multiple lines.
REM Convert into single line format (as %SLOG%) for monitoring via SystemEDGE.
if exist %SLOG% del %SLOG%
for /f "tokens=*" %i in ('type "%SDIR%\schedlgu.txt"') do set
LINE=%i&call :parseline

REM Return number of configured tasks
set COUNT=0
for /f "tokens=*" %i in ('dir /b "%SDIR%\*.job"') do call :countline
echo %COUNT%

endlocal

goto :EOF

:countline
set /a COUNT+=1
goto :EOF

:parseline
REM Replace double quotes with pound sign
set LINE=%LINE:"=#%
REM Check if we have already processed the most recent record in the log
if "%DONE%" EQU "Y" goto :EOF
REM Check for start of next logical record
if "%LINE:~0,1%" EQU "#" goto :startrecord
if "%LINE%" EQU "[ ***** Most recent entry is above this line ***** ]" set
DONE=Y&goto :startrecord
REM Concatenate lines for current record
set SREC=%SREC% %LINE%
goto :EOF
```

```

:startrecord
if "%SREC%"="" goto :initrecord
REM Restore double quotes
set SREC=%SREC:#= "%
REM Write current record
@echo %SREC%>>"%SLOG%"

:initrecord
set SREC=%LINE%

```

监控隔离文件是否存在

此示例监控是否存在由 %ProgramFiles%\CA\eTrustITM\Move 目录中的 eTrust Antivirus 每十分钟隔离一次的文件。此示例包含 sysedge.cf 文件的两个条目，并使用日志文件参数监控目录中的条目。0x1808 标志指定递归监控目录，但不记录任何陷阱。

```

# Windows sample configuration for monitoring the existence of any files
quarantined by eTrust Antivirus
watch logfile 45 0x1808 '%ProgramFiles%\CA\eTrustITM\Move' '' 'Monitor eTrust
Antivirus Quarantine directory' '' 10 warning
monitor logMonitorEntry '%ProgramFiles%\CA\eTrustITM\Move'
logMonitorLogFileCount 45 0x100 600 absolute > 0 'eTrust Antivirus Quarantine
directory not empty' '' '' '' '' 'major'

```

监控文件系统

此示例监控 Windows 上的文件系统，且包含 sysedge.cf 文件的三组条目：

- 因不同阈值而具有不同重要级别的对象状态模型的用法。
- 与相同类、实例、属性和重要级别组合的标志 0x800 的用法。标志 0x800 指定所有具有相同重要级别的监视器的合计状态并应用 AND 关系。
- aggregate_level 控制的用法。

注意：此示例中省略了默认情况下添加到对象实例的前缀 //./。

```
# Derives a state for file system C: (System).
# Creates an aggregate 'FSys' 'System' 'PercentUsed' , which can be in state
ok, minor, major, or fatal.

monitor devTableEntry 'C:' devCapacity 51 0x0 120 absolute > 90 'Description' ''
'FSys' 'System' 'PercentUsed' minor
monitor devTableEntry 'C:' devCapacity 52 0x0 120 absolute > 95 'Description' ''
'FSys' 'System' 'PercentUsed' major
monitor devTableEntry 'C:' devCapacity 53 0x0 120 absolute > 99 'Description' ''
'FSys' 'System' 'PercentUsed' fatal

# Derives warning state if three conditions are met simultaneously, that is, only
if all three Backup file systems E:, F:, and G: are above 95%.
# Creates an aggregate 'FSys' 'Backup' 'PercentUsed' , which can be in state
ok or warning.

monitor devTableEntry 'E:' devCapacity 54 0x800 120 absolute > 95 'Description'
'' 'FSys' 'Backup' 'PercentUsed' warning
monitor devTableEntry 'F:' devCapacity 55 0x800 120 absolute > 95 'Description'
'' 'FSys' 'Backup' 'PercentUsed' warning
monitor devTableEntry 'G:' devCapacity 56 0x800 120 absolute > 95 'Description'
'' 'FSys' 'Backup' 'PercentUsed' warning

# Derives worst state of all file systems.
# Creates a super aggregate 'FSys' '*' 'PercentUsed' , which can be in state
ok, warning, minor, major, or fatal.

aggregate_level 0x0002
```

通过 SNMP 重新配置扩展脚本

这些脚本允许您使用扩展脚本 ping 任意系统，您可以使用 SNMP 远程配置该扩展脚本。因此，不需要编辑 sysedge.cf 或任何其他自定义脚本。可以使用 MIB 浏览器中的 SNMP SET 或 SystemEDGE 附带的事件 snmpset 实用工具设置或更改要查看的系统的名称。读取扩展变量提供字符串以及 ping 信息。

通过 SNMP 重新配置扩展脚本

1. 将以下行添加到 sysedge.cf 配置文件中并重新启动 SystemEDGE:

```
extension 31 octetstring readwrite '/opt/CA/test/ping.sh' (UNIX, Linux)
extension 31 octetstring readwrite 'c:\test\ping.bat' (Windows)
```

2. 将脚本 (ping.sh, ping.bat) 复制到扩展指令中指定的目录。

3. 输入读取请求，但不设置 ping 目标:

```
C:\SystemEDGE\bin>.\snmpget -o 1.3.6.1.4.1.546.14.31.0
1.3.6.1.4.1.546.14.31.0 Ping target must be set first
```

此时将显示错误消息。

4. 将 my-sles3 设置为要 ping 的系统:

```
C:\SystemEDGE\bin>.\snmpset -c private -o 1.3.6.1.4.1.546.14.31.0 -s
my-sles3
1.3.6.1.4.1.546.14.31.0 my-sles3
```

5. 通过读取变量来 ping 系统:

```
C:\SystemEDGE\bin>.\snmpget -o 1.3.6.1.4.1.546.14.31.0
1.3.6.1.4.1.546.14.31.0 my-sles3 is alive
```

此脚本检查扩展是否具有索引 31。但是技术上不需要此检查，可以将其删除，因此扩展脚本可以分配给任何索引。

脚本 ping.bat 和 ping.sh 将配置信息存储在临时文件中。

UNIX 和 Linux 的 ping 脚本

```
#!/bin/sh
#
# ping.sh example extension script
#
# A simple example script for the SystemEDGE agent extension group.
# This script implements a "Ping" SNMP MIB variable whereby
# Sets to this variable indicate what system should be pinged
# and Gets to this variable cause the Ping to occur and the
# output to be returned as an OctetString.
#
# For reference, the following are passed as parameters by the SystemEDGE
# Agent. They are not used in this script.
#
# The Extension Group Leaf Number
LeafNumber=$1
# The SNMP Operation: GET, GETNEXT, or SET
Operation=$2
# The value if a SET Operation
SetValue=$3

# determine which ping program to use
OSNAME=`/bin/uname -r`
OSTYPE=`/bin/uname -s`

if [ "$LeafNumber" = "31" ]; then

    # setup who we should be pinging
    if [ "$Operation" = "SET" ]; then
        echo $SetValue > /tmp/sysedge.ping.target
        echo $SetValue
    fi

    if [ "$Operation" = "GETNEXT" ]; then
        Operation="GET"
    fi

    if [ "$Operation" = "GET" ]; then
        if [ ! -f /tmp/sysedge.ping.target ]; then
            echo "Ping target must be Set first"
            exit 0
        fi
    fi
fi
```

```

case $OSTYPE in
    Linux)
        /bin/ping -c 1 -q -n `/bin/cat /tmp/sysedge.ping.target`
        ;;

    HP-UX)
        /etc/ping `/bin/cat /tmp/sysedge.ping.target` -n 1
        ;;

    AIX)
        /etc/ping -c 1 `/bin/cat /tmp/sysedge.ping.target`
        ;;

    SunOS)
        /usr/sbin/ping `/bin/cat /tmp/sysedge.ping.target` 2
        ;;

    *)
        case $OSNAME in
            6.*)
                /usr/etc/ping -c 1 -q -n `/bin/cat /tmp/sysedge.ping.target`

                # IRIX ping is very annoying since it doesnt appear you can get
                # one-line output indicating if the machine is alive or not
                alive.

                # If we were trickier shell programmers, we'd catch the output
                # and only echo the last line.
                ;;

            5.*)
                /usr/sbin/ping `/bin/cat /tmp/sysedge.ping.target` 2
                ;;

            4.*)
                /usr/etc/ping `/bin/cat /tmp/sysedge.ping.target` 2
                ;;

            *)
                echo "Cant determine ping command for system " $OSNAME
                exit 0
                ;;
        esac
    esac
fi
fi

```

Windows 的 ping 脚本

```
@ECHO OFF
@rem ping.bat example extension script
@rem
@rem A simple example script for the SystemEDGE agent extension group.
@rem This script implements a "Ping" SNMP MIB variable whereby
@rem Sets to this variable indicate what system should be pinged
@rem and Gets to this variable cause the Ping to occur and the
@rem output to be returned as an OctetString.
@rem
@rem For reference, the following are passed as parameters by the SystemEDGE
@rem Agent.
@rem

@rem The Extension Group Leaf Number
set LeafNumber=%1%

@rem The SNMP Operation: GET, GETNEXT, or SET
set Operation=%2%

@rem The value if a SET Operation
set SetValue=%3%

@rem Which ping command should we use
set PING=%SystemRoot%\system32\ping.exe

@rem We store our target in the "environment" for future use
@rem using variable SYSEGE_PING_TARGET

if not %LeafNumber% == 31 goto fi

    @rem if set, we want to ensure there is a value,
    @rem then store it away, then echo the value to STDOUT
    @rem so as to not confuse extension code
    if %Operation% == SET if !%SetValue%==! goto error
    if %Operation% == SET @echo set
    SYSEGE_PING_TARGET=%SetValue% > %TMP%\pingtgt.bat
    if %Operation% == SET echo %SetValue%
    if %Operation% == SET goto fi
```



```
@rem Run the script to recover the target name
call %TMP%\pingtgt.bat
if !%SYSEDGE_PING_TARGET% ==! goto error

@rem echo pinging %SYSEDGE_PING_TARGET%
%PING% -n 1 -w 250 %SYSEDGE_PING_TARGET% > nul

if ERRORLEVEL 1 echo %SYSEDGE_PING_TARGET% not alive
if not ERRORLEVEL 1 echo %SYSEDGE_PING_TARGET% is alive

:fi
goto exit

:error
echo Ping target must be set first

:exit
rem cleanup because command.com is really pathetic, sigh
set PING=
set Operation=
set LeafNumber=
set SetValue=
```

监控日志文件并通过电子邮件发送通知

此示例监控 Windows 修补程序日志文件是否出现致命字符串。如果出现致命字符串，SystemEDGE 将通过电子邮件发送通知。此示例由 `sysedge.cf` 文件的条目以及关联的 `Send_Notification_Mail.cmd` 脚本组成。

注意： `#no_actions` 指令不得处于活动状态。可在其前面放置注释 (`#`) 字符。

```
watch logfile 44 0x8 '%windir%\KB*.log 'fatal' 'check KB for fatal occurrence in
MS hotfix log' "%CASYSEDGE%\My_Scripts\Send_Notification_Mail.cmd" 1 warning
```

脚本 `Send_Notification_Mail.cmd`

```
rem *** Get arguments in which we are interested ***
set regexpr=%3
set lasttime=%5
set description=%6

rem *** Shift to get argument %10 as %9 ***
shift
set current_file=%9

rem *** Call email ***
"c:\Program Files\CA\SystemEDGE\bin\email.exe" -r smtp.company.com sysedge@%
COMPUTERNAME% you@company.com 'KB log fatal found' ' File monitor %description%
activated. %regexpr% pattern found in %current_file% at time %lasttime%'
```

监控进程

监控包含参数的进程（注意标志值）并创建受管对象：类=notepad、实例=bad、属性=alive、状态=severity

```
watch process procAlive 'notepad.*bad.*' 122 0x800 30 absolute nop 0 'Description:
procAlive(notepad)' ''notepad' 'bad' 'alive' 'minor'
```

创建 **procgrou** 来监控一组进程的属性。可以与具有监视器指令的其他 OID 一样，监控该 **procgrou** 的属性（如 RSS=Real memory Resident Set）。

```
watch procgroup 'svchost' 222 0x0 30 'Descr: procgroup(svchost)' ''
monitor processGroupMonEntry svchost pgmonRSS 20 0x0 60 absolute > 15000 'Descr:
procgroup svchost' '' ProcHealth svchost pgmonRSS critical
```

创建受监控的对象

检查是否安装了特定修补程序，如果已安装，则创建受监控的对象。

```
monitor hrSWInstalledEntry 'Microsoft Hotfix Patch KB958644' hrSWInstalledIndex
41 0x100 30 absolute = 0 'Descr: Hot' '' 'SysHealth' 'SWInstalled' 'Microsoft
Hotfix Patch KB958644' 'ok'
```

创建一个 MIB 属性，该属性具有 PerfObject “UDPv4” 中的性能计数器 “Datagrams Sent/sec” 的值。

注意：使用 Microsoft 命令 **perfmon** 可检查可用性能对象和计数器。

```
ntregperf 2 counter performance 'UDPv4' 'Datagrams Sent/sec' ''
```

第 20 章：故障排除和用法建议

本章提供使用 SystemEDGE 代理的帮助提示。

在 Windows 上，Windows 控制面板作为中央位置来

- 查看代理基础知识：系统信息和 SNMP 团体
- 查看并修改代理配置文件
- 查看代理诊断文件
- 启动和停止代理

此部分包含以下主题：

[使用 diagsysedge.exe](#) (p. 411)

[常见问题](#) (p. 413)

使用 diagsysedge.exe

可以使用 diagsysedge.exe 程序来验证代理正在运行，并获得有关可用于故障排除的代理的信息

本节中的示例假设代理配置为使用 SNMPv1 或 SNMPv2c，代理具有公共的读团体，且代理位于端口 161 或 1691 上。

注意：有关 diagsysedge.exe 支持的所有选项的详细信息，请参阅“命令行实用工具”这一章。

使用 diagsysedge 确定代理是否正在运行

可以运行使用 -B 选项的 diagsysedge.exe 来确定代理是否正在运行。

确定代理是否正在运行

1. 打开命令提示符并切换到 SystemEDGE 的 bin 目录。

2. 输入以下内容:

UNIX:

```
./diagsysedge -B
```

注意: 除了 diagsysedge 外, 也可使用 UNIX 上的符号链接 diagsysedge.exe 来运行该实用工具。

Windows:

```
diagsysedge.exe -B
```

此命令的输出如下所示:

```
#./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
```

如果代理未在运行, 将收到说明 SNMP 操作失败的错误消息。检查端口号, 如果代理在除 161 以外的端口上启动, 则在命令行上指定 **-p 端口** 并重试, 或者可以使用“启动代理”这一章中的说明再次尝试启动代理。

获取故障排除报告

如果使用 SystemEDGE 时遇到问题, 技术支持可能会请求您运行 diagsysedge.exe 来生成可用于故障排除的报告。

生成可发送给技术支持的报告

1. 打开命令提示符并切换到 sysedge/bin 目录。
2. 输入以下内容:

UNIX:

```
./diagsysedge
```

注意: 除了 diagsysedge 外, 也可使用 UNIX 上的符号链接 diagsysedge.exe 来运行该实用工具。

Windows:

使用 Windows 控制面板创建 diagsysedge 输出。

如果代理未在运行, 将收到说明 SNMP 操作失败的错误消息。使用本章中关于操作系统的说明再次尝试启动代理。

验证 AIM 插件

可以运行使用 -Q 选项的 `diagsysedge.exe` 来确定 AIM 是否正在运行。

确定 AIM 是否正在运行

1. 打开命令提示符并切换到 SystemEDGE 的 bin 目录。
2. 输入以下内容：

UNIX/Linux:

```
./diagsysedge -Q
```

注意：除了 `diagsysedge` 外，也可使用 UNIX 上的符号链接 `diagsysedge.exe` 来运行该实用工具。

Windows:

```
diagsysedge.exe -Q
```

此命令的输出如下所示：

```
C:\Program Files\CA\SystemEDGE\bin>diagsysedge -Q
--- diagsysedge plugin query result ---
monwin : RESULT: ok BRANCH: 1.3.6.1.4.1.546.16.20 DESCR: CA SystemEDGE AIM
for Monitor Timing Windows, VERSION:5.0.0 BUILD:10092 REQUIRES:5.0.0
perfcube : RESULT: ok BRANCH: 1.3.6.1.4.1.546.16.21 DESCR: CA SystemEDGE AIM
for Performance Cubes, VERSION:5.0.0 BUILD:10092 REQUIRES:5.0.0
cavmvcaim : RESULT: ok BRANCH: 1.3.6.1.4.1.546.16.52 DESCR: CA VMware VC
Server AIM 2.0 PL0
rmonwbem : RESULT: ok BRANCH: 1.3.6.1.4.1.546.16.22 DESCR: SysEDGE RMONWBEM
AIM, VERSION:1.0.0 BUILD:10102 REQUIRES:5.0.0
```

```
C:\Program Files\CA\SystemEDGE\bin>
```

如果代理未在运行，将收到说明 SNMP 操作失败的错误消息。检查端口号，如果代理在除 161 以外的端口上启动，则在命令行上指定 -p 端口并重试，或者可以使用“启动代理”这一章中的说明再次尝试启动代理。

常见问题

本节介绍安装和使用 SystemEDGE 时可能出现的问题。还提供解决这些问题的说明。本节说明问题的表现症状以及找出问题原因要执行的验证步骤。

故障排除的诊断日志记录信息

如果使用 SystemEDGE 时遇到问题，需要通过编辑 `sysedge.cf` 文件来调整诊断日志记录设置。“配置诊断日志记录机制”这一章介绍了编辑 `sysedge.cf` 文件的步骤。

记录的信息量通常不断增加，必须调整日志文件最大大小和日志文件数目。

示例：

```
sysedge_logfile sysedge.log 20000 2
sysedge_loglevel debug
```

指导 SystemEDGE 使用两个其他日志文件，每个文件的最大大小为 20000 KB。

代理未响应 SNMP 请求

代理未响应 SNMP 请求。要解决此问题，必须执行以下步骤来找出原因：

- [验证代理是否正在运行](#) (p. 415)
- [验证代理是否在系统初始化时启动](#) (p. 415)
- [验证代理是否响应查询](#) (p. 416)
- 验证是否已安装代理二进制文件和配置文件
- 验证代理是否正确地与其他代理共存
- [验证代理配置](#) (p. 417)
- [验证管理软件配置](#) (p. 417)

验证代理是否正在运行

可以验证代理是否以下列方法之一运行，具体取决于平台。

验证代理是否正在 UNIX 上运行

1. 输入 `netstat -a` 并查找 UDP/161、UDP/1691 或 SNMP。
2. 输入 `ps -aux` 或 `ps -aef` 并查找 `sysedge`。
3. 运行 `walktree` 或 `snmpget`。
4. 检查代理错误消息的 `sysedge.log` 文件。

验证代理是否正在 Windows 上运行

1. 选择“开始”、“控制面板”、“CA SystemEDGE”并查看状态。
2. 打开 Windows 任务管理器并选择“进程”选项卡。查找名为 `sysedge.exe` 的进程。如果此进程显示在列表中，则表示 CA SystemEDGE 正在运行。
3. 选择“开始”、“控制面板”、“管理工具”、“服务”，并在“服务”对话框中查找 CA SystemEDGE。
4. 运行 `walktree` 或 `snmpget`。
5. 检查代理错误消息的 `sysedge.log` 文件。

验证代理在系统初始化时启动

要验证代理是否在系统初始化时启动，请根据平台执行以下操作之一：

- 检查 UNIX 启动脚本。
- 验证 Windows SystemEDGE 服务是否配置为自动启动。

验证代理响应查询

使用 `sysvariable` 实用工具查询系统，并证明代理正在响应查询。如果代理配置为接受 `SNMPv1` 或 `SNMPv2c` 通信，则必须指定代理的端口号（除非您使用的是默认端口 `161`）以及系统的团体字符串。否则，必须指定所有必要的 `SNMPv3` 通信（具体取决于安全级别—`Authpasswd`、`Authprotocol`、`Privpasswd`、`Privprotocol`）。

要显示 UNIX 系统的操作系统版本号，其中代理位于端口 `1691` 上且团体字符串为 `public`，则为 `SNMPv1` 输入以下内容：

```
#./sysvariable -h 127.0.0.1 -p 1691 -c public -v 1 -o systype
```

要显示 UNIX 系统的操作系统版本号，其中代理位于端口 `1691` 上且团体字符串为 `public`，则为 `SNMPv2c` 输入以下内容：

```
#./sysvariable -h 127.0.0.1 -p 1691 -c public -v 2c -o systype
```

要显示 UNIX 系统的操作系统版本号，其中代理位于端口 `1691` 上且 `SecLevel` 配置为 `AuthPriv(3)`（`USM` 用户名为 `secuser`，身份验证密码为 `authpasswd`，身份验证协议为 `MD5`，隐私密码为 `privpasswd`，隐私协议为 `DES`），则为 `SNMPv3` 输入以下内容：

```
#./sysvariable -h 127.0.0.1 -p 1691 -v 3 -u secuser -s 3 -a authpasswd -A MD5 -x privpasswd -X DES -o systype
```

对于代理位于端口 `1691` 上且团体字符串为 `public` 的 Windows 系统，为 `SNMPv1` 输入以下内容：

```
C:\sysedge\bin>sysvariable.exe -h 127.0.0.1 -p 1691 -c public -v 1 -o systype
```

对于代理位于端口 `1691` 上且团体字符串为 `public` 的 Windows 系统，为 `SNMPv2c` 输入以下内容：

```
C:\sysedge\bin>sysvariable.exe -h 127.0.0.1 -p 1691 -c public -v 2c -o systype
```

对于 Windows 系统，其中代理位于端口 `1691` 上且 `SecLevel` 配置为 `AuthPriv(3)`（`USM` 用户名为 `secuser`，身份验证密码为 `authpasswd`，身份验证协议为 `MD5`，隐私密码为 `privpasswd`，隐私协议为 `DES`），则为 `SNMPv3` 输入以下内容：

```
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
```


验证代理配置

要验证代理配置是否正确，请检查访问控制列表。对于 SNMPv2 和 SNMPv2c，需检查 sysedge.cf 中的团体。对于 SNMPv3，需检查 sysedgeV3.cf 中的 USM 安全配置。

为各个平台执行如下验证：

- 在 UNIX 系统上，团体字符串的 SYSEGE_DATADIR/sysedge.cf 和 USM 安全设置的 SYSEGE_DATADIR/sysedgeV3.cf。
- 在 Windows 系统上，通过 CA SystemEDGE 控制面板，验证 SNMP v1 和 SNMPv2c 团体。验证 SYSEGE_DATADIR\sysedgeV3.cf 中的 USM 安全信息。

注意：sysedgeV3.cf 可能被加密。

有关详细信息，请参阅“命令行实用工具”一章中的“se_enc 实用工具—加密 SNMPv3 配置文件”。

验证管理系统配置

如果 SystemEDGE 在备用 UDP 端口（例如，在 UDP/1691 上）上运行，要验证管理系统软件配置是否正确，则检查其查询的系统和端口号是否正确。

未接收 SNMP 陷阱消息的管理系统

如果管理系统没有从 SystemEDGE 接收 SNMP 陷阱，则必须验证代理是否发送陷阱 PDU 且发送的地址是否正确。如果已验证这些条件，最有可能的问题是管理系统的配置问题。执行以下验证找出问题根源：

- [验证代理是否正在运行](#) (p. 415)
- [验证代理是否配置为发送陷阱消息](#) (p. 418)
- [验证代理是否正在发送陷阱](#) (p. 418)
- [验证管理系统陷阱配置](#) (p. 418)

验证代理陷阱配置

要验证代理是否配置为将陷阱消息发送至相应地址，请执行以下操作之一：

- 对于 **SNMPv1**，需验证 **sysedge.cf** 文件中的陷阱团体。
- 对于 **SNMPv3**，需验证 **sysedgeV3.cf** 文件中的 **USM** 安全配置。
- 在 **UNIX** 系统上，需验证 **sysedge.cf** 文件中的陷阱团体。例如，确保为每个陷阱团体语句仅指定一个 IP 地址。有关陷阱团体配置的详细信息，请参阅“代理配置”一章。
- 在 **Windows** 系统上，通过“控制面板”应用程序验证陷阱配置（仅 **SNMPv1**）。有关详细信息，请参阅“代理配置”一章中的“配置陷阱团体”。

验证代理是否正在发送陷阱

要验证代理是否正在发送陷阱，请执行以下操作：

- 在 **UNIX** 系统上，检查 **CASYSEDGE_DATADIR/sysedge.log**。
- 在 **Windows** 系统上，检查 **ASYSEDGE_DATADIR\sysedge.log**。

验证管理系统陷阱配置

要验证管理系统的配置是否正确，请确保 **empire.asn1** 文件已导入管理站（以便管理系统了解 **SystemEDGE** 代理生成的 **SNMP** 陷阱 **PDU** 的格式）。有关管理系统软件的故障排除的详细信息，请联系您的管理系统供应商。

绑定失败：地址已被使用

如果绑定因为地址已被使用而失败，则代理首次启动时将显示类似于以下内容的消息：

```
sysedge: 绑定调用失败：地址已被使用
sysedge: 其他代理可能正在端口 X 上运行
```

该消息通常在其他 **SNMP** 代理已运行且绑定至端口 **UDP/161**（默认 **SNMP** 代理端口）时出现。

有关将 **SystemEDGE** 代理与其他 **SNMP** 代理一起使用的详细信息，请参阅附录“与其他 **SNMP** 代理共存”。有关在备用 **UDP** 端口上使用代理的详细信息，请参阅“启动代理”一章。

CA SystemEDGE: xtrapmon.exe 实用工具被 edgetrapmon.exe 取代

edgetrapmon 实用工具取代 Windows、Linux 和 UNIX 上的 xtrapmon。
xtrapmon 实用工具不再可用。另请参阅 《SystemEDGE 版本说明》。

附录 A: FIPS 140-2 加密

此附录介绍如何为 SystemEDGE 安装和启用 FIPS 模式。

此部分包含以下主题：

[FIPS 140-2 模式](#) (p. 421)

[FIPS 库安装](#) (p. 421)

[平台支持](#) (p. 422)

[支持的加密协议](#) (p. 422)

[支持的身份验证协议](#) (p. 422)

[配置 FIPS 140-2 模式](#) (p. 423)

[FIPS 模式注意事项](#) (p. 423)

[密钥保护](#) (p. 424)

FIPS 140-2 模式

美国联邦法规要求，所有销售给美国联邦政府的新软件产品都使用联邦信息处理标准 (FIPS 140-2) 验证的加密算法（如果该产品包含加密）。

可以使用某一版本的加密库以 FIPS 模式操作 SystemEDGE，该库已根据 FIPS 140-2 标准的规则进行认证。

FIPS 库安装

要使用 FIPS 操作模式，除 SystemEDGE 核心软件包之外，还必须安装 SystemEDGE 高级加密软件包。SystemEDGE 高级加密软件包将 FIPS 库与其他库一起安装，这些其他库可以提供比核心代理软件包更高级的加密。

SystemEDGE 高级加密软件包安装以下 FIPS 认证的文件：

Windows

- cryptocme2.dll [FIPS 库]
- cryptocme2.sig [FIPS 库特征码文件]

UNIX

- libcryptocme2.so [用于 UNIX 的 FIPS 库，不用于 HP-UX PA-RISC]
- libcryptocme2.sl [用于 HP-UX PA-RISC 的 FIPS 库]
- libcryptocme2.sig [FIPS 库特征码文件]

有关如何安装 SystemEDGE 高级加密的详细信息，请参阅附录“SystemEDGE 高级加密”。

平台支持

RSA B-safe Crypto FIPS 兼容的库当前仅可用于以下支持的代理平台：

- Windows 32-bit
- Windows 64 位非 Itanium
- Redhat 32 位和 64 位非 Itanium
- SuSE Linux 32 位和 64 位非 Itanium
- Solaris SPARC
- Solaris 32 位和 64 位非 Intel
- 注意：Solaris Intel 体系结构平台不支持 FIPS 140-2。
- HP-UX IA-64
- HP-UX 11.31 PA-Risc 64 位
- 注意：Hp-ux PA-Risc 的早期版本不支持 FIPS 140-2。
- AIX 32 位和 64 位

支持的加密协议

FIPS 操作模式的 SystemEDGE 支持以下加密协议。不支持未在下面列出的任何内容。

- 三重数据加密标准 (3DES)
- 使用 128 位或 256 位密钥的高级加密标准 (AES)

支持的身份验证协议

仅 FIPS 操作模式的 SystemEDGE 仅支持以下身份验证协议：

- 安全哈希算法 1 (SHA)

配置 FIPS 140-2 模式

可以使用 `sysedge.cf` 文件中的 `sysedge_fips_mode` 参数配置代理处理其加密的方式。

`sysedge_fips_mode` 参数的选项如下：

0（零）

表示非 FIPS 模式。代理启用 CA eTrust 公钥基础架构库，如果该方法失败，将使用基于非认证的 FIPS 代码的基本内置加密和身份验证协议。

1

表示 FIPS 共存模式。代理尝试使用 FIPS 认证的加密和身份验证协议（使用 FIPS 认证的库）。但是，如果找不到 FIPS 认证的库，或如果代理无法加载这些库，代理的功能将返回使用非 FIPS 兼容的 CA eTrust 公钥基础架构库。如果 `sysedge_fips_mode` 未配置，则其为默认模式。

2

表示仅 FIPS 模式。仅支持 FIPS 认证的加密和身份验证代码和协议，禁用所有未认证的 FIPS 代码和协议。如果 FIPS 库失败，则不执行任何加密。

示例

将以下行添加到 `sysedge.cf` 文件中，以仅 FIPS 模式运行代理：

```
sysedge_fips_mode 2
```

FIPS 模式注意事项

使用 FIPS 模式时有以下注意事项：

- 在不支持 FIPS 模式的平台上以 FIPS 模式（`sysedge_fips_mode 1` 或 `sysedge_fips_mode 2`）启用代理时，其行为与支持 FIPS 模式的平台相同。

例如，如果启用仅 FIPS 模式（`sysedge_fips_mode 2`），且在不支持 FIPS 模式的平台上使用授权协议 MD5，则使用 MD5 进行的访问将失败（这与支持仅 FIPS 模式的平台上的行为相同）。

密钥保护

SystemEDGE 使用配置文件 `sysedgeV3.cf` 提供 SNMPv3 用户配置。可以加密该文件以保护密钥和密码。有关详细信息，请参阅“代理配置”一章。

附录 B: SystemEDGE 高级加密

本章介绍如何安装 SystemEDGE 高级加密软件包，以便向 SystemEDGE 提供其他加密功能和 FIPS 140-2 操作模式。

此部分包含以下主题：

[SystemEDGE 高级加密](#) (p. 425)

[支持的平台](#) (p. 425)

[支持的加密协议](#) (p. 425)

[支持的身份验证协议](#) (p. 426)

[FIPS 兼容性](#) (p. 426)

[安装先决条件](#) (p. 426)

[安装 SystemEDGE 高级加密](#) (p. 426)

[安装的文件](#) (p. 429)

SystemEDGE 高级加密

SystemEDGE 高级加密为 SystemEDGE 核心软件包中的软件提供其他加密。此软件包包含 CA eTrust 公钥基础架构 (ETPKI) 库，这些库提供其他加密功能且包括 FIPS 操作模式的 RSA BSAFE Crypto 库。

支持的平台

除 Solaris 8 和 9 32 位操作系统以外，SystemEDGE 支持的所有平台上都支持安装 SystemEDGE 高级加密。

支持的加密协议

安装了 SystemEDGE 高级加密软件包的 SystemEDGE 支持以下加密协议：

- 数据加密标准 (DES)
- 三重数据加密标准 (3DES)
- 使用 128 位和 256 位密钥的高级加密标准 (AES)

支持的身份验证协议

安装了 SystemEDGE 高级加密软件包的 SystemEDGE 支持以下加密协议：

- 消息摘要算法 5 (MD5)
- 安全哈希算法 1 (SHA)

FIPS 兼容性

SystemEDGE 高级加密软件包自动安装 RSA B-safe Crypto FIPS 兼容的库。有关 FIPS 模式支持的平台以及特定注意事项的详细信息，请参阅附录“FIPS 140-2 加密”。

安装先决条件

安装 SystemEDGE 高级加密的先决条件是安装有 SystemEDGE。

安装 SystemEDGE 高级加密

下节介绍如何在 Windows 和 UNIX 上安装和卸载 SystemEDGE 高级加密。

在 Windows 上安装 SystemEDGE 高级加密

SystemEDGE 高级加密软件包可用于所有支持的 Windows 硬件体系结构。运行安装时，它会针对您的系统体系结构自动安装适当的版本。

在 Windows 上安装 SystemEDGE 高级加密

1. 以管理员身份登录 Windows 系统，打开 DVD1\Installers\Windows\Agent\SysMan\CA_SystemEDGE_AdvancedEncryption 文件夹，然后双击 ca-setup.exe。

此时将显示 SystemEDGE 高级加密安装向导。

2. 单击“下一步”。

此时会出现“许可协议”页面。

3. 选择“我接受此许可协议的条款”，然后单击“下一步”。

此时将显示“检查设置”页面。

4. 单击“安装”。

安装完成后将显示“安装完成”页面。

要验证安装，请检查 SystemEDGE 安装目录中是否存在加密库。有关详细信息，请参阅“安装的文件”。

在 Windows 上卸载 SystemEDGE 高级加密

验证您是否未安装依赖于 SystemEDGE 高级加密的软件包。请执行下列操作：

在 Windows 上卸载 SystemEDGE 高级加密

1. 打开“添加或删除程序”对话框，选择“SystemEDGE 高级加密”，然后单击“删除”。

这时将会出现确认对话框。

2. 单击“是”。

软件包已卸载。

在 UNIX 上安装 SystemEDGE 高级加密

SystemEDGE 高级加密安装软件包可用于 SystemEDGE 支持的所有 UNIX 平台和体系结构。安装适合系统和体系结构的适当软件包。

在 UNIX 上安装 SystemEDGE 高级加密

1. 以 root 用户身份登录系统，在本地复制
DVD2\Installers\<platform>\Agent\SysMan\CA_SystemEDGE_Advanced Encryption 文件夹，选择与操作系统相对应的 *platform* 文件夹。

2. 从以下目录运行安装：

```
./ca-setup.sh CA_SETUP_JRE_DIR=""
```

CA_SETUP_JRE_DIR

在系统上指定 Java 目录，这样安装程序可以显示安装向导对话框。如果 Java 不可用，对话框将以文本模式显示在控制台中。

安装完成后将显示“安装完成”页面。

要验证安装，请检查 SystemEDGE 安装目录中是否存在加密库。有关详细信息，请参阅“安装的文件”。

在 UNIX 上卸载 SystemEDGE 高级加密

验证您是否未安装依赖于 SystemEDGE 高级加密的软件包。请执行下列操作：

在 UNIX 上卸载高级加密

1. 打开终端控制台。
2. 导航到系统上的 CA_SystemEDGE_AdvancedEncryption 文件夹并运行以下命令：

```
./ca-setup.sh -x
```

3. 代理卸载。

安装的文件

安装 SystemEDGE 高级加密时，在 Windows 系统上安装以下文件：

- 以下安装的文件提供身份验证和加密（非 FIPS 模式）：
 - libcapki.dll
 - libcapki_thread.dll
 - libcaopenssl_crypto.dll
 - libcaopenssl_ssl.dll
 - libcapki_ipthread.dll
- 对于 FIPS 操作模式，安装以下文件：
 - cryptocme2.dll
 - cryptocme2.sig
 - ccme_base.dll
 - ccmd_ecc.dll
 - ccme_eccaccel.dll

有关详细信息，请参阅附录“FIPS 140-2 加密”。

安装 SytemEDGE 高级加密时，在 UNIX 系统上安装以下文件：

- 以下安装的文件提供身份验证和加密（非 FIPS 模式）：
 - libcapki
 - libcapki_thread_posix
 - libcaopenssl_crypto
 - libcaopenssl_ssl
 - libetpki_openssl_ssl

注意：文件扩展名为 .so、.sl 或 .dylib，具体取决于操作系统。

- 对于 FIPS 操作模式，安装以下文件：

- libcryptocme2
- libcryptocme2.sig
- libccme_base
- libccme_ecc
- libccme_eccaccel

有关详细信息，请参阅附录“FIPS 140-2 加密”。

注意：文件扩展名为 .so 或 .sl（除了 .sig 文件），具体取决于操作系统。

附录 C： 与其他 SNMP 代理共存

本章介绍了如何将 SystemEDGE 代理与其他 SNMP 代理结合使用。

此部分包含以下主题：

[支持多个 SNMP 代理](#) (p. 431)

[支持多个 SystemEDGE 实例](#) (p. 432)

[与 Microsoft Windows SNMP 代理共存](#) (p. 432)

[与 HP SNMP 代理共存](#) (p. 432)

[与 AIX SNMP 代理共存](#) (p. 433)

支持多个 SNMP 代理

默认情况下，SNMP 代理会使用 UDP 端口 161，但在任何一个时刻只有一个进程可以绑定到给定的端口。因此，一次只有一个 SNMP 代理可以使用 UDP 端口 161。如果要同时运行多个 SNMP 代理，必须开发用于共存的策略。

为使多个 SNMP 代理能够共享 UDP 端口 161，存在以下几种可能的解决方案：

- 在多个 SNMP 代理之间使用多路复用 UDP 端口 161（通常是主/从关系）。
- 如果每个代理分别绑定到单独的 UDP 端口，且管理软件配置为分别与这些代理进行通信，则同时运行这些代理。

注意：保留 UDP 端口 1691 的目的是供 SystemEDGE 代理使用，该端口是从 CA Virtual Assurance 执行代理部署的默认端口。可以将代理配置为使用该端口（而非端口 UDP/161）。有关在备用端口上启动代理的详细信息，请参阅“启动代理”一章。

- 为每个代理编写代码，使其符合每个现有的专用主代理/子代理 API。就编码、测试和授权大量专用 API 而言，该解决方案的成本极高。

代理多路复用的最常见概念涉及绑定到 UDP 端口 161 并与实施不同 MIB 的单独子代理或从代理进行通信的主代理。主代理和子代理使用可以注册子代理并在主/子代理之间交换信息的已定义协议进行通信。存在几种协议可用于控制主代理与子代理之间的此类通信协议。

通过 SystemEDGE 安装，您还可以将本地操作系统替换为 SystemEDGE 代理。如果确实替换了代理，则可以选择自动继承本地代理的设置。有关在安装期间替换本地 SNMP 代理的详细信息，请参阅“安装和部署”一章。

支持多个 SystemEDGE 实例

尽管它不是正式支持的功能，但您可以在单个服务器的命令 `shell` 中运行 SystemEDGE 代理的多个实例（例如测试实例中的测试配置更改）。通过用于启动代理的命令行选项，您可以指定侦听端口和配置文件位置。通过在启动每个实例时在命令行中输入不同的端口，可以同时启动多个代理实例。还可以编辑和配置每个代理以使用不同配置文件（如有必要）。

重要信息！ 代理安装或 CA Virtual Assurance 部署和管理并不支持多个 SystemEDGE 实例。仅支持您在安装期间指定的默认端口上运行的实例。如果需要卸载代理，则仅删除默认实例。必须手动配置、实施和维护同一服务器上的任何其他实例。

有关在一个服务器上运行多个代理实例的详细信息，请参阅“启动代理”一章。

与 Microsoft Windows SNMP 代理共存

SystemEDGE 代理不再是 Microsoft SNMP 代理的子代理。它可以与 Microsoft Windows 可扩展代理共存。

采取以下任一共存方法：

- 在系统上同时运行 Microsoft 代理和 SystemEDGE 代理，其中，在端口 1691 上运行 SystemEDGE 代理，在端口 161 上运行 Microsoft 代理。
- 关闭或禁用 Microsoft 主代理，并在端口 161 上运行 SystemEDGE 代理。在安装期间替换本地代理会实施该方法。

与 HP SNMP 代理共存

HP-UX 系统附带有可扩展代理和子代理，它们为 MIB-II 和 HP 专用企业 MIB 提供支持。主/子代理接口基于 SystemEDGE 代理不支持的专用 API。

采取以下任一共存方法：

- 在系统上同时运行 HP 代理和 SystemEDGE 代理，其中，在端口 UDP/1691 上运行 SystemEDGE 代理。
- 关闭或禁用 HP 主代理或子代理，并在端口 UDP/161 上运行 SystemEDGE 代理。在安装期间替换本地代理会实施该方法。

与 AIX SNMP 代理共存

AIX 4.1 和更高版本的系统附带有也支持 SMUX 的 MIB-II SNMP 代理。可以同时运行 AIX 代理和 SystemEDGE 代理，或者用后者替换前者。

采取以下任一共存方法：

- 同时运行 SystemEDGE 代理和 AIX 代理。要执行此操作，请确保在 AIX TCP/IP 启动脚本 `/etc/rc.tcpip` 中使用 `-p 1691` 选项调用了 SystemEDGE 代理。
- 运行 SystemEDGE 代理（替换 AIX 代理）。要执行此操作，请在以下行的开头添加磅符号（#）以取消 AIX 代理在 `/etc/rc.tcpip` 中的调用。

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

在安装期间替换本地代理也会实施该方法。

附录 D： 主机资源 MIB

本章介绍了通过 IETF 主机资源 MIB (RFC 1514) 提供的管理信息。

此部分包含以下主题：

[主机资源 MIB 概述](#) (p. 436)

[主机资源系统组](#) (p. 437)

[主机资源存储组](#) (p. 438)

[主机资源设备组](#) (p. 438)

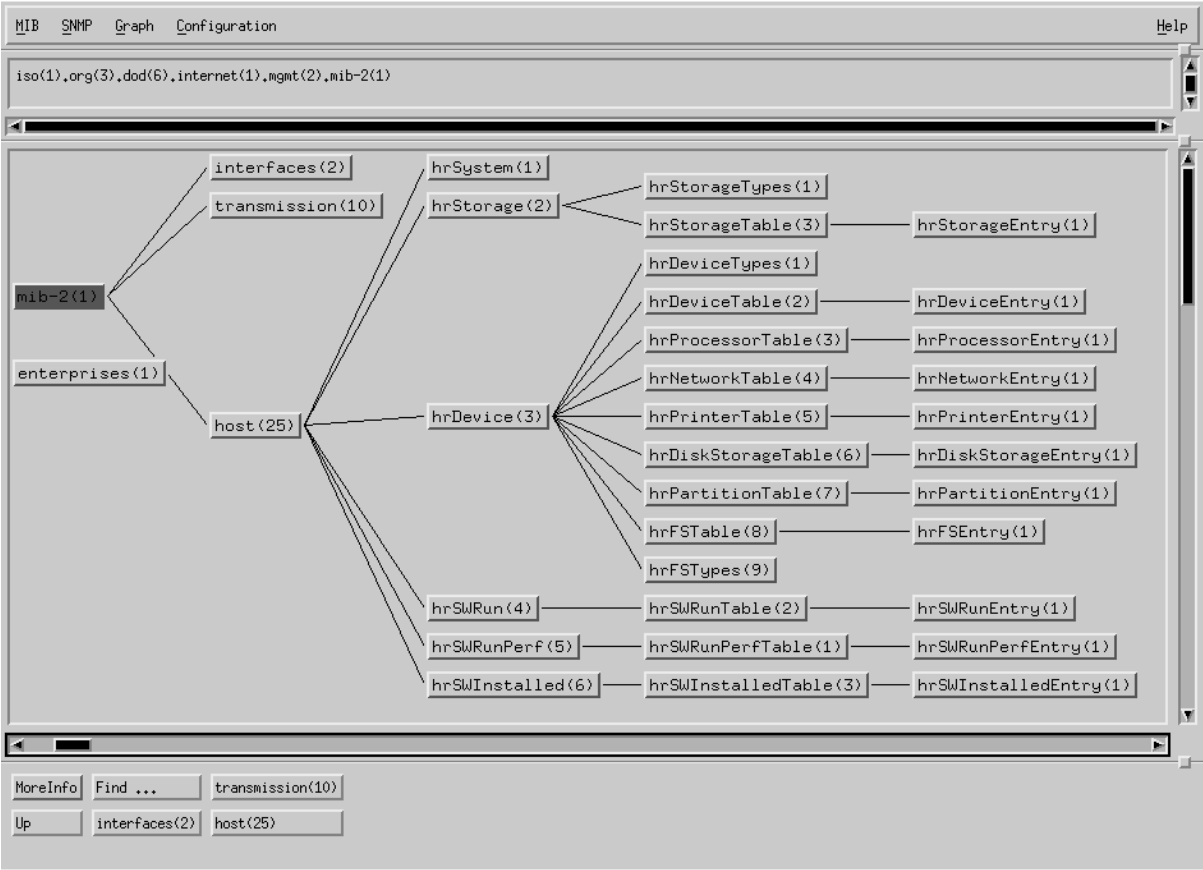
[主机资源运行软件组](#) (p. 441)

[主机资源安装软件组](#) (p. 442)

[不支持的主机资源 MIB 对象](#) (p. 442)

主机资源 MIB 概述

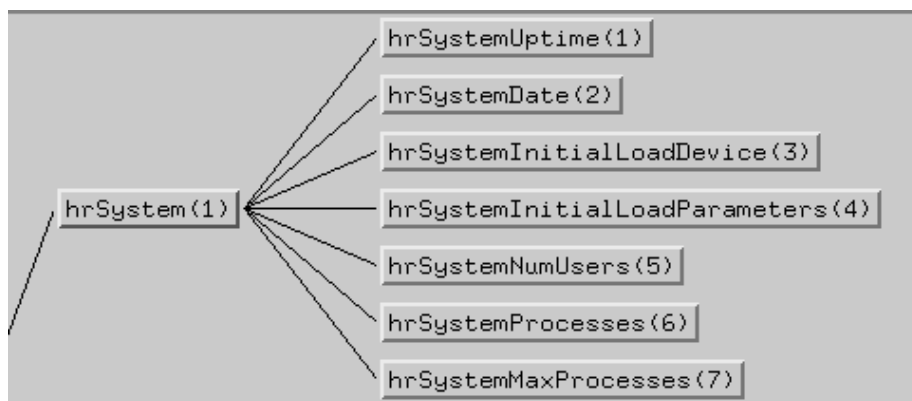
主机资源 MIB 定义了一组可以管理主机计算机的对象，这些对象与操作系统、网络服务或任何软件应用程序无关。主机资源 MIB 中定义的对象在许多计算机系统体系结构之间都是通用的。下图显示了主机资源 MIB 的总体结构：



注意：本章中的示例没有介绍整个主机资源 MIB。有关整个 MIB 的说明，请参阅 `hostmib.asn1` 文件（包括在代理安装的 `mib` 子目录中）。

主机资源系统组

主机资源系统 (hrSystem) 组提供与整个主机系统有关的信息。下图显示了主机资源 hrSystem 组的结构：



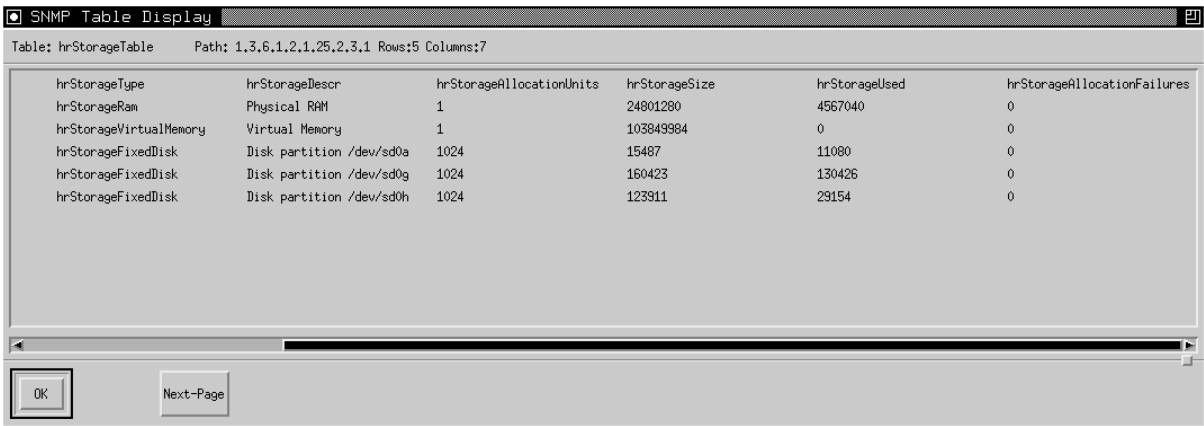
通过主机资源系统组可以提供以下信息：

- 系统正常运行时间和日期
- 操作系统加载设备、路径名和参数
- 主机为其存储状态信息的用户会话的数量
- 当前加载或正在运行的进程数量
- 可以运行的进程的最大数量

主机资源存储组

主机资源存储组 (**hrStorageTable**) 列出了在主机系统上分配的存储逻辑区域。这些存储区域包括应用程序可能会识别的文件系统和磁盘分区，而非物理存储（如磁带和软盘驱动器）。

对于内存不足和缓冲区溢出等类型的失败，使用下表进行诊断十分有用。而对于跟踪内存、磁盘或缓冲区利用率而言，它还是十分有用的性能监控工具。下图显示了主机资源存储组的输出示例：



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 data is as follows:

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

主机资源设备组

主机资源设备 (**hrDevice**) 组包含多个表，可提供有关主机系统上的设备的信息。该组中的主要表为 **hrDeviceTable**。它列出了主机包含的所有设备。**hrDevice** 组还包括大量特定于设备的表，这些表提供有关特定设备类型的更详细信息。

例如，该组包括有关以下各项的特定于设备的表：

- 处理器
- 网络
- 打印机
- 磁盘存储
- 分区
- 文件系统

例如，如果 **hrDeviceTable** 显示主机包含打印机设备，则可以从 **hrPrinterTable** 检索有关打印机的更详细信息。以下各节介绍了 **hrDeviceTable** 和特定于设备的表。

设备表

hrDeviceTable 列出了主机系统上的设备。该表列出了每个设备的以下信息：

- 设备类型
- 设备说明
- 状态（例如，正在运行或关闭）
- 该设备上检测到的错误数

清单跟踪和资产管理

对于清单跟踪和资产管理，hrDeviceTable 尤其有用。您可以使用中央管理系统从网络的每个系统中检索 hrDeviceTable，而非通过手动打开并检查网络上的每个工作站来确定以太网卡、串行设备、音频设备、磁盘存储设备等设备的数量和类型。

处理器表

hrProcessorTable 列出了有关系统处理器的设备特定的信息。如果系统仅包含单个处理器，则 hrProcessorTable 仅包括一行。hrProcessorTable 的列提供与处理器有关的固件的产品 ID（如果已分配此类 ID 并且处理器供应商已将其设置为可用）以及处理器负载。负载是指处理器处于非空闲状态的时间百分比（过去一分钟）的平均负载。

处理器负载跟踪

通过监控（用图表表示）平均处理器负载，可以直观地跟踪施加在处理器上的负载。例如，如果图表显示一直保持在高水平（表示处理器负载过重），则可以确定主机系统因执行的用户和进程过多而正在超负荷工作，需要您限制用户的数量或将一部分进程转移到其他系统上运行。

处理器负载图表中的临时峰值可能表明有占用大量处理器时间的进程正在运行。要避免因处理器负载过重而导致的问题，可以重新排定该进程，使其在处理器负载较轻的时候运行。

磁盘存储表

hrDiskStorageTable 提供有关主机的磁盘存储设备的信息。该表列出了每个磁盘存储设备的以下信息：

- 实例
- 磁盘存储设备允许读写访问还是只读访问
- 存储介质类型（例如硬盘或软盘）
- 磁盘是否可移动
- 存储容量 (KB)

分区表

hrPartitionTable 显示已为磁盘存储设备配置的分区。该表列出了每个分区的以下信息：

- 实例
- 分区索引
- 分区标签
- 分区 ID
- 分区大小 (KB)
- 挂接在该分区上的文件系统的索引

文件系统表

hrFSTable 提供有关主机的文件系统（本地和远程）的信息。该表列出了每个文件系统的以下信息：

- 本地挂载点
- 远程挂载点（如果文件系统从远程服务器挂载）
- 文件系统的类型（例如 BerkeleyFFS）
- 文件系统允许读写访问还是只读访问

下图显示了 hrFSTable 的输出示例：

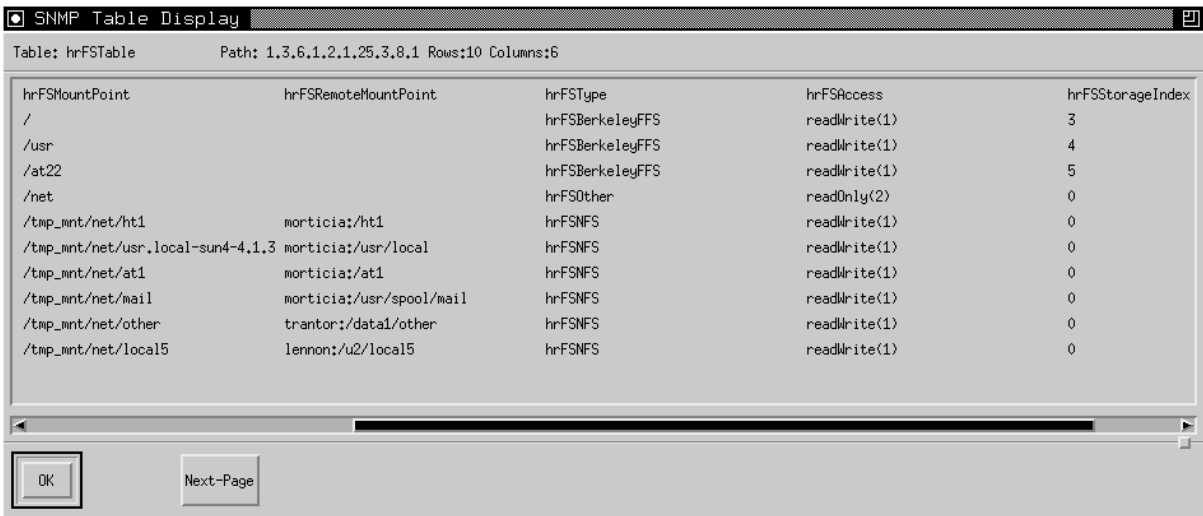


Table: hrFSTable Path: 1.3.6.1.2.1.1.25.3.8.1 Rows:10 Columns:6

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

主机资源运行软件组

hrSWRun 表列出了当前在主机系统上运行的软件。该表列出了每个运行软件进程的以下信息：

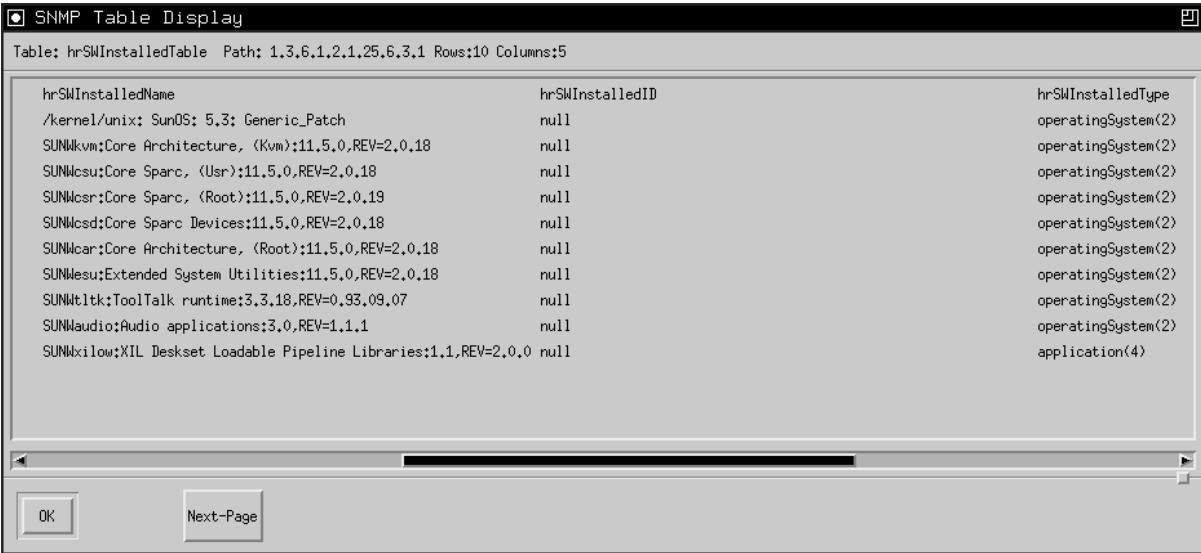
- 唯一标识号
- 名称
- Product ID
- 软件所在的目录路径
- 启动软件时所用的运行时参数
- 软件的类型（例如操作系统或应用程序）
- 软件的状态（例如正在运行、可运行、不可运行或无效）

主机资源安装软件组

hrSWInstalled 表列出了当前安装在主机系统上的软件。该表列出了每个软件包的以下信息：

- 唯一标识符
- 软件包名称
- 描述
- 安装日期

该表显示了操作系统修补程序以及安装的软件版本，非常适合于检查系统之间的软件版本是否一致。下图显示了 hrSWInstalledTable 示例的一部分：



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, (Usr):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)		
SUNWtltk: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)		

注意：并非所有的系统都可以支持该表。要确定您的操作系统是否支持该表，请参阅 《SystemEDGE 版本说明》。

不支持的主机资源 MIB 对象

SystemEDGE 代理在所有支持的操作系统上都支持主机资源 MIB (RFC 1514)，但是该代理或基础操作系统不支持该 MIB 模块中的一些 MIB 对象。因此，这些对象无法由安装在受影响操作系统上的代理版本实施。

有关所有操作系统上不受支持的主机资源 MIB 对象的列表，请参阅 《SystemEDGE 版本说明》。

附录 E：私有企业陷阱

除支持标准 MIB-II 陷阱之外，SystemEDGE 代理还支持大量定义的私有企业陷阱类型，用于代理的自主监控功能。本附录介绍 SystemEDGE 代理可以发送的陷阱 PDU 的格式。

此部分包含以下主题：

[陷阱 PDU 的格式](#) (p. 443)

[SNMPv1 陷阱格式](#) (p. 452)

陷阱 PDU 的格式

Systems Management Empire MIB 文件 `empire.asn1` 定义用于 SystemEDGE 代理的企业特定陷阱。本节介绍每种类型陷阱的 PDU 格式，包括变量绑定字段中的信息。

变量绑定是提供与陷阱关联的其他重要信息的陷阱 PDU 中的可选字段。

注意：本章中的文本摘自 Systems Management Empire MIB 定义，该定义位于代理安装的 `mib` 子目录中。

SystemEDGE 代理发送的所有陷阱包含以下企业系统对象标识符 (sysObjectID)：`empire(546).1.1`。

monitor 陷阱

下面显示了 SystemEDGE 代理发送的 `monitor` 陷阱的格式，表示自主监控事件已发生。当到达受监控属性的阈值且自主监视器的重要级别为无(1)时，代理将发送该陷阱，否则它将发送 `aggregateState` 陷阱。

```
monitorEvent OBJECT IDENTIFIER ::= { traps 1 }
```

```
monitorTrap TRAP-TYPE
```

```
ENTERPRISE sysmgmt
```

```
VARIABLES { monDescr, monOID, monCurrVal, monValue, monRowStatus, monOperator,
monIndex, monFlags, monObjClass, monObjInstance, monObjAttribute, monCurrState }
```

```
DESCRIPTION
```

```
“违反了自主监视器（见 monitorTable）的阈值条件。”
```

```
::= 1
```

monitorEntryNotReady 陷阱

下面显示了 SystemEDGE 代理发送的 monitorEntryNotReady 陷阱的格式，表示将自主监视器表条目的 monRowStatus 字段设置为 notReady(3)。如果监视器的重要级别为无(1)，则代理仅发送该陷阱。

```
monitorEntryNotReadyEvent OBJECT IDENTIFIER ::= { traps 3 }
```

```
monitorEntryNotReadyTrap TRAP-TYPE
```

```
ENTERPRISE sysmgmt
```

```
VARIABLES { monDescr, monOID, monCurrVal, monValue, monRowStatus, monOperator,  
monIndex, monFlags, monObjClass, monObjInstance, monObjAttribute, monCurrState }
```

DESCRIPTION

“自主监视器（见 monitorTable）的行状态已变为 notReady(3)，通常是因为受监控的 OID 不存在。”

```
::= 3
```

logMonMatch 陷阱

下面显示了 SystemEDGE 代理发送的 logMonMatch 陷阱的格式，表示日志文件监控子系统已在此代理当前监控的日志文件中检测到一个匹配项。监视器重要级别包含在变量绑定中。

```
logMonMatchEvent OBJECT IDENTIFIER ::= { traps 4 }
```

```
logMonMatchTrap TRAP-TYPE
```

```
ENTERPRISE sysmgmt
```

```
VARIABLES { logMonitorLogFile, logMonitorRegularExpression,  
logMonitorLastTrap, logMonitorLastMatch,  
logMonitorDescr, logMonitorIndex, logMonitorFlags,  
logMonitorLogFileName, logMonitorSeverity }
```

DESCRIPTION

“日志文件监视器（见 logMonitorTable）的正则表达式模式匹配。”

```
::= 4
```

logMonNotReady 陷阱

下面显示了 SystemEDGE 代理发送的 logMonNotReady 陷阱的格式，表示日志监视器表中的条目状态已更改为 notReady。

```
logMonNotReadyEvent OBJECT IDENTIFIER ::= { traps 5 }
```

```
logMonNotReadyTrap TRAP-TYPE
```

```
ENTERPRISE sysmgmt
```

```
VARIABLES { logMonitorLogFile, logMonitorRegularExpression,
logMonitorLastTrap, logMonitorLastMatch,
logMonitorDescr, logMonitorIndex, logMonitorFlags,
logMonitorLogFileName, logMonitorSeverity }
```

DESCRIPTION

“日志文件监视器（见 logMonitorTable）的行状态已更改为 notReady(3)，通常是因为日志文件不存在。”

```
::= 5
```

ntEventMonMatch 陷阱

下面显示了 SystemEDGE 代理发送的 ntEventMonMatch 陷阱的格式，表示事件日志监控子系统已在此代理当前监控的 Windows 事件日志文件中检测到一个匹配项。监视器重要级别包含在变量绑定中。

```
ntEventMonMatchEvent OBJECT IDENTIFIER ::= { traps 7 }
```

```
ntEventMonMatchTrap TRAP-TYPE
```

```
ENTERPRISE sysmgmt
```

```
VARIABLES { ntEventMonLog, ntEventMonTypeLastMatch,
ntEventMonTime, ntEventMonSrcLastMatch,
ntEventMonDescLastMatch, ntEventMonDescr,
ntEventMonIndex, ntEventMonFlags, ntEventMonSeverity }
```

DESCRIPTION

“NT 事件日志监视器（见 ntEventMonTable）的所有筛选（日志、类型、源、说明）均匹配。”

```
::= 7
```

ntEventMonNotReady 陷阱

下面显示了 SystemEDGE 代理发送的 ntEventMonNotReady 陷阱的格式，表示 Windows 事件日志监控条目的状态已变为 notReady(3)。

```
ntEventMonNotReadyEvent OBJECT IDENTIFIER ::= { traps 8 }
```

```
ntEventMonNotReadyTrap TRAP-TYPE
```

```
ENTERPRISE sysmgmt
```

```
VARIABLES { ntEventMonLog, ntEventMonTypeFilter,  
ntEventMonSrcFilter, ntEventMonDescFilter,  
ntEventMonDescr, ntEventMonIndex, ntEventMonFlags, ntEventMonSeverity }
```

DESCRIPTION

“Windows 事件日志监视器（见 ntEventMonTable）的行状态已变为 notReady(3)，通常是因为事件日志不存在。”

```
::= 8
```

monitorClear 陷阱

下面显示了 SystemEDGE 代理发送的 monitorClear 陷阱的格式，表示先前存在（现在已不存在）的阈值违反条件。当自主监视器表条目将“发送清除陷阱”标志从 True 转换为 False 时，SystemEDGE 代理将发送该陷阱。如果监视器的重要级别为无(1)，则代理仅发送该陷阱，否则它将发送 aggregateState 陷阱。

```
monitorClearEvent OBJECT IDENTIFIER ::= { traps 9 }
```

```
monitorClearTrap TRAP-TYPE
```

```
ENTERPRISE sysmgmt
```

```
VARIABLES { monDescr, monOID, monCurrVal, monValue, monRowStatus, monOperator,  
monIndex, monFlags, monObjClass, monObjInstance, monObjAttribute, monCurrState }
```

DESCRIPTION

“不再违反自主监视器（见 monitorTable）的阈值条件。”

```
::= 9
```

processStop 陷阱

下面显示了 SystemEDGE 代理发送的 processStop 陷阱的格式，表示其监控的进程已停止运行或者处于无法运行的状态。如果进程监视器的重要级别为无(1)，则代理仅发送该陷阱；否则它将发送 aggregateState 陷阱。

```
processStopEvent OBJECT IDENTIFIER ::= { traps 10 }
```

```
processStopTrap TRAP-TYPE
```

```
ENTERPRISE sysmgmt
```

```
VARIABLES { pmonIndex, pmonDescr, pmonAttribute, pmonCurrVal, pmonOperator,
pmonValue, pmonFlags, pmonRegExpr, pmonCurrentPID, pmonObjClass, pmonObjInstance,
pmonObjAttribute, pmonCurrState, pmonProcName}
```

```
DESCRIPTION
```

```
“使用 procAlive 的进程监视器（见 processMonTable）所监控的进程（或 Windows 服务）未运行。”
```

```
::= 10
```

processStart 陷阱

下面显示了 SystemEDGE 代理发送的 processStart 陷阱的格式，表示停止的进程已重新启动（并且由 SystemEDGE 代理再次获取）。如果进程监视器的重要级别为无(1)，则代理仅发送该陷阱；否则它将发送 aggregateState 陷阱。

```
processStartEvent OBJECT IDENTIFIER ::= { traps 11 }
```

```
processStartTrap TRAP-TYPE
```

```
ENTERPRISE sysmgmt
```

```
VARIABLES { pmonIndex, pmonDescr, pmonAttribute, pmonCurrVal, pmonOperator,
pmonValue, pmonFlags, pmonRegExpr, pmonCurrentPID, pmonObjClass, pmonObjInstance,
pmonObjAttribute, pmonCurrState, pmonProcName}
```

```
DESCRIPTION
```

```
“使用 procAlive 的进程监视器（见 processMonTable）所监控的进程（或 Windows 服务）已重新启动。”
```

```
::= 11
```

processThreshold 陷阱

下面显示了 SystemEDGE 代理发送的 processThreshold 陷阱的格式，表示进程的受监控属性违反了指定的阈值条件。如果进程监视器的重要级别为无(1)，则代理仅发送该陷阱；否则它将发送 aggregateState 陷阱。

```
processThresholdEvent OBJECT IDENTIFIER ::= { traps 12 }
```

```
processThresholdTrap TRAP-TYPE
```

```
ENTERPRISE sysmgmt
```

```
VARIABLES { pmonIndex, pmonDescr, pmonAttribute, pmonCurrVal, pmonOperator,  
pmonValue, pmonFlags, pmonRegExpr, pmonCurrentPID, pmonObjClass, pmonObjInstance,  
pmonObjAttribute, pmonCurrState, pmonProcName}
```

```
DESCRIPTION
```

```
“违反了进程监视器（见 processMonTable）的阈值条件。”
```

```
::= 12
```

processClear 陷阱

下面显示了 SystemEDGE 代理发送的 processClear 陷阱的格式，表示以前用于发送 processThreshold 陷阱的进程属性不再属于阈值级别。如果进程监视器的重要级别为无(1)，则代理仅发送该陷阱；否则它将发送 aggregateState 陷阱。

```
processClearEvent OBJECT IDENTIFIER ::= { traps 13 }
```

```
processClearTrap TRAP-TYPE
```

```
ENTERPRISE sysmgmt
```

```
VARIABLES { pmonIndex, pmonDescr, pmonAttribute, pmonCurrVal, pmonOperator,  
pmonValue, pmonFlags, pmonRegExpr, pmonCurrentPID, pmonObjClass, pmonObjInstance,  
pmonObjAttribute, pmonCurrState, pmonProcName}
```

```
DESCRIPTION
```

```
“不再违反进程监视器（见 processMonTable）的阈值条件。”
```

```
::= 13
```


processNotReady 陷阱

下面显示了 SystemEDGE 代理发送的 processNotReady 陷阱的格式，表示进程监视器表条目的行状态已设置为 notReady(3)。如果正在使用 procAlive 监控进程，则发送 processStop 陷阱。如果进程监视器的重要级别为无(1)，则代理仅发送该陷阱。

```
processNotReadyEvent OBJECT IDENTIFIER ::= { traps 14 }
```

```
processNotReady TRAP-TYPE
```

```
ENTERPRISE sysmgmt
```

```
VARIABLES { pmonIndex, pmonDescr, pmonAttribute, pmonCurrVal, pmonOperator,
pmonValue, pmonFlags, pmonRegExpr, pmonCurrentPID, pmonObjClass, pmonObjInstance,
pmonObjAttribute, pmonCurrState, pmonProcName}
```

```
DESCRIPTION
```

“未使用 procAlive 的进程监视器（见 processMonTable）的行状态已变为 notReady(3)，通常是因为进程不存在。”

```
::= 14
```

addrChange 陷阱

下面显示了 SystemEDGE 代理发送的 addrChange 陷阱的格式，表示基础 IP 地址已更改。

```
addrChangeEvent OBJECT IDENTIFIER ::= { traps 18 }
```

```
addrChangeTrap TRAP-TYPE
```

```
ENTERPRISE sysmgmt
```

```
VARIABLES { nodename, sysedgeAddressList }
```

```
DESCRIPTION
```

“代理的计算机 IP 地址已更改，通常是由于 DHCP 或其他管理方式。”

```
::= 18
```

procGroupChange 陷阱

下面显示了 SystemEDGE 代理发送的 procGroupChange 陷阱的格式，表示进程组已更改。监视器重要级别包含在变量绑定中。

```
procGroupChangeEvent OBJECT IDENTIFIER ::= { traps 19 }
```



```
procGroupChangeTrap TRAP-TYPE
```



```
ENTERPRISE sysmgmt
```



```
VARIABLES { pgmonIndex, pgmonDescr, pgmonFlags, pgmonNumProcs, pgmonProcRegExpr,
pgmonRowStatus, pgmonPIDList, pgmonStatusList, pgmonSeverity,
pgmonProcNameList}
```



```
DESCRIPTION
```

“进程组监视器（见 processGroupMonTable）的成员身份已更改，通常是因为进程的加入或离开。”


```
::= 19
```

aggregateState 陷阱

下面显示了 SystemEDGE 代理发送的 aggregateState 陷阱的格式，表示聚合的状态已更改。当阈值违反监视器更改聚合状态时，代理仅发送该陷阱。例如，如果违反了重要级别为严重的监视器，而聚合当前状态为“严重”，则代理不发送陷阱，因为聚合状态不受影响。

```
aggregateStateEvent OBJECT IDENTIFIER ::= {traps 20}
```



```
aggregateStateTrap TRAP-TYPE
```



```
ENTERPRISE sysmgmt
```



```
VARIABLES { aggIndex, aggObjClass, aggObjInstance, aggObjAttribute, aggCurrState,
aggPrevState, aggMonTable, aggMonIndex, aggMonCurrVal, aggMonOperator,
aggMonValue }
```



```
DESCRIPTION
```

“聚合（见 aggregateTable）的状态已更改。”


```
::= 20
```

aggregateAdd 陷阱

下面显示了 SystemEDGE 代理发送的 aggregateAdd 陷阱的格式，表示聚合已添加到 Aggregate 表。

```
aggregateAddEvent OBJECT IDENTIFIER ::= {traps 21}

aggregateAddTrap TRAP-TYPE

ENTERPRISE sysmgmt

VARIABLES { aggIndex, aggObjClass, aggObjInstance, aggObjAttribute }

DESCRIPTION
    “已添加新的聚合（见 aggregateTable）。”

::= 21
```

aggregateDelete 陷阱

下面显示了 SystemEDGE 代理发送的 aggregateDelete 陷阱的格式，表示聚合已从 Aggregate 表中删除。

```
aggregateDeleteEvent OBJECT IDENTIFIER ::= {traps 22}

aggregateDeleteTrap TRAP-TYPE

ENTERPRISE sysmgmt

VARIABLES { aggIndex, aggObjClass, aggObjInstance, aggObjAttribute }

DESCRIPTION
    “已删除一个聚合（见 aggregateTable）。”

::= 22
```

SNMPv1 陷阱格式

SystemEDGE 代理将陷阱 PDU 发送到 SNMPv1 陷阱端口 (UDP/162)。以下是 SNMPv1 陷阱 PDU 的组件：

Enterprise

指定发件人的系统对象 ID： empire(546).1.1。

代理地址

指定发送主机的 IP 地址。

常规陷阱类型

指定常规陷阱类型，可以是以下类型之一：

- coldStart(0)
- warmStart(1)
- linkDown(2)
- linkup(3)
- authenFailure(4)
- egpNeighborloss(5)
- enterpriseSpecific(6)

特定陷阱类型

指定企业特定的陷阱类型。

时间戳

在发送陷阱时，指定 sysUptime 的值。对于 sendtrap，值始终是 0。

变量绑定

指定 MIB 变量及其值的阵列。

索引

符号

\$CASYSEDGE 变量的传统支持 - 99

A

addrChange 陷阱 - 449
aggregateAdd 陷阱 - 451
aggregateDelete 陷阱 - 451
aggregateState 陷阱 - 450
AIM 与 CA Virtual Assurance 的集成 - 72
AIM 体系结构 - 71
AIX 逻辑分区组 - 175
Application Insight Module (AIM) - 31

C

CA eHealth - 35
CA NSM - 35
CA Server Automation - 33
CA Spectrum - 34
CA SystemEDGE - 17
CA SystemEDGE: xtrapmon.exe 实用工具被
edgetrapmon.exe 取代 - 419
CA Systems Management Empire MIB - 26
CA Technologies 产品引用 - 3
CA Virtual Assurance - 32
CA Virtual Assurance 安全选项 - 70
CPU 统计信息表 - 183

E

edgemon 示例 - 226
edgemon 示例 - 268, 313, 333, 359
edgemon 自动监测器示例 - 285
edgemon 实用工具--使用自动监测器 - 278
edgemon 实用工具--监控 Windows 事件 -
354
edgemon 实用工具--监控日志文件 - 329
edgemon 实用工具--监控进程组 - 308
emphistory 示例 - 373
emphistory 指令--将条目添加到历史记录控
制表中 - 370
extension 关键字--向扩展组中添加条目 -
377

F

File-Based 配置 - 117
FIPS 140-2 加密 - 421
FIPS 140-2 模式 - 421
FIPS 库安装 - 421
FIPS 兼容性 - 426
FIPS 模式注意事项 - 423

I

I/O 缓冲区缓存组 - 175
IP-MIB 表格 - 29

L

logMonMatch 陷阱 - 444
logMonNotReady 陷阱 - 445
LPAR 组 - 185

M

MIB 扩展 - 30
MIB-II - 28
monitor 指令--将条目添加到自主监视器表
- 205
monitor 陷阱 - 443
monitorClear 陷阱 - 446
monitorEntryNotReady 陷阱 - 444

N

NFS 统计信息组 - 181
NT 内存性能组 - 178
NT 页面文件性能组 - 178
NT 系统性能组 - 178
NT 系统组 - 176
NT 事件日志监控表 - 179
NT 事件监视器组 - 179
NT 事件监视器表 - 341
NT 事件监视器表列 - 341
NT 事件监视器表标志 - 346
NT 事件监视器表操作参数 - 348
NT 性能组 - 177
NT 服务表 - 177

NT 注册表和性能扩展组 - 179

NT 注册表组 - 177

NT 线程表 - 177

NT 缓存性能组 - 178

ntEventMonMatch 陷阱 - 445

ntEventMonNotReady 陷阱 - 446

ntregperf 关键字—添加 Windows 注册表和性能 MIB 对象 - 390

P

Perfcube 模块配置 - 141

Perl 兼容的正则表达式 (PCRE) 支持 - 145

processClear 陷阱 - 448

processNotReady 陷阱 - 449

processStart 陷阱 - 447

processStop 陷阱 - 447

processThreshold 陷阱 - 448

procGroupChange 陷阱 - 450

R

RPC 统计信息组 - 180

S

SNMP 版本和访问 - 38

SNMP 表格 - 40

SNMP 陷阱 - 39

SNMPv1 和 SNMPv2c 社区 - 38

SNMPv1 陷阱格式 - 452

SNMPv2 行状态 - 44

SNMPv3 用户和安全 - 39

SNMPv3 用户的地址筛选 - 153

SNMPv3 配置 - 150

Solaris Zone 进程组监控 - 314

Solaris Zone 进程监控 - 272

sysedge.cf 和 sysedge.mon 之间的交互 - 117

SystemEDGE AIM - 395

SystemEDGE Control Panel for Windows - 159

SystemEDGE 其他命令行实用工具 - 400

SystemEDGE 命令行实用工具 - 399

SystemEDGE 的陷阱代理地址 - 126

SystemEDGE 高级加密 - 425

Systems Management Empire MIB - 167

Systems Management Empire MIB ntRegPerf 组 - 386

Systems Management Empire MIB 扩展组 - 375

Systems Management Empire MIB 组 - 50

Systems Management Empire MIB 概述 - 168

U

UNIX 系统的服务启动脚本 - 162

W

watch logfile 指令—将条目添加到日志监视器表 - 326

watch ntevent 指令—将条目添加到 NT 事件监视器表中 - 351

watch ntservice 指令—将服务监控条目添加到进程监视器表 - 257

watch process 指令—将条目添加到进程监视器表 - 254

watch procgroup 指令--将条目添加到 - 303

watch procgroupex 指令--将条目添加到 - 305

Windows 事件监视器陷阱 - 57

Windows 事件监控 - 57, 339

Windows 事件监控示例 - 353

Windows 事件监控配置 - 350

Windows 事件监控概述 - 339

Windows 事件搜索条件 - 340

Windows 服务监控 - 230

Windows 注册表和性能 MIB 对象 - 385

Windows 注册表和性能功能 - 385

Windows 注册表和性能示例 - 391

Windows 注册表和性能变量的配置 - 389

三划

工作负荷管理程序 (WLM) 类的表 - 176

四划

不支持的主机资源 MIB 对象 - 442

不支持的性能数据类型 - 389

不受支持的 Systems Management Empire MIB 对象 - 186

三划

与 AIX SNMP 代理共存 - 433

与 CA Virtual Assurance 集成 - 67

与 HP SNMP 代理共存 - 432
与 Microsoft Windows SNMP 代理共存 - 432
与其他 SNMP 代理共存 - 431

四划

为 AIX 系统配置磁盘性能统计信息收集 - 183
为子程序配置用户权限和组权限（仅 UNIX） - 133
为操作配置支持 - 132
从 SystemEDGE 4.3.4 升级代理 - 110
内核配置组 - 170
分区表 - 440
分布式系统组 - 180
升级受管节点和 AIM 服务器 - 109
历史记录收集 - 58, 363
历史记录收集示例 - 371
历史记录收集的工作原理 - 58
历史记录收集概述 - 363
历史记录抽样 - 363
历史记录表 - 181
历史记录表的列 - 366
历史记录采样示例 - 367
历史记录控制表和历史记录数据表 - 364
历史记录控制表的列 - 364
历史记录控制表配置 - 369
反向 OID 查找 - 208
支持多个 SNMP 代理 - 431
支持多个 SystemEDGE 实例 - 432
支持的 MIB - 25, 49
支持的加密协议 - 422, 425
支持的平台 - 425
支持的身份验证协议 - 422, 426
文件系统空间监控 - 169
文件系统表 - 441
无状态监控的工作方式 - 46
日志文件监视器陷阱 - 56
日志文件监控 - 56, 315
日志文件监控示例 - 327
日志文件监控建议 - 336
日志文件监控概述 - 315
日志监视器表 - 182, 316
日志监视器表列 - 316
日志监视器表标志 - 321
日志监视器表操作参数 - 323

日志监控配置 - 325

五划

主机资源 MIB - 29, 435
主机资源 MIB 概述 - 436
主机资源存储组 - 438
主机资源安装软件组 - 442
主机资源设备组 - 438
主机资源系统组 - 437
主机资源运行软件组 - 441
代理未响应 SNMP 请求 - 414
代理和 AIM 升级 - 109
代理热启动 - 165
代理部署 - 102
代理配置 - 115

四划

以传统模式安装代理 - 100

五划

功能和集成概述 - 21
功能概述 - 21
加载 SystemEDGE AIM - 140
加密 SNMPv3 配置文件 - 158
处理器负载跟踪 - 439
处理器表 - 439
对象合计 - 61
对象状态 - 60
平台支持 - 422
平台管理模块 - 68
未接收 SNMP 陷阱消息的管理系统 - 417
术语 - 19
正则表达式示例 - 147
用于 Windows 事件监控的 edgewatch 命令 - 357
用于日志文件监控的 edgewatch 命令 - 332
用于自主监控的 edgemon 命令 - 223
用于自动监测器的 edgewatch 命令 - 281
用于进程组监控的 edgewatch 命令 - 312
用于进程监控的 edgewatch 命令 - 265
用于管理历史记录控制表中的条目的
 emphistory 命令 - 371
用户表 - 170, 172
目录名称查找缓存组 - 175
目录监控 - 316

示例自主监视器表条目 - 197

示例脚本测试 - 381

纠正性操作 - 30

六划

传统监视器 - 66

传统模式操作 - 66

创建受监控的对象 - 410

在 64 位 Linux 版本上安装 SystemEDGE 失败 - 93

在 UNIX 上从命令行安装代理 - 93

在 UNIX 上安装 SystemEDGE 高级加密 - 428

在 UNIX 上卸载 SystemEDGE 高级加密 - 428

在 UNIX 系统上自动启动代理 - 164

在 UNIX 系统上启动/停止代理 - 162

在 UNIX 系统上卸载 SystemEDGE 和 AIM - 107

在 UNIX 和 Linux 系统上安装代理 - 88

在 UNIX 和 Linux 系统上进行安装 - 88

在 Windows 上从命令行安装代理 - 82

在 Windows 上安装 SystemEDGE 高级加密 - 427

在 Windows 上安装代理 - 77

在 Windows 上卸载 SystemEDGE 和 AIM - 105

在 Windows 上卸载 SystemEDGE 高级加密 - 427

在 Windows 系统上安装 - 76

在 Windows 系统上自动启动代理 - 164

在 Windows 系统上启动/停止代理 - 161

在多层次结构中配置监视器合计 - 137

在系统启动时自动启动代理 - 164

在管理软件中使用 Windows 注册表和性能变量 - 392

在管理软件中使用扩展变量 - 382

多行正则表达式 - 146

如何为 ntRegPerf 变量添加单独的 MIB 规格 - 392

如何为 ntRegPerf 变量编辑 empire.asn1 - 392

如何为扩展变量编辑 empire.asn1 - 383

如何为扩展变量编辑单独的 MIB 规格 - 383

安装 SystemEDGE 高级加密 - 426

安装先决条件 - 426

安装和部署 - 75

安装的文件 - 429

安装说明 - 75

安装期间的代理配置 - 76

扩展示例 - 378

扩展组 - 184

扩展组功能 - 376

扩展变量配置 - 376

扩展脚本 - 380

有关进程和服务监控的建议 - 273

有关配置安全的建议 - 142

有状态监控的工作方式 - 45

约定 - 18

自主监视器的状态管理 - 188

自主监视器表 - 180, 190

自主监视器表列 - 191

自主监视器表标志 - 199

自主监视器表操作参数 - 201

自主监视器陷阱 - 52

自主监控 - 48, 187

自主监控示例 - 209

自主监控配置 - 204

自主监控概述 - 187

自动监测器 - 275

自动监测器的工作方式 - 275

自动监测器表 - 186

自定义 MIB 对象 - 375

行创建 - 198

行创建对象 - 243, 298, 320, 344, 366

设备表 - 169, 439

设置绑定地址 - 120

七划

体系结构概述 - 37

删除 IPC 机制 - 174

删除 Windows 事件监控条目 - 360

删除日志监控条目 - 335

删除自主监控条目 - 227

删除进程组监控条目 - 313

删除进程监控条目 - 271

启用性能多维数据集 AIM - 397

启用监视器维护窗口 AIM - 396

启动代理 - 161

启动配置组 - 170

启动期间的初始配置 - 369

更改进程的 nice 值（仅 UNIX） - 171

状态管理 - 25
状态管理和聚合的工作原理 - 63
状态管理配置选项 - 65
状态管理陷阱 - 65
状态管理模型 - 59

六划

私有企业陷阱 - 443

七划

系统组 - 169
运行远程命令 - 172
进程自动监测器示例 - 291
进程间通信组 - 173
进程和服务自动监测器 - 277
进程和服务监控 - 53, 229
进程和服务监控概述 - 229
进程组监视器表 - 185, 293
进程组监视器表标志 - 299
进程组监视器表操作参数 - 301
进程组监视器陷阱 - 55
进程组监控 - 55, 293
进程组监控示例 - 307
进程组监控配置 - 302
进程组监控概述 - 293
进程表 - 171
进程监视器的状态管理 - 231
进程监视器表 - 184, 233
进程监视器表列 - 234
进程监视器表标志 - 245
进程监视器表操作参数 - 250
进程监视器陷阱 - 54
进程监控示例 - 259
进程监控配置 - 252
进程属性 - 241
远程外壳组 - 172
远程配置和部署 - 68
针对 SNMPv1 团体的访问列表 - 123
针对使用扩展的建议 - 384

八划

事件引擎 - 69
使用 CA Virtual Assurance 安装 - 76
使用 CA Virtual Assurance 的配置 - 117
使用 CA Virtual Assurance 部署代理 - 103

使用 diagsysedge 确定代理是否正在运行 - 411

使用 diagsysedge.exe - 411
使用 SystemEDGE 代理的准则 - 72
使用状态管理的示例 - 401
其他参数 - 378
命令行启动选项 - 163
命令行实用工具 - 399
性能多维数据集 AIM - 397
性能组 - 173
性能数据 - 387
抽样进程监视器表条目 - 240
服务自动监测器示例 - 290
注册表数据 - 386
组表 - 171
轮转日志文件 - 338
顶级进程 AIM - 395

九划

卸载 SystemEDGE - 105
卸载已安装的设备 - 170

八划

受管对象创建 - 188, 232

九划

将 SystemEDGE 监视器导入策略 - 112
将信号发送到进程 - 171
指定 SNMPv1 陷阱源 - 123
指定 sysedge.cf 中的自动监测器 - 286
按实例创建监视器 - 41
按需模块 - 71
故障排除和用法建议 - 411
故障排除的诊断日志记录信息 - 414
查看 NT 事件监视器表 - 349
查看历史记录控制表 - 369
查看日志监视器表 - 324
查看自主监视器表 - 203
查看进程组监视器表 - 302
查看进程监视器表 - 252
相关出版物 - 17
绑定失败：地址已被使用 - 418
重新安装 - 108

十划

核心 AIM - 395
消息缓冲区组 - 174
热启动例外 - 165
监视器示例 - 67
监视器和自动监测器的正则表达式示例 - 142
监视器维护窗口 AIM 的限制 - 397
监视器维护窗口模块配置 - 141
监控 Windows 任务排定程序 - 401
监控文件系统 - 403
监控日志文件并通过电子邮件发送通知 - 409
监控功能 - 22
监控体系结构 - 43
监控进程 - 410
监控配置 - 139
监控维护 Windows AIM - 396
监控隔离文件是否存在 - 403
获取故障排除报告 - 412
读者 - 17
通过 SNMP 重新配置扩展脚本 - 405
配置 FIPS 140-2 模式 - 423
配置 GetBulk 响应消息大小 - 128
配置 Linux 可用内存计算 - 136
配置 MIB 表限制 - 133
配置 SNMP Set 限制 - 127
配置 SNMP Set 通知 - 126
配置 SNMP 信息 - 119
配置 SNMP 绑定地址 - 120
配置 SNMPv1 陷阱目标 - 124
配置 SNMPv2c 和 SNMPv3 陷阱目标 - 156
配置 SNMPv3 引擎 ID - 151
配置 SNMPv3 用户信息 - 151
配置 SystemEDGE 命令行实用工具 - 400
配置文本模式排除 - 147
配置文件 - 41
配置设备状态检查限制 - 134
配置访问团体 - 122
配置系统信息 - 118
配置诊断日志记录机制 - 130
配置身份验证失败陷阱 - 126
配置和使用响应文件 - 100

配置适用于 SNMP 用户数据报协议通信的 IP 族 - 119
配置热启动发现 - 136
配置基础 - 115
配置维护模式 - 132
配置联邦信息处理标准模式 - 129
陷阱 PDU 的格式 - 443
陷阱团体表 - 176
验证 AIM 插件 - 413
验证代理在系统初始化时启动 - 415
验证代理响应查询 - 416
验证代理是否正在发送陷阱 - 418
验证代理是否正在运行 - 415
验证代理配置 - 417
验证代理陷阱配置 - 418
验证管理系统配置 - 417
验证管理系统陷阱配置 - 418

十一划

密钥保护 - 424
常见问题 - 413
常规自动监测器 - 277
常规自动监测器示例 - 289
清单跟踪和资产管理 - 439
维护模式 - 165

十二划

插件和集成 - 31
缓冲区组 - 174
联系技术支持 - 3
超级聚合 - 62

十三划

数据流组 - 170
数据流缓冲区分配表 - 175
数据流缓冲区组 - 174
概念 - 37
禁用 SNMPv1 和 SNMPv2c - 158
简介 - 17

十四划

磁盘存储表 - 440
磁盘统计信息表 - 182
聚合状态表 - 185, 190, 233

十六划

操作期间的动态配置 - 369

镜像表 - 185