

CA Application Performance Management

环境性能代理实施指南

版本 9.5



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

本文档涉及以下 CA Technologies 产品和功能:

- CA Application Performance Management (CA APM)
- CA Application Performance Management ChangeDetector (CA APM ChangeDetector)
- CA Application Performance Management ErrorDetector (CA APM ErrorDetector)
- CA Application Performance Management for CA Database Performance (CA APM for CA Database Performance)
- CA Application Performance Management for CA SiteMinder® (CA APM for CA SiteMinder®)
- CA Application Performance Management for CA SiteMinder® Application Server Agents (CA APM for CA SiteMinder® ASA)
- CA Application Performance Management for IBM CICS Transaction Gateway (CA APM for IBM CICS Transaction Gateway)
- CA Application Performance Management for IBM WebSphere Application Server (CA APM for IBM WebSphere Application Server)
- CA Application Performance Management for IBM WebSphere Distributed Environments (CA APM for IBM WebSphere Distributed Environments)
- CA Application Performance Management for IBM WebSphere MQ (CA APM for IBM WebSphere MQ)
- CA Application Performance Management for IBM WebSphere Portal (CA APM for IBM WebSphere Portal)
- CA Application Performance Management for IBM WebSphere Process Server (CA APM for IBM WebSphere Process Server)
- CA Application Performance Management for IBM z/OS® (CA APM for IBM z/OS®)
- CA Application Performance Management for Microsoft SharePoint (CA APM for Microsoft SharePoint)
- CA Application Performance Management for Oracle Databases (CA APM for Oracle Databases)
- CA Application Performance Management for Oracle Service Bus (CA APM for Oracle Service Bus)
- CA Application Performance Management for Oracle WebLogic Portal (CA APM for Oracle WebLogic Portal)

- CA Application Performance Management for Oracle WebLogic Server (CA APM for Oracle WebLogic Server)
- CA Application Performance Management for SOA (CA APM for SOA)
- CA Application Performance Management for TIBCO BusinessWorks (CA APM for TIBCO BusinessWorks)
- CA Application Performance Management for TIBCO Enterprise Message Service (CA APM for TIBCO Enterprise Message Service)
- CA Application Performance Management for Web Servers (CA APM for Web Servers)
- CA Application Performance Management for webMethods Broker (CA APM for webMethods Broker)
- CA Application Performance Management for webMethods Integration Server (CA APM for webMethods Integration Server)
- CA Application Performance Management Integration for CA CMDB (CA APM Integration for CA CMDB)
- CA Application Performance Management Integration for CA NSM (CA APM Integration for CA NSM)
- CA Application Performance Management LeakHunter (CA APM LeakHunter)
- CA Application Performance Management Transaction Generator (CA APM WTG)
- CA Cross-Enterprise Application Performance Management
- CA Customer Experience Manager (CA CEM)
- CA Embedded Entitlements Manager (CA EEM)
- CA eHealth® Performance Manager (CA eHealth)
- CA Insight™ Database Performance Monitor for DB2 for z/OS®
- CA Introscope®
- CA SiteMinder®
- CA Spectrum® Infrastructure Manager (CA Spectrum)
- CA SYSVIEW® Performance Management (CA SYSVIEW)

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：安装、配置和使用 EAgent 11

环境性能代理概述.....	11
EAgent 可从进程接收输入.....	11
EAgent 和脚本环境.....	12
EAgent 可接受来自网络源的输入.....	13
EAgent 和 CA Introscope®.....	14
支持的插件.....	14
环境性能代理安装.....	16
EAgent 独立安装程序.....	16
配置 EAgent.....	17
配置 EAgent 属性.....	17
配置 EAgent 日志记录选项.....	18
配置 EAgent 插件.....	18
为网络数据源配置 EAgent.....	21
运行 EAgent.....	22
在 UNIX 上运行 EAgent.....	23
创建自定义插件并在 EAgent 中使用.....	23
度量标准数据格式.....	23
针对任何一种自定义插件格式类型的预防措施.....	25
错误或事件数据格式.....	25
创建 XML 格式的错误快照.....	26
EAgent 事件和事务跟踪.....	27
错误或事件数据的 XML 架构.....	29

第 2 章：EAgent 默认插件 31

solaris/diskStats.pl.....	31
生成的度量标准.....	31
命令行参数.....	31
windows/processAvailability.pl AND solaris/processAvailability.pl.....	31
生成的度量标准.....	32
命令行参数.....	32
windows/diskStats.pl.....	32
生成的度量标准.....	32
命令行参数.....	33
apacheLogReader.pl、weblogicLogReader.pl 和 websphereLogReader.pl.....	33
生成的度量标准.....	33
命令行参数.....	34

httpSvcAvailability.pl.....	34
生成的度量标准.....	34
命令行参数.....	35
Perl 支持问题	36
kstat.pl.....	36
生成的度量标准.....	36
命令行参数.....	36
Perl 支持问题	37
iisLogReader.pl.....	37
生成的度量标准.....	37
命令行参数.....	38
iisLogReader.cfg 配置文件	38
在 IIS4.0 和 IIS5.0 中配置 IIS Web 服务器日志记录	46

第 3 章：EAgent 默认插件度量标准 49

无效字符替换.....	49
IIS 日志阅读器度量标准	49
网页点击总数.....	50
访问者总数.....	50
<URL> 的网页点击数	50
传输的字节数.....	51
ErrorCount_400_BadRequest	51
ErrorCount_401_Unauthorized.....	51
ErrorCount_403_Forbidden	52
ErrorCount_404_ServerConfiguration	52
ErrorCount_500_InternalServerError	52
ErrorMessage_400_BadRequest.....	53
ErrorMessage_401_Unauthorized	53
ErrorMessage_403_Forbidden.....	53
ErrorMessage_404_PageNotFound	54
ErrorMessage_500_InternalServerError.....	54
HTTP/HTTPS 服务可用性度量标准.....	54
可用	55
响应时间.....	55
平均吞吐量.....	55
Solaris 内核统计信息 (Kstat) 度量标准	56
Solaris 内核 <class> <module> <name> 度量标准	56
进程可用性度量标准.....	56
计数	57
磁盘统计信息 (Windows) 度量标准.....	57
磁盘读取数/秒	58
磁盘写入数/秒	58
磁盘读取字节数/秒	58

磁盘写入字节数/秒	59
当前磁盘队列长度	59
平均磁盘读取时间（毫秒）	59
平均磁盘写入时间（毫秒）	60
平均磁盘字节数/读取	60
平均磁盘字节数/写入	60
可用磁盘空间 (MB)	61
可用磁盘空间 (%)	61
磁盘统计信息 (Solaris) 度量标准	61
磁盘时间 (%)	62
事务等待时间 (%)	62
平均活动事务数	63
平均等待事务数	63
磁盘读取数/秒	63
磁盘写入数/秒	64
每秒读取 KB	64
每秒写入 KB	64
等待队列的平均服务时间（毫秒）	65
活动事务的平均服务时间（毫秒）	65
可用磁盘空间 (MB)	65
可用磁盘空间 (%)	66
Apache 日志度量标准	66
PageHitCount	67
ErrorCount_400_BadRequest	67
ErrorCount_401_Unauthorized	67
ErrorCount_403_Forbidden	68
ErrorCount_404_PageNotFound	68
ErrorCount_500_ServerConfiguration	68
ErrorMessage_400_BadRequest	69
ErrorMessage_401_Unauthorized	69
ErrorMessage_403_Forbidden	69
ErrorMessage_404_PageNotFound	70
ErrorMessage_500_ServerConfiguration	70
WebSphere 日志度量标准	70
MessageCount_Audit	71
MessageCount_Warning	71
MessageCount_EventDebug	72
MessageCount_Error	72
MessageCount_Fatal	72
MessageText_Audit	73
MessageText_Warning	73
MessageText_EventDebug	73
MessageText_Error	74
MessageText_Fatal	74

WebLogic 日志度量标准	77
服务器版本.....	78
MessageCount_Warning.....	78
MessageCount_Error	78
MessageCount_Notice	79
MessageCount_Critical	79
MessageCount_Alert.....	79
MessageCount_Emergency.....	80
MessageText_Warning.....	80
MessageText_Error	80
MessageText_Notice.....	81
MessageText_Critical	81
MessageText_Alert	81
MessageText_Emergency	82

第 4 章：EPAgent 属性 83

IntroscopeEPAgent.properties 文件.....	83
代理和进程名称设置.....	83
代理度量标准老化.....	84
配置代理度量标准老化.....	85
introscope.agent.metricAging.turnOn	86
introscope.agent.metricAging.heartbeatInterval	87
introscope.agent.metricAging.dataChunk	87
introscope.agent.metricAging.numberTimeslices	87
introscope.agent.metricAging.metricExclude.ignore.0	88
区分大小写的度量标准.....	88
introscope.epagent.metricscasesensitive.....	88
ChangeDetector 配置.....	89
introscope.changeDetector.enable.....	89
introscope.changeDetector.rootDir	89
introscope.changeDetector.isengardStartupWaitTimeInSec.....	90
introscope.changeDetector.waitTimeBetweenReconnectInSec.....	90
introscope.changeDetector.enableEPA	90
introscope.changeDetector.agentID.....	91
introscope.changeDetector.profile	91
introscope.changeDetector.profileDir	91
企业管理器连接.....	92
introscope.agent.enterprisemanager.connectionorder	92
introscope.agent.enterprisemanager.transport.tcp.host.DEFAULT	92
EPAgent 配置	92
introscope.epagent.config.networkDataPort	93
introscope.epagent.config.httpServerPort	93
扩展目录.....	93
introscope.agent.extensions.directory	93

日志记录.....	94
log4j.logger.EPAgent.....	94
log4j.appender.logfile.File.....	94
流程.....	95
introscope.agent.customProcessName	95
introscope.agent.defaultProcessName.....	95
introscope.agent.agentName	96
远程配置.....	96
introscope.agent.remoteagentconfiguration.enabled	96
有状态插件.....	96
introscope.epagent.plugins.stateful.names.....	97
introscope.epagent.stateful.<pluginname>.command.....	97
introscope.epagent.stateful.<pluginname>.class	98
无状态插件.....	98
introscope.epagent.plugins.stateless.names	98
introscope.epagent.stateless.<pluginname>.command.....	99
introscope.epagent.stateless.<pluginname>.class	99
introscope.epagent.stateless.<pluginname>.delayInSeconds.....	99
introscope.epagent.stateless.<pluginname>.schedule.....	100
introscope.epagent.config.stalledStatelessPluginTimeoutInSeconds	100
introscope.epagent.stateless.<pluginname>.metricNotReportedAction	101

第 1 章： 安装、配置和使用 EPAgent

本节介绍了如何安装、配置和使用 CA APM 环境性能代理 (EPAgent)。

此部分包含以下主题：

[环境性能代理概述](#) (p. 11)

[环境性能代理安装](#) (p. 16)

[配置 EPMagent](#) (p. 17)

[运行 EPMagent](#) (p. 22)

[在 UNIX 上运行 EPMagent](#) (p. 23)

[创建自定义插件并在 EPMagent 中使用](#) (p. 23)

环境性能代理概述

CA APM 环境性能代理 (EPAgent) 可将非 Java 源的度量标准数据集成到 CA Introscope® 中。

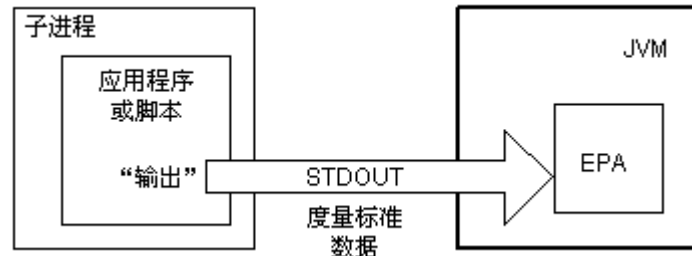
EPAgent 利用可在许多操作系统中使用的功能，使单个进程能繁衍子进程，并通过操作系统管道（标准 IPC 机制）从子进程接收标准输出。将子进程启动功能和灵活的脚本环境（例如 Perl）结合在一起时，就会形成一种非常强大且适应性很强的功能，几乎可以从任何源收集应用程序性能信息。

EPAgent 可从进程接收输入

EPAgent 可在 Java 中实施并在 Java 虚拟机 (JVM) 中运行。EPAgent 使用 JVM 功能调用子进程，并从该进程处接收 STDOUT 和 STDERR。子进程可以是任何脚本、可执行文件或者任何可在命令行上调用的内容。例如：

- 用户以 C 语言编写的编译程序
- Perl 脚本

通过执行简单的 PRINT 函数，子进程中的应用程序或脚本可以与 EAgent 通信。下图显示了应用程序或脚本如何通过输出至 STDOUT 将度量标准数据传送至 EAgent。



EAgent 希望其使用 STDOUT 接收的文本符合下述指定格式之一：

- 简单的名称/值对

或

- XML

本指南后面部分将详细说明 EAgent 文本格式的要求和选项。

在本指南中，向 EAgent 发送数据的应用程序或脚本称为 EAgent 插件。EAgent 包括提供详细度量标准数据的几个默认插件。您也可以创建自定义插件。

详细信息：

[配置 EAgent 插件](#) (p. 18)

[创建自定义插件并在 EAgent 中使用](#) (p. 23)

[EAgent 默认插件](#) (p. 31)

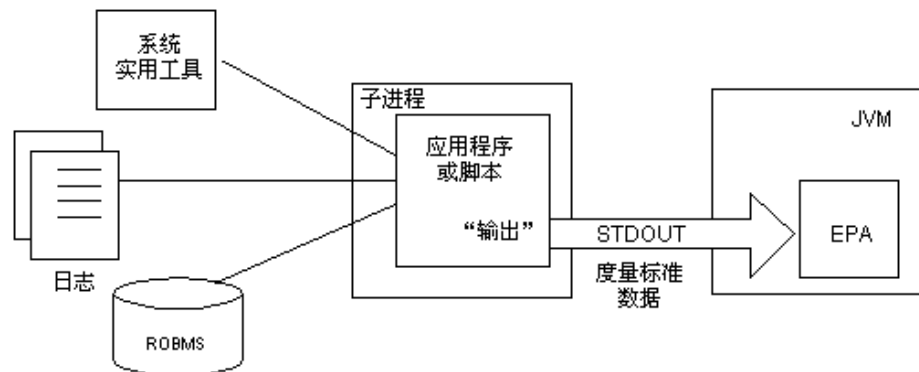
EAgent 和脚本环境

尽管 EAgent 可以从子进程的任何可执行实体（包括编译的应用程序）接收文本，但最灵活的方法仍是使用脚本环境，例如 Perl、KornShell 或 rexx。解释性的脚本环境具有易于实施的优势，并且通常为包括关系数据库、OS 子系统（进程表、文件系统等）和第三方产品在内的现实数据来源提供库或界面。

脚本可以从各种源收集信息。例如，脚本可以：

- 调用库函数或系统实用工具以查看是否存在关键应用程序，如数据库。
- 扫描并解析应用程序日志文件以检测应用程序错误。

脚本环境的灵活性让 EPAgent 几乎可以从任何源收集性能和管理信息。下图显示了脚本从多个源访问信息的方式。



由于 Perl 解释器适用于各种平台，并且在多种操作系统、中间件和第三方产品上广泛支持数据 API，因此 CA Technologies 强烈建议将 Perl 与 EPAgent 结合使用。CA Introscope® EPAgent 包括一组 Perl 示例脚本，可用于执行检查进程可用性、获取磁盘性能统计信息和读取 HTTP 日志等各种功能。这些脚本通过扩展后可用于执行其他功能。

注意：有关对 PBD、EPAgent 脚本或 JavaScript 计算器的自定义项的支持，请联系 CA Technologies 专业服务。

EPAgent 可接受来自网络源的输入

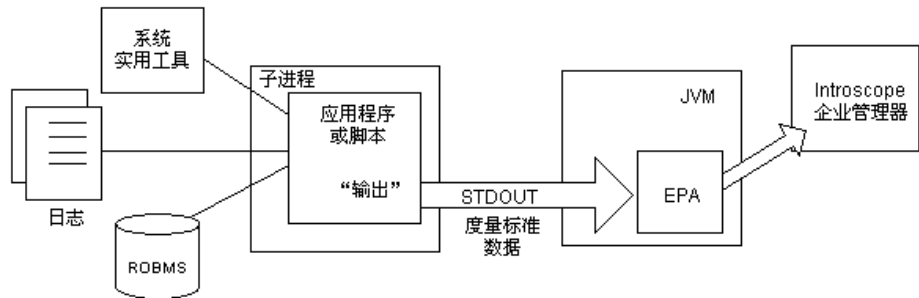
除接受本地进程的输入之外，EPAgent 还可以接受网络源的输入。此能力有助于从联网进程（无法在其中运行 EPAgent）收集信息，例如，C++ 后端系统，该系统使用网络端口提供监控信息，或者定期发布数据到 Web URL。

详细信息：

[为网络数据源配置 EPAgent](#) (p. 21)

EPAgent 和 CA Introscope®

EPAgent 以独立进程的方式运行。与 Java 代理或 .NET 代理类似，EPAgent 也可通过连接到企业管理器来传送管理和性能信息。下图显示了 EPAgent 如何将数据转发到企业管理器。



使用 EPAgent 可以更全面地了解您的应用程序环境。分布式应用程序体系结构很复杂，通常由 Java 和非 Java 应用程序子系统混合组成。通过 EPAgent 可以方便地从应用程序环境的非 Java 部分和非 .NET 部分收集信息，以便在 CA Introscope® Workstation 中与 Java 和 .NET 度量标准一起显示。

支持的插件

EPAgent 支持有状态插件和无状态插件。

注意：所有默认插件均要求使用 Perl 5.00503 或以上版本。

插件类型

以下为可用插件类型：

- **有状态插件**—有状态插件通常为长期运行脚本（后台进程），在 EPAgent 启动时启动并一直运行下去。这些有状态插件可随时通过插件的标准输出通道将数据反馈到 CA Introscope® 中。如果有状态插件终止，EPAgent 将重新启动该插件。
- **无状态插件**—无状态插件设计为按定期计划运行，并可配置为适合运行的频率（指定为两次运行之间的延迟）。这些插件通常为短期运行脚本，其目的是收集一些数据，通过标准输出通道将这些数据发送至 EPAgent，然后终止。

注意：EPAgent 不进行特定的错误检查以确保一次只运行一个无状态插件实例，因此插件开发者必须将无状态插件设计为在合理的短时期内运行并完成。

错误日志

可以通过插件的标准错误通道记录错误，这样 EPAgent 就能够像记录任何其他错误一样记录该错误输出。

默认插件

EPAgent 包含一个只需很少的配置即可使用的默认插件库。下表概述了各插件的功能以及支持这些插件的操作系统。

插件	功能	支持系统
进程可用性检查	查找具有给定名称的运行进程	Solaris
OS 级磁盘统计信息	报告读取/写入、事务处理时间、等待时间百分比、磁盘容量等	Solaris
Apache 日志阅读器	报告 HTTP 错误数、事务数和平均事务大小	Solaris
WebLogic 日志阅读器	报告错误计数和警告计数	Solaris
WebSphere 日志阅读器	报告错误计数和警告计数	Solaris
IIS 日志阅读器	报告错误数（按类型）、提供的页数、用户数（按 IP 地址）	Windows
HTTP 服务可用性 HTTPS 服务可用性	报告连接成功/失败、连接往返时间、平均吞吐量/带宽	Solaris
Solaris Kstat 统计信息	报告可通过 Kstat 实用工具获得的任何统计信息，例如输入/输出统计信息、所有设备的网络统计信息、虚拟内存统计信息	Solaris

自定义插件

EPAgent 支持收集任意程序和插件的数据，但要求它们生成的输出符合支持的两种格式之一。

详细信息：

[创建自定义插件并在 EPAgent 中使用 \(p. 23\)](#)

环境性能代理安装

使用独立的代理安装程序安装 EPAgent。

注意：有关详细信息，请参阅《CA APM 安装和升级指南》。

EPAgent 独立安装程序

可以使用以下独立代理安装程序之一将 EPAgent 安装在其他计算机上。您可以从 [CA Support](#) 上的 CA APM 软件下载区域下载 CA APM 产品。

- EPAgent<version>windows.zip
- EPAgent<version>ebcdic.tar
- EPAgent<version>unix.tar

解压缩 EPAgent 安装程序后，将以下文件安装在 <EPAgent_Home>/epagent 目录中。

注意： <EPAgent_Home> 是已安装 EPAgent 的计算机。

- IntroscopeEPAgent.properties
- lib/EPAgent.jar
- epaplugins/（插件的顶层目录）
 - epaplugins/apache/apacheLogReader.pl
 - epaplugins/iis/iisLogReader.pl
 - epaplugins/solaris/diskStats.pl
 - epaplugins/solaris/kstat.pl
 - epaplugins/solaris/httpSvcAvailability.pl
 - epaplugins/solaris/processAvailability.pl
 - epaplugins/weblogic/weblogicLogReader.pl
 - epaplugins/websphere/websphereLogReader.pl
 - epaplugins/windows/diskStats.pl
 - epaplugins/windows/httpSvcAvailability.pl
 - epaplugins/windows/processAvailability.pl

- /xml (ChangeDetector 文件的顶级目录)
 - ChangeDetectorAgent.jar
 - ChangeDetector-Agent_Server.jar
 - ChangeDetector-CommonAll.jar
 - ChangeDetector-config.xml
 - Windows: cdnativefile.dll
 - UNIX: libcdnativefile.so

配置 EPAgent

EPAgent 包括一组代理配置属性，如企业管理器位置和端口设置。

注意：EPAgent 在属性文件设置中遇到任何错误（例如，文件名路径无效）时，EPAgent 将无法启动。如果您要在启动时自动启动 EPAgent，请确认已正确设置这些属性。

详细信息：

[配置 EPAgent 属性 \(p. 17\)](#)

[配置 EPAgent 日志记录选项 \(p. 18\)](#)

[配置 EPAgent 插件 \(p. 18\)](#)

[为网络数据源配置 EPAgent \(p. 21\)](#)

配置 EPAgent 属性

EPAgent 所用属性与 Java 代理所用属性相同。

遵循这些步骤：

1. 在 *IntroscopeEPAgent.properties* 文件中配置 EPAgent 设置。EPAgent 设置的类型与 CA Introscope® 代理配置文件中的设置类型相同。

注意：有关属性的详细信息，请参阅《CA APM Java 代理实施指南》。

2. 如果更改 *IntroscopeEPAgent.properties* 文件的名称或位置，您可以使用 Java 系统属性对其进行设置：

`-Dcom.wily.introscope.epagent.properties=filename`

注意：该系统属性应当紧跟在命令行中 "java" 之后。如果将其置于命令行中比较靠后的位置（例如，在 `-jar` 之后），将不起作用。

配置 EPAgent 日志记录选项

默认情况下，EPAgent 将消息和错误输出发送至命令控制台。您可以配置 EPAgent，将消息和错误输出同时发送至日志文件。

遵循这些步骤:

1. 打开 `<EPAgent_Home>/config/IntroscopeEPAgent.properties` 文件。
2. 修改属性。

以详细模式运行 EPAgent 并重定向输出文件

在详细模式下运行 EPAgent 时，系统会将详细信息记录到日志中，这有助于调试。控制详细模式的属性也控制输出日志文件的放置。

遵循这些步骤:

1. 打开 `<EPAgent_Home>/epagent/IntroscopeEPAgent.properties` 文件。
2. 注释掉属性 `log4j.logger.EPAgent`，例如：
`#log4j.logger.EPAgent=INFO, console`
3. 取消注释属性 `log4j.logger.IntroscopeAgent`，将 `INFO` 替换为 `VERBOSE#com.wily.util.feedback.Log4JSeverityLevel`，并将 `console` 替换为 `logfile`，以便将输出发送至日志文件而非控制台。例如，如果想要 EPAgent 以详细模式报告至日志文件，该属性应如下所示：
`log4j.logger.IntroscopeAgent=VERBOSE#com.wily.util.feedback.Log4JSeverityLevel,logfile`
4. 保存更改并关闭 `IntroscopeEPAgent.properties` 文件。

重要信息！ 保存 `IntroscopeEPAgent.properties` 文件之后，EPAgent 日志仍会显示在命令控制台中，直至属性文件读取并实施完毕。实施属性文件后，会开始将日志写入日志文件。

配置 EPAgent 插件

要在运行 EPAgent 时使用默认插件，只需取消注释 `IntroscopeEPAgent.properties` 文件中的特定属性即可。

但是，您可能想要删除 `IntroscopeEPAgent.properties` 文件默认插件中不需要的插件，也可能想要添加其他插件。

在 `IntroscopeEPAgent.properties` 文件中，EPAgent 插件分为两个部分：有状态插件和无状态插件。

添加有状态插件

在 *IntroscopeEPAgent.properties* 文件中配置 EPAgent 使用的插件。

遵循这些步骤:

1. 打开 `<EPAgent_Home>/epagent/IntroscopeEPAgent.properties` 文件。
2. 取消注释属性 `introscope.epagent.plugins.stateful.names`，并为所有有状态插件添加以逗号分隔的名称，如下例所示：
`introscope.epagent.plugins.stateful.names=APACHE,WEBLOGIC,WEBSPHERE,IIS`
3. 为前一步骤中添加的各个有状态插件名称添加一行，指定启动有状态插件所需的完整命令，包括所需的任何命令行参数。

注意：在文件中取消注释默认属性后即可使用这些属性，但必须提供启动插件所需的完整命令行。

前一步骤中指定的有状态名称属性需要四行：

```
introscope.epagent.stateful.APACHE.command=/usr/bin/perl
<EPAgent_Home>/epagent/epaplugins/apache/apacheLogReader.pl
introscope.epagent.stateful.WEBLOGIC.command=/usr/bin/perl
<EPAgent_Home>/epagent/epaplugins/weblogic/weblogicLogReader.pl
introscope.epagent.stateful.WEBSPHERE.command=/usr/bin/perl
<EPAgent_Home>/epagent/epaplugins/websphere/websphereLogReader.pl
introscope.epagent.stateful.IIS.command=/usr/bin/perl
<EPAgent_Home>/epagent/epaplugins/iis/iisLogReader.pl
```

注意：在 Windows 计算机上，添加路径时，反斜杠 (\) 必须用另一个反斜杠转义（每次使用双反斜杠），例如：

```
introscope.epagent.stateful.TEST.command=perl c:\\mydirectory\\myscript.pl
```

详细信息：

[EPAgent 默认插件](#) (p. 31)

后续步骤

您可以[添加无状态插件](#) (p. 20)。如果不添加无状态插件，请将更改保存至 *IntroscopeEPAgent.properties* 文件，并重新启动 EPAgent。

添加无状态插件

您可以添加无状态插件。

遵循这些步骤:

1. 打开 `<EPAgent_Home>/config/IntroscopeEPAgent.properties` 文件。
2. 取消注释属性 `introscope.epagent.plugins.stateless.names` 中，然后为所有无状态插件添加以逗号分隔的逻辑名称，如下例所示：

```
introscope.epagent.plugins.stateless.names=DISK,ORACLE,HTTP,KSTAT
```

3. 为前一步骤中添加的每个无状态插件名称另外添加两行，指定：

- 启动无状态插件所需的完整命令行（包括所需的任何命令行参数，请参阅“EPAgent 默认插件”），语法如下：

```
introscope.epagent.plugins.stateless.<pluginname>.command= <command>
```

- EPGent 运行插件的频率，以延迟的秒数来表示，语法如下：

```
introscope.epagent.stateless.<pluginname>.delayInSeconds=<value in seconds>
```

例如：

```
introscope.epagent.stateless.DISK.command=/usr/bin/perl
<EPAgent_Home>/epagent/epaplugins/solaris/diskStats.pl
introscope.epagent.stateless.DISK.delayInSeconds=900
introscope.epagent.stateless.ORACLE.command=/usr/bin/perl
<EPAgent_Home>/epagent/solaris/processAvailability.pl
introscope.epagent.stateless.ORACLE.delayInSeconds=900
introscope.epagent.stateless.HTTP.command=/usr/bin/perl
<EPAgent_Home>/epaplugins/solaris/httpSvcAvailability.pl
introscope.epagent.stateless.HTTP.delayInSeconds=900
introscope.epagent.stateless.KSTAT.command=/usr/bin/perl
<EPAgent_Home>/epaplugins/solaris/kstat.pl
introscope.epagent.stateless.KSTAT.delayInSeconds=900
```

注意：这些示例为这些属性的默认值。要使用这些值，请取消注释各属性。

4. 完成后，保存更改并重新启动 EPGent。

详细信息:

[EPAgent 默认插件](#) (p. 31)

删除有状态插件和无状态插件

您可以随意删除插件。

遵循这些步骤：

1. 打开 `<EPAgent_Home>/epagent/IntroscopeEPAgent.properties` 文件。
2. 在属性 `introscope.epagent.plugins.stateful.names` 和 `introscope.epagent.plugins.stateless.names` 中，删除要删除的插件名称。
3. 注释掉或删除与要删除的插件关联的任何命令行。
4. 完成后，保存更改并重新启动 EPAgent。

配置 Java 类插件

要配置在 EPAgent JVM 中执行的 Java 类插件，请在 EPAgent 属性文件中配置这些属性：

```
introscope.epagent.stateful.JAVA-CLASS.class=my.package.path.Class arg1 arg2 arg3
```

注意： 请提供完全限定的类名称，后跟任意可选参数，例如：

```
introscope.epagent.plugins.stateful.names=add  
introscope.epagent.stateful.add.class=com.wily.math.Add 1 2
```

为网络数据源配置 EPAgent

可以配置 EPAgent，使之接受网络源数据。

- [为简单或 XML 网络输入配置 EPAgent \(p. 21\)](#)。
- [为 HTTP GET 输入配置 EPAgent \(p. 21\)](#)。

为简单或 XML 网络输入配置 EPAgent

要让 EPAgent 接受网络数据，请取消注释 EPAgent 属性文件中的这一行：

```
introscope.epagent.config.networkDataPort=8000
```

为 HTTP GET 输入配置 EPAgent

可以在侦听 HTTP GET 请求的 EPAgent 中创建简单的 HTTP 服务器。HTTP GET 请求确定发送至企业管理器的度量标准。

HTTP GET 请求使用以下语法：

```
http://epaHost:epaPort/?metricType=type&metricName=name&metricValue=value
```

启用和测试简单的 HTTP 服务器

1. 在 EPAgent 属性文件中取消注释以下属性以启用简单的 HTTP 服务器。

```
introscope.epagent.config.httpServerPort=8080
```

2. 重新启动 EPAgent。

这些更改将生效。

3. 通过浏览器测试 HTTP GET 请求。例如：

```
http://localhost:9090/?metricType=IntCounter&metricName=MyHTTPMetric|Test:MyCounter&metricValue=15
```

15 秒后，在 EPAgent 下会显示 MyHTTPMetric|Test:MyCounter 度量标准。

要更新度量标准，EPAgent 必须收到新的 HTTP GET 请求。因此，简单的 HTTP 服务器功能适用于能够发出多个 HTTP GET 请求的工具。例如，可以使用开放源代码工具 Wget 发出多个 wget 命令来创建多个 HTTP GET 请求。有关 Wget 的详细信息，请访问：

<http://www.gnu.org/software/wget/wget.html>

运行 EPAgent

EPAgent 可以作为独立 .jar 文件或 Java 应用程序来运行。

以独立 .jar 文件运行 EPAgent

- 运行带有相应 `-jar` 标志的 Java 命令行，如下例所示：

```
java -jar <EPAgent_Home>/epagent/lib/EPAgent.jar  
-Dcom.wily.introscope.epagent.properties="<EPAgent_Home>/epagent/IntroscopeEPAgent.properties"
```

以 Java 应用程序运行 EPAgent

- 将 EPAgent 文件添加到相应的类路径，如下例所示：

```
java -classpath "<EPAgent_Home>/epagent/lib/EPAgent.jar"  
-Dcom.wily.introscope.epagent.properties="<EPAgent_Home>/epagent/IntroscopeEPAgent.properties"
```

在 UNIX 上运行 EPAgent

可以使用控制脚本（shell 脚本）在 UNIX 操作系统上运行 Introscope EPAgent。

遵循这些步骤：

1. 打开命令提示符。
2. 导航到包含该控制脚本的目录。例如：

```
cd Introscope<version_number>/bin
```

3. 运行与所需操作相对应的命令：

EPACtrl.sh start

启动 EPAgent。

EPACtrl.sh status

显示 EPAgent 进程的状态（运行还是停止）。

EPACtrl.sh stop

停止 EPAgent 进程。

EPACtrl.sh help

显示帮助菜单。

创建自定义插件并在 EPAgent 中使用

运行 EPAgent 时使用默认插件，收集的信息通常足以满足用户的需求。但是，您可能想在 EPAgent 中使用自定义插件。为此，请遵循以下两个准则：

- 确保自定义插件的输出可以由 EPAgent 解析（参阅 [“度量标准数据格式”](#) (p. 23) 和 [“错误或事件数据格式”](#) (p. 25)）。
- 根据 [“配置 EPAgent 插件”](#) (p. 18) 一节中的说明，在 *IntroscopeEPAgent.properties* 文件中添加自定义插件。

度量标准数据格式

EPAgent 可以解析插件（或插入 EPAgent 中的其他生成度量标准的程序）提供的两种不同格式的度量标准数据：

- 简单
- XML

度量标准数据的简单格式

按以下格式每行指定一个度量标准名称和值：

```
<metric_name>=<value>
```

例如：

```
diskwrites=37
```

也可以在其中包括资源段参考：

```
<resource_segment>:<metric_name>=<value>
```

例如（全部在同一行）：

```
Resource Usage|File IO:diskwrites=37
```

```
Apache Errors:LastErrorString=ERROR: Apache shutdown unexpectedly
```

简单格式准则：

- 在“简单”格式中，度量标准名称不应包含等号 (=)。如果名称中需要使用等号，请使用 XML 格式。
- 值可能包含等号（字符串），EPAgent 总是将第一个等号之前的所有字符（从左到右）解析为度量标准名称，并将第一个等号之后的所有字符解析为值。
- 由数字组成的任何值都将视为数值数据，并将作为 CA Introscope® “IntCounter” 类型进行报告。
- 由数字以外的任何内容组成的任何值都将视为字符串数据，并将作为 CA Introscope® “字符串事件” 类型进行报告。

度量标准数据的 XML 格式

“简单”格式限制了 Introscope 度量标准类型，而 XML 格式允许插件报告其他信息（例如 Introscope 度量标准名称、Introscope 度量标准类型和值），如下例所示：

```
<metric type="LongCounter" name="Resource Usage|File IO:diskwrites" value="37" />
```

```
<metric type="StringEvent" name="Apache Errors:LastErrorString" value="ERROR: Apache shutdown unexpectedly" />
```

XML 格式准则：

- 此格式全面支持在度量标准名称和值中使用 Introscope 数据类型和等号。
- 度量标准的 "type" 属性必须是以下属性之一：
 - PerIntervalCounter 一值为“每个时间间隔”速率，其中间隔时间可以更改。这些度量标准通过合计值随时间聚合。例如，如果每 15 秒有 10 次方法调用，随后每 15 秒 15 次方法调用，那么聚合到 30 秒则为“每 30 秒 25 次方法调用”。

- IntCounter—int 值可上下浮动
- IntAverage—某段时间的平均 int 值
- IntRate—每秒速率值。这些度量标准是对某段时间的值进行平均后合计出来的。
- LongCounter—long 值可上下浮动
- LongAverage—某段时间的平均 long 值
- StringEvent—表示可周期性生成字符串的类型。该记录器没有当前值的概念；它仅仅按照事件向其报告的顺序来报告事件。
- Timestamp—一种类型，可生成连续递增型时间戳。
- 比较不区分大小写（目的是方便插件编写者）。如果提供的是数字类型，但值为非数字值，则不会向 Introscope 报告任何信息，但 EPAgent 将记录错误。

针对任何一种自定义插件格式类型的预防措施

使用自定义插件格式的任一类型时，请考虑以下预防措施：

- 支持两种格式所引发的交互将导致系统不识别以小于号开头的度量标准名称。
- 对于这两种格式，如果格式不可解析（乱码或错误语法），EPAgent 将忽略行并记录错误。
- 如果插件返回了多个行，将继续解析下一行。
- 每个度量标准名称只能指定一个度量标准类型。如果指定一个以上的类型，会发生以下错误：

```
mm/dd/yy hh:mm:ss PM PDT [ERROR] [EPAgent] Metric name from plugin 'Plugin
<plugin_name>' is invalid: "<metric_name>" is already in use by another
DataRecorder of a different type
```

错误或事件数据格式

EPAgent 可解析插件提供的两种不同格式的错误或事件数据：

- 简单
- XML

错误或事件数据的简单格式

一般来说，简单格式脚本以下列固定字符串开头：

event:

冒号后的所有内容：

- 是 "name=value" 对的一部分，每对都以和号字符 (&) 分隔开。
- 是事件的可选参数。

以下示例是监控 Firefox 浏览器进程并在浏览器退出时发送通知的假设脚本的输出。

```
event:type=processWentAway&processName=firefox
```

错误或事件数据的简单 XML 格式

事件也可以采用 XML 格式指定，该格式可在代理中对事件进行完整的表述。最简单的 XML 格式事件提供生成事件的资源的名称（例如“连接池”或“Java 虚拟机”）。以下示例表示在 *Some Resource* 中发生了某事件的通知。

```
<event resource="Some Resource"/>
```

时间戳是事件的创建时间，事件的持续时间为零。

错误或事件数据的带有参数和时间数据的 XML 格式

可以为事件通知配置显式时间戳和显式持续时间。时间戳格式是 Java 可解析的任何格式。持续时间以毫秒为单位。以下示例是持续时间为一分钟的事件（6 万毫秒）：

```
<event resource="Some Resource" startTime="123003000" duration="60000">  
  <param name="urgent" value="true"/>  
</event>
```

创建 XML 格式的错误快照

错误快照必须在参数中将其类型标示为错误快照：

```
<event resource="Some Resource" startTime="123003000" duration="60000">  
  <param name="Trace Type" value="ErrorSnapshot"/>  
</event>
```

嵌套组件

以下示例显示了具有嵌套子组件的事件。事件可以有零到无限个子组件，每个子组件也可以有零到无限个子组件。实际上，嵌套层一般较小或为零。

```
<event resource="Some Resource">
  <calledComponent resource="Another Resource">
    <param name="isCorrelated" value="uncertain"/>
    <calledComponent resource="A Third Resource"/>
    <calledComponent resource="A Fourth Resource"/>
  </calledComponent>
</event>
```

EPAgent 事件和事务跟踪

您可以选择“跟踪视图”选项卡，在事件查看器中将 EPAgent 事件作为事务跟踪来查看。

当 EPAgent 发送的事件中包含时间信息时，跟踪视图会比较容易理解。要包含时间信息，请在 `<event>` 和 `<calledComponent>` 标记上使用 `startTime` 和 `offset` 属性。

`startTime` 属性是绝对时间。其格式是 `java.util.Date.parse()` 可以解析的任何格式。不需要在 `<event>` 元素中指定 `startTime`—如果没有该属性，那么根据 Java 方法 `System.currentTimeMillis()` 或 `new Date().getTime()` 的指定，起始时间默认为当前时间值。如果在 `<calledComponent>` 元素中忽略 `startTime`，时间会默认为包含元素的时间。因此，如果没有在任何位置指定 `startTime` 属性，所有起始时间都将默认为当前时间。

`offset` 属性是整数值。它解释为以毫秒为单位的时间，可添加至 `startTime` 属性（无论 `startTime` 是默认时间还是显式指定的时间）以生成 `<event>` 或 `<calledComponent>` 的实际报告时间。

示例 1

```
<event resource="自定义 Web 服务器" startTime="123456789" duration="500">
  <calledComponent resource="Web 服务器模块" offset="300" duration="100"/>
</event>
```

该事件跟踪视图的“自定义 Web 服务器”从 123456789 开始，“Web 服务器模块”从 123457089 (123456789 + 300) 开始。在各元素中指定持续时间将生成有用的跟踪视图，其中显示：

- “自定义 Web 服务器”运行 300 毫秒
- “Web 服务器模块”由“自定义 Web 服务器”调用并运行 100 毫秒
- “自定义 Web 服务器”在“Web 服务器模块”返回后又运行 100 毫秒

示例 2

```
<event resource="自定义 web 服务器" duration="500">  
<calledComponent resource="web 服务器模块" offset="300" duration="100"/>  
</event>
```

本示例与示例 1 类似，除了“自定义 Web 服务器”从当前时间开始，“Web 服务器模块”从 300 毫秒后开始。请注意为何本示例中没有任何部分需要 EPAgent 脚本来获知当前时间。

示例 3

```
<event resource="自定义 web 服务器" startTime="123000000" offset="1000"  
duration="5000">  
<calledComponent resource="web 服务器模块" startTime="123003000"  
duration="200"/>  
</event>
```

这里的“自定义 Web 服务器”从 123001000 (123000000 +1000) 开始，“Web 服务器模块”从 123003000 开始。

请再次注意如何指定持续时间可提升可读性和可用性。错误指定“起始时间”、“偏移量”和“持续时间”会使跟踪视图难以读取，因此在使用时务必小心。特别是开始时间（通过添加 `<calledComponent>` 元素的 `startTime` 和 `offset` 来计算）应该始终在其包含的 `<event>` 或 `<calledComponent>` 的起始时间之后，并且 `<calledComponent>` 的（开始时间 + 持续时间）应该总在其包含的 `<event>` 或 `<calledComponent>` 的（开始时间 + 持续时间）之前。

错误或事件数据的 XML 架构

支持的正式 XSD 架构为:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:element name="event" type="eventElement">
    <xs:annotation>
      <xs:documentation>The root element for events. This element is nearly
equivalent to the calledComponent element, except that the event element must
occur only once, at the outermost level.</xs:documentation>
    <xs:annotation>
    </xs:annotation>
  </xs:element>
  <xs:element name="param">
    <xs:complexType>
      <xs:attribute name="name" type="xs:string" use="required"/>
      <xs:attribute name="value" type="xs:string" use="required"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="calledComponent" type="eventElement">
    <xs:annotation>
      <xs:documentation>A component called by the containing element. This
element is nearly equivalent to the event element, except that this element cannot
occur at the outermost level.
    </xs:documentation>
    <xs:annotation>
    </xs:annotation>
  </xs:element>
  <xs:complexType name="eventElement">
    <xs:sequence>
      <xs:element ref="param" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="calledComponent" minOccurs="0"
maxOccurs="unbounded"/>
    <xs:sequence>
      <xs:attribute name="startTime" type="xs:dateTime" use="optional"/>
      <xs:attribute name="offset" type="xs:integer" use="optional"
default="0"/>
      <xs:attribute name="duration" type="xs:dateTime" use="optional"
default="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```


第 2 章：EAgent 默认插件

本节介绍了 CA APM 环境性能代理提供的默认插件及其可选命令行参数。

此部分包含以下主题：

[solaris/diskStats.pl](#) (p. 31)

[windows/processAvailability.pl](#) AND [solaris/processAvailability.pl](#) (p. 31)

[windows/diskStats.pl](#) (p. 32)

[apacheLogReader.pl](#)、[weblogicLogReader.pl](#) 和 [websphereLogReader.pl](#) (p. 33)

[httpSvcAvailability.pl](#) (p. 34)

[kstat.pl](#) (p. 36)

[iisLogReader.pl](#) (p. 37)

solaris/diskStats.pl

该插件返回度量标准，其中包含有关系统硬盘的信息。

生成的度量标准

Solaris 内核统计信息 (Kstat) 度量标准中详细描述了由 */solaris/diskStats.pl* 插件生成的度量标准。

命令行参数

该插件将磁盘名称作为命令行参数。如果没有指定磁盘名称，将报告所有磁盘的相关信息。

- 例如，以下命令报告所有本地磁盘的度量标准：
`perl diskStats.pl`
- 以下命令仅报告磁盘 '/' 和 '/usr' 上的度量标准：
`perl diskStats.pl / /usr`

注意：您可能想使用该参数来缩小结果范围。

windows/processAvailability.pl AND solaris/processAvailability.pl

该插件返回度量标准，其值等于与传入参数值相匹配的进程名称的数目。返回的度量标准类型是 'IntCounter'。

生成的度量标准

“进程”可用性度量标准中详细描述了 */windows/processAvailability.pl* 和 *solaris/processAvailability.pl* 插件生成的度量标准。

命令行参数

这些插件中的每个都有五个可选的命令行参数：

match

用于搜索的正则 Perl 表达式

resource

度量标准名称的资源部分

subResource

度量标准名称的子资源部分

metric

度量标准名称的主要部分

distinctMatch

该参数没有值，用于确定是否为各正则表达式 **match** 分别保存和报告计数。该选项在指定的情况下会覆盖 **subResource** 参数的值。

以下示例报告了以字符 'sh' 开头的进程数量：

```
perl processAvailability.pl -match ^sh
```

windows/diskStats.pl

该插件可根据 Windows 性能监控器提供的统计信息报告磁盘使用情况/容量统计信息。

为使此插件能够正常报告所有统计信息，必须发出 'diskperf -y' 命令。之后必须重新启动计算机以激活新设置。

生成的度量标准

磁盘统计信息 (Windows) 度量标准中详细描述了 */windows/diskStats.pl* 插件生成的度量标准。

命令行参数

该插件将磁盘名称作为命令行参数。如果没有指定磁盘名称，将报告所有磁盘的相关信息。

- 例如，以下命令报告所有本地磁盘的度量标准：
`perl diskStats.pl`
- 以下命令仅报告磁盘 c: 和 d: 的度量标准
`perl diskStats.pl c: d:`

注意：您可能想使用该参数来缩小结果范围。

apacheLogReader.pl、weblogicLogReader.pl 和 websphereLogReader.pl

这些插件根据 <webserver>LogReader.cfg 文件中配置的度量标准来报告度量标准。

每个 Web 服务器应用程序日志文件解析器均自带配置文件，并为睡眠延迟、子资源名称、资源名称、日志文件目录以及日志文件名称模式设置了默认值。

注意：使用任何这些日志阅读器插件时，除非有新条目添加到日志文件中，否则 EAgent 不会向企业管理器报告。

生成的度量标准

这些插件生成的度量标准详细记录在以下度量标准中：

- Apache 日志度量标准
- WebSphere 日志度量标准
- WebLogic 日志度量标准

命令行参数

这些插件中的每个插件都有四个可选的命令行参数：

logfileDir

日志文件所在目录

sleepTime

两次检查日志文件中是否有新消息之间的延迟秒数

logfile

与生成的日志文件名称相对应的正则表达式

subResource

度量标准名称的子资源部分

示例：

```
perl apacheLogReader.pl -sleepTime 30 -logfileDir "c:/Apache/logs"
```

httpSvcAvailability.pl

该插件根据指定 URL 的 HTTP 服务可用性和性能来报告度量标准。

生成的度量标准

HTTP/HTTPS 服务可用性度量标准中详细描述了 *httpSvcAvailability.pl* 插件生成的度量标准。

命令行参数

该插件具有三个可选的命令行参数：

url

一组包含可选端口号的逗号分隔的 URL。URL 可包含端口号（如果与默认端口不同），并使用 `http:` 或 `https:` 来区分 HTTP 和 HTTPS。

get

请求逗号分隔的 HTTP 请求列表（无空格）以发布到服务器。

某些网站可能不允许 HEAD 请求，某些应用程序服务器也可能不支持 GET。

必须至少指定 GET 或 HEAD

如果同时指定了 GET 和 HEAD，将先执行 HEAD，

然后执行 GET。

默认： `-request HEAD`

metric

要返回的逗号分隔的度量标准列表

- `av` = 可用性/可以连接（0=否，1=是）
- `rt` = 往返响应时间（用于连接）
- `tp` = 吞吐量（字节数。#返回的千字节数/往返响应）
- `all` = 所有统计信息

默认： `all`

示例：

```
perl epaplugins/solaris/httpSvcAvailability.pl -url  
https://secure.server.com,http:8080//IIS.server.com
```

Perl 支持问题

对于 *httpSvcAvailability.pl* 插件 (HTTP):

httpSvcAvailability.pl 插件 (HTTP) 要求安装 Perl 库 `libwww-perl` (v5.65+) 以执行基本的 `http` 服务检查。www.activestate.com 提供 Perl v5.6.1, 其中包括安装过程中需要用到的 `libwww-perl`。

对于 *httpSvcAvailability.pl* 插件 (HTTPS):

- 对于 Windows 上的 `https` 服务检查, 应安装 `Crypt-SSLeay` (v0.17.1+)。
- 对于 Solaris 上的 `https` 服务检查, 应安装 `Net-SSLeay` (v1.05+)。这两种软件均可使用 ActiveState 推出的 Perl 中随附的 `ppm Package Manager` 工具轻松完成安装。

kstat.pl

该插件根据 Solaris 推出的 `Kstat` 实用工具提供的内核统计信息来报告度量标准。

生成的度量标准

Solaris 内核统计信息 (`Kstat`) 度量标准中详细描述了 *kstat.pl* 插件生成的度量标准。

命令行参数

该插件具有四个命令行参数:

class

将与 `kstat` 类匹配的正则表达式

module

将与 `kstat` 模块匹配的正则表达式

name

将与 `kstat` 名称匹配的正则表达式

statistic

将与 `kstat` 统计信息匹配的正则表达式

注意：您可能想要使用这些参数来缩小结果范围。如果没有指定命令行切换，插件将返回类、模块、名称和统计信息的所有 Solaris 内核信息。

示例：

```
perl epaplugins/solaris/kstat.pl -class vm -module unix -name segmap -statistic  
'^free'
```

Introscope 度量标准将创建节点 VM，然后依次是资源 UNIX、segmap、"free" 以及值。

注意：如果 `kstat` 报告了资源名称空值，将不会生成度量标准。

Perl 支持问题

为获得最佳性能，建议使用 Solaris 提供的 Perl（通常位于 `/usr/bin/perl` 中，而 Perl 库则安装在 `/usr/perl5` 中）来运行 `kstat.pl` 插件，因为这样将支持通过随附的 Perl 库连接到 Solaris 内核子系统的直接接口。如果使用其他 Perl 安装来运行 `kstat.pl` 插件，将使用 `kstat.pl` 命令行实用工具，这样开销会稍有增加。

iisLogReader.pl

该插件根据 `iisLogReader.cfg` 文件中的配置来报告度量标准。

注意：使用该日志阅读器插件时，除非有新条目添加到日志文件中，否则 EPAgent 不会向企业管理器报告。

生成的度量标准

IIS 日志阅读器度量标准中详细描述了 `iisLogReader.pl` 插件生成的度量标准。

命令行参数

该插件具有四个可选命令行参数：

logfileDir

日志文件所在目录

sleepTime

两次检查日志文件中是否有新消息之间的延迟秒数

logfile

与生成的日志文件名称相对应的正则表达式

subResource

度量标准名称的子资源部分

示例：

```
perl iisLogReader.pl -sleepTime 30 -logfileDir "c:/Program Files/iis/Togs"
```

iisLogReader.cfg 配置文件

iisLogReader.cfg 文件包含 *iisLogReader.pl* 文件中命令行参数的值。

每个 Web 服务器应用程序日志文件解析器均自带配置文件，并为睡眠延迟、子资源名称、资源名称、日志文件目录以及日志文件名称模式设置了默认值。配置文件也包含用以匹配和报告的度量标准列表。每个度量标准配置均具有 **match**（正则表达式）、**metricType**、**metricName** 和 **metricValue** 参数。

LOGFILEDIR

完整的目录路径名，包含关联 Web 服务器的日志文件。

注意：正斜杠是 Windows 路径的有效路径分隔符（也可以使用反斜杠）。

默认

C:/WINNT/system32/LogFiles/W3SVC1

LOGFILE

要读取的日志文件的名称。如果该名称随时间推移而变化，应该是正则表达式（例如旋转日志文件）。**match** 不区分大小写。

注意：必须与整个文件名称（不含路径）相匹配。

注意：针对 W3C 扩展日志文件格式（*iisLogReader.pl* 插件的唯一支持格式）配置 IIS 时，默认设置为与日志文件名称相匹配。

默认

ex(tend)?\d+\.log

SLEEPTIME

两次检查日志文件是否有新消息之间的延迟秒数。

由于计数器度量标准的定义是“每个时间间隔”（参见度量标准表），因此 **SLEEPTIME** 应设置为等于或大于企业管理器时间片（即 15 秒），否则计数器度量标准将不正确。

默认

15

RESOURCE

Introscope 度量标准名称的资源部分。对应于 Workstation 调查器中“代理”节点下的第一个节点。

默认

IISWebServer

SUBRESOURCE

Introscope 度量标准名称的子资源部分。对应于上面指定的“资源”节点下的子节点。

默认

<空>

LINE_FORMAT

定义由 IIS Web 服务器记录的精确行格式。

LINE_FORMAT 子属性表中描述了行格式的子属性。

默认

n/a

LINE_FORMAT 子属性

下表描述了属性 LINE_FORMAT 及其所有子属性。

LINE_FORMAT 的下列属性必须与 IIS Web 服务器记录到日志文件的字段精确匹配。如果属性不匹配，iisLogReader.pl 插件将无法报告部分或全部度量标准。

IIS Web 服务器的精确字段在 IIS 管理工具（或 IIS 旧版本的网页）内配置。您可以在[在 IIS4.0 和 IIS5.0 中为 IIS Web 服务器日志配置日志字段](#) (p. 46)。

默认情况下，IIS Web 服务器仅记录以下字段：

- Time
- ClientIPAddress
- Method
- URISem
- ProtocolStatus

相应地，默认 iisLogReader.cfg 文件也仅启用这五个 LINE_FORMAT 子属性。

在不进行设置的情况下，iisLogReader.pl 插件需要使用默认的 IIS Web 服务器日志记录配置。插件可以报告所有度量标准，但不包括“传输的字节数”。“传输的字节数”要求 IIS Web 服务器记录“发送的字节数”数据字段，并且取消注释/启用 iisLogReader.cfg 子属性“发送的字节数”。如果用户无法在 IIS Web 服务器和 iisLogReader.cfg 中启用“发送的字节数”，则不会向 CA Introscope® 报告度量标准“传输的字节数”。

如果取消注释子属性行，则 iisLogReader.pl 插件预计会在 IIS Web 服务器日志文件（由 LOGFILEDIR 和 LOGFILE 属性指定）中找到该字段。反过来，如果为子属性行添加注释，则 iisLogReader.pl 插件不会在日志文件中找到该字段。子属性行已符合 IIS Web 服务器从左至右的数据记录顺序（如果在 IIS Web 服务器中启用了相应的字段）。不应更改该排序。

DELIMITER

连续的空格（空格、选项卡）

默认

\s+

FIELDS

n/a

默认

n/a

FIELDS->Date

活动发生的日期。YYYY-MM-DD。

默认

\d\d\d\d-\d\d-\d\d

FIELDS->Time

默认情况下处于启用状态。

活动发生的时间。24HH:MM:SS GMT。

默认

\d\d\d\d\d\d-\d\d-\d\d

FIELDS->ClientIPAddress

默认情况下处于启用状态。

访问您服务器的客户端的 IP 地址。四位数字，由句点分隔。

默认

\d+\.\d+\.\d+\.\d+

FIELDS->UserName

访问您服务器的用户的名称。-、反斜杠和字母数字字符的任意组合。

默认

`[-\\w]+`

FIELDS->ServiceName

客户端计算机上正在运行的 Internet 服务。字母数字字符的任意组合。

默认

`\w+`

FIELDS->ServerName

生成日志条目的服务器的名称。句点、连字符和字母数字字符的任意组合。

默认

`[-.\w]+`

FIELDS->ServerIPAddress

生成日志条目的服务器的 IP 地址。四位数字，由句点分隔。

默认

`\d+\.\d+\.\d+\.\d+`

FIELDS->ServerPort

客户端连接到的端口号。一个或多个数字。

默认

`\d+`

FIELDS->Method

默认情况下处于启用状态。

客户端试图执行的操作（例如，GET 命令）。

默认

GET|HEAD|PUT|POST|DELETE

FIELDS->URISem

默认情况下处于启用状态。

访问的资源；例如，HTML 页面、CGI 程序或脚本。一系列非空格字符。

默认

\S+

FIELDS->URIQuery

客户端试图执行的查询（如果有）；即客户端寻求匹配的一个或多个搜索字符串。一系列非空格字符。

默认

\S+

FIELDS->ProtocolStatus

默认情况下处于启用状态。

操作的状态，使用 HTTP 术语。

默认

\d+

FIELDS->Win32Status

操作的状态，使用 Windows 所用术语。一个或多个数字。

默认

\d+

FIELDS->BytesSent

服务器发送的字节数。一个或多个数字。

注意：默认情况下不在 IIS 中启用该字段。必须打开该字段才能收集“传输的字节数”度量标准。

默认

\d+

FIELDS->BytesReceived

服务器接收的字节数。一个或多个数字。

默认

\d+

FIELDS->TimeTaken

操作所用时间长度，以毫秒为单位。一个或多个数字。

默认

\d+

FIELDS->ProtocolVersion

客户端使用的协议（HTTP、FTP）版本。如果使用 HTTP，将使用 HTTP 1.0 或 HTTP 1.1。一系列字母数字字符、正斜杠和句点。

默认

[\w/.]+

FIELDS->Host

客户端主机。一系列字母数字字符、句点、冒号。

默认

[\w.:]+

FIELDS->UserAgent

客户端所用浏览器。一系列非空格字符。

默认

\S+

FIELDS->Cookie

发送或接收的 Cookie 内容（如果有）。一系列非空格字符。

默认

\S+

FIELDS->Referer

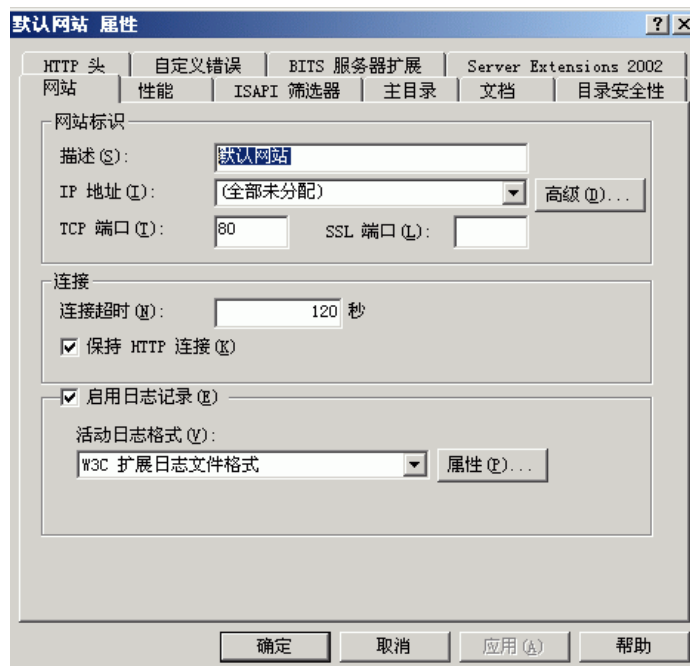
将用户引导到当前站点的站点。一系列非空格字符。

默认

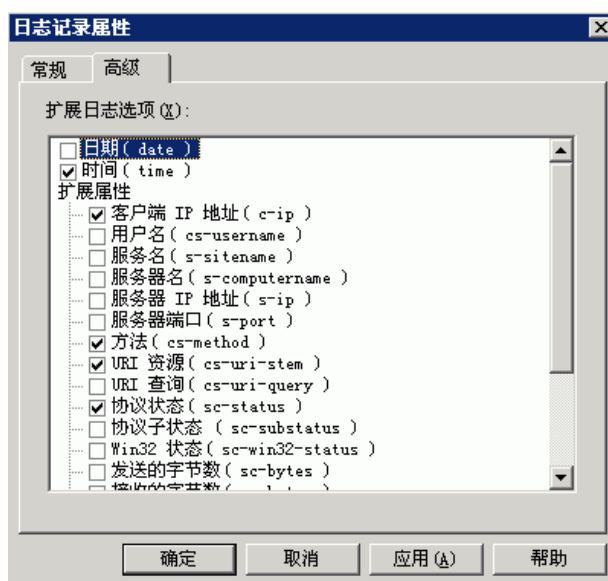
\S+

在 IIS4.0 和 IIS5.0 中配置 IIS Web 服务器日志记录

1. 启动 Internet 服务管理器。
2. 在左侧展开计算机的树节点。
此时应该显示三个节点：
 - FTP 站点
 - 网站
 - SMTP 虚拟服务器
3. 右键单击“网站”节点并选择“属性”。
此时将显示“属性”对话框。



4. 选择“网站”选项卡并确保：
 - 已选中“启用日志记录”复选框
 - 已在“活动日志格式”下拉框中选择“W3C 扩展日志文件格式”。
5. 单击“属性”。
此时将显示“扩展日志记录属性”对话框。



6. 在“常规属性”选项卡中，“日志文件目录”已指定且必须与 *iisLogReader.cfg* 文件中的 LOGFILEDIR 属性相匹配。
7. 在“扩展属性”选项卡中，以下字段是 *iisLogReader.pl* 插件报告其所有度量标准所必需的：
 - 时间
 - 客户端 IP 地址
 - 方法
 - URISTem
 - 协议状态
 - 发送的字节数（查看“传输的字节数”所必需的字段）

启用（选中）的所有字段必须与 *iisLogReader.cfg* 文件中 LINE_FORMAT->FIELDS 部分以下的已启用（取消注释）字段相匹配。*iisLogReader.cfg* 文件的此部分为 *iisLogReader.pl* 插件指示了在日志文件中将要使用的具体行格式。如果有不匹配的情况，*iisLogReader.pl* 插件将无法报告度量标准，或者会报告不正确的度量标准。

第 3 章：EAgent 默认插件度量标准

本章描述 EAgent 随附的默认插件生成的度量标准。

所有度量标准都是使用您的系统时钟设置的时间来计算。如果在事务期间重置了系统时钟，则针对该事务报告的用时可能会具有误导性。

此部分包含以下主题：

[无效字符替换 \(p. 49\)](#)

[IIS 日志阅读器度量标准 \(p. 49\)](#)

[HTTP/HTTPS 服务可用性度量标准 \(p. 54\)](#)

[Solaris 内核统计信息 \(Kstat\) 度量标准 \(p. 56\)](#)

[进程可用性度量标准 \(p. 56\)](#)

[磁盘统计信息 \(Windows\) 度量标准 \(p. 57\)](#)

[磁盘统计信息 \(Solaris\) 度量标准 \(p. 61\)](#)

[Apache 日志度量标准 \(p. 66\)](#)

[WebSphere 日志度量标准 \(p. 70\)](#)

[WebLogic 日志度量标准 \(p. 77\)](#)

无效字符替换

所有资源、子资源和度量标准名称中都会被检查是否存在 Introscope 度量标准名称中不允许使用的无效字符。注意，管道字符 (|) 和冒号 (:) 为保留字符。任何包含这些字符的用户或系统生成的名称都会自动替换这些字符：

原始字符	替换为	说明
	_	管道字符替换为下划线
:	^	冒号替换为插入符号

IIS 日志阅读器度量标准

下节介绍了 *iisLogReader.pl* 插件生成的度量标准。这些度量标准的计数基于时间间隔，自插件最后一次检查日志文件开始。还介绍了 IISWebServer <Sub-Resource> 度量标准。

网页点击总数

CA Introscope® 数据类型

IntCounter

单位

1（点击）

默认

开

访问者总数

CA Introscope® 数据类型

IntCounter

单位

1（点击）

默认

开

<URL> 的网页点击数

CA Introscope® 数据类型

IntCounter

单位

1（点击）

默认

开

传输的字节数

CA Introscope® 数据类型

IntCounter

单位

1 (MB)

默认

开

ErrorCount_400_BadRequest

CA Introscope® 数据类型

IntCounter

单位

1 (错误)

默认

开

ErrorCount_401_Unauthorized

CA Introscope® 数据类型

IntCounter

单位

1 (错误)

默认

开

ErrorCount_403_Forbidden

CA Introscope® 数据类型

IntCounter

单位

1（错误）

默认

开

ErrorCount_404_ServerConfiguration

CA Introscope® 数据类型

IntCounter

单位

1（错误）

默认

开

ErrorCount_500_InternalServerError

CA Introscope® 数据类型

IntCounter

单位

1（错误）

默认

开

ErrorMessage_400_BadRequest

CA Introscope® 数据类型

StringEvent

单位

N/A

默认

关闭

ErrorMessage_401_Unauthorized

CA Introscope® 数据类型

StringEvent

单位

N/A

默认

关闭

ErrorMessage_403_Forbidden

CA Introscope® 数据类型

StringEvent

单位

N/A

默认

关闭

ErrorMessage_404_PageNotFound

CA Introscope® 数据类型

StringEvent

单位

N/A

默认

关闭

ErrorMessage_500_InternalServerError

CA Introscope® 数据类型

StringEvent

单位

N/A

默认

关闭

HTTP/HTTPS 服务可用性度量标准

下节介绍了 *httpSvcAvailability.pl* 插件生成的度量标准，还介绍了远程计算机 `<hostnameOrIPAddr> | Service | HTTP<Sub-Resource>` 度量标准。

可用

CA Introscope® 数据类型

IntCounter

单位

0-1

默认

开

响应时间

CA Introscope® 数据类型

IntCounter

单位

秒

默认

开

平均吞吐量

CA Introscope® 数据类型

IntCounter

单位

KB/msec

默认

开

Solaris 内核统计信息 (Kstat) 度量标准

以下度量标准由 *kstat.pl* 插件生成。

kstat 插件生成的特定度量标准的名称根据情况会有所不同，具体取决于 Solaris 系统、Solaris 命名概念，以及环境性能代理对 *kstat.pl* 插件的命令行调用（在 *IntroscopeEPAgent.properties* 中进行配置）。

Solaris 内核 <class>|<module>|<name> 度量标准

下节介绍 Solaris 内核 <class>|<module>|<name> 度量标准：

<stat>

CA Introscope® 数据类型

IntCounter

单位

可变

默认

开

进程可用性度量标准

这些是 *processAvailability.pl* 插件生成的度量标准。

下节介绍了 RunningProcesses <Sub-Resource> 度量标准。

计数

CA Introscope® 数据类型

IntCounter

单位

1 (进程)

默认

开

磁盘统计信息 (Windows) 度量标准

这些是 <EPA_Home>/windows/diskStats.pl 插件生成的度量标准。

以下各节介绍 RunningProcesses <Sub-Resource> 度量标准。

详细信息:

[磁盘读取数/秒](#) (p. 58)

[磁盘写入数/秒](#) (p. 58)

[磁盘读取字节数/秒](#) (p. 58)

[磁盘写入字节数/秒](#) (p. 59)

[当前磁盘队列长度](#) (p. 59)

[平均磁盘读取时间 \(毫秒\)](#) (p. 59)

[平均磁盘写入时间 \(毫秒\)](#) (p. 60)

[平均磁盘字节数/读取](#) (p. 60)

[平均磁盘字节数/写入](#) (p. 60)

[可用磁盘空间 \(MB\)](#) (p. 61)

[可用磁盘空间 \(%\)](#) (p. 61)

磁盘读取数/秒

CA Introscope® 数据类型

IntCounter

单位

1 (计数)

默认

开

磁盘写入数/秒

CA Introscope® 数据类型

IntCounter

单位

1 (计数)

默认

开

磁盘读取字节数/秒

CA Introscope® 数据类型

IntCounter

单位

1 个字节

默认

开

磁盘写入字节数/秒

CA Introscope® 数据类型

IntCounter

单位

1 个字节

默认

开

当前磁盘队列长度

CA Introscope® 数据类型

IntCounter

单位

1 (请求)

默认

开

平均磁盘读取时间 (毫秒)

CA Introscope® 数据类型

IntCounter

单位

1 毫秒

默认

开

平均磁盘写入时间（毫秒）

CA Introscope® 数据类型

IntCounter

单位

1 毫秒

默认

开

平均磁盘字节数/读取

CA Introscope® 数据类型

IntCounter

单位

1 个字节

默认

开

平均磁盘字节数/写入

CA Introscope® 数据类型

IntCounter

单位

1 个字节

默认

开

可用磁盘空间 (MB)

CA Introscope® 数据类型

IntCounter

单位

1 MB

默认

开

可用磁盘空间 (%)

CA Introscope® 数据类型

IntCounter

单位

1 (进程)

默认

开

磁盘统计信息 (Solaris) 度量标准

该表显示 `<EPAgent_Home>/solaris/diskStats.pl` 插件生成的度量标准。

以下各节介绍磁盘 `<Sub-Resource>` 度量标准。

详细信息:

[磁盘时间 \(%\)](#) (p. 62)

[事务等待时间 \(%\)](#) (p. 62)

[平均活动事务数](#) (p. 63)

[平均等待事务数](#) (p. 63)

[磁盘读取数/秒](#) (p. 63)

[磁盘写入数/秒](#) (p. 64)

[每秒读取 KB](#) (p. 64)

[每秒写入 KB](#) (p. 64)

[等待队列的平均服务时间 \(毫秒\)](#) (p. 65)

[活动事务的平均服务时间 \(毫秒\)](#) (p. 65)

[可用磁盘空间 \(MB\)](#) (p. 65)

[可用磁盘空间 \(%\)](#) (p. 66)

磁盘时间 (%)

CA Introscope® 数据类型

IntCounter

单位

1%

默认

开

事务等待时间 (%)

CA Introscope® 数据类型

IntCounter

单位

1%

默认

开

平均活动事务数

CA Introscope® 数据类型

IntCounter

单位

1 个事务

默认

开

平均等待事务数

CA Introscope® 数据类型

IntCounter

单位

1 个事务

默认

开

磁盘读取数/秒

CA Introscope® 数据类型

IntCounter

单位

1 (计数)

默认

开

磁盘写入数/秒

CA Introscope® 数据类型

IntCounter

单位

1 (计数)

默认

开

每秒读取 KB

CA Introscope® 数据类型

IntCounter

单位

1 个字节

默认

开

每秒写入 KB

CA Introscope® 数据类型

IntCounter

单位

1 个字节

默认

开

等待队列的平均服务时间 (毫秒)

CA Introscope® 数据类型

IntCounter

单位

1 毫秒

默认

开

活动事务的平均服务时间 (毫秒)

CA Introscope® 数据类型

IntCounter

单位

1 毫秒

默认

开

可用磁盘空间 (MB)

CA Introscope® 数据类型

IntCounter

单位

1 MB

默认

开

可用磁盘空间 (%)

CA Introscope® 数据类型

IntCounter

单位

1%

默认

开

Apache 日志度量标准

该表显示 *apacheLogReader.pl* 插件生成的度量标准。

以下各节介绍 ApacheWebServer <Sub-Resource> 度量标准。

详细信息:

[PageHitCount](#) (p. 67)

[ErrorCount 400 BadRequest](#) (p. 67)

[ErrorCount 401 Unauthorized](#) (p. 67)

[ErrorCount 403 Forbidden](#) (p. 68)

[ErrorCount 404 PageNotFound](#) (p. 68)

[ErrorCount 500 ServerConfiguration](#) (p. 68)

[ErrorMessage 400 BadRequest](#) (p. 69)

[ErrorMessage 401 Unauthorized](#) (p. 69)

[ErrorMessage 403 Forbidden](#) (p. 69)

[ErrorMessage 404 PageNotFound](#) (p. 70)

[ErrorMessage 500 ServerConfiguration](#) (p. 70)

PageHitCount

CA Introscope® 数据类型

IntCounter

单位

1（点击）

默认

开

ErrorCount_400_BadRequest

CA Introscope® 数据类型

IntCounter

单位

1（错误）

默认

开

ErrorCount_401_Unauthorized

CA Introscope® 数据类型

IntCounter

单位

1（错误）

默认

开

ErrorCount_403_Forbidden

CA Introscope® 数据类型

IntCounter

单位

1（错误）

默认

开

ErrorCount_404_PageNotFound

CA Introscope® 数据类型

IntCounter

单位

1（错误）

默认

开

ErrorCount_500_ServerConfiguration

CA Introscope® 数据类型

IntCounter

单位

1（错误）

默认

开

ErrorMessage_400_BadRequest

CA Introscope® 数据类型

StringEvent

单位

1 (错误)

默认

开

ErrorMessage_401_Unauthorized

CA Introscope® 数据类型

StringEvent

单位

1 (错误)

默认

开

ErrorMessage_403_Forbidden

CA Introscope® 数据类型

StringEvent

单位

1 (错误)

默认

开

ErrorMessage_404_PageNotFound

CA Introscope® 数据类型

StringEvent

单位

1 (错误)

默认

开

ErrorMessage_500_ServerConfiguration

CA Introscope® 数据类型

StringEvent

单位

1 (错误)

默认

开

WebSphere 日志度量标准

这些是 *websphereLogReader.pl* 插件生成的度量标准。

以下各节介绍 WebSphereApplicationServer <Sub-Resource> 度量标准。

详细信息:

[MessageCount_Audit](#) (p. 71)

[MessageCount_Warning](#) (p. 71)

[MessageCount_EventDebug](#) (p. 72)

[MessageCount_Error](#) (p. 72)

[MessageCount_Fatal](#) (p. 72)

[MessageText_Audit](#) (p. 73)

[MessageText_Warning](#) (p. 73)

[MessageText_EventDebug](#) (p. 73)

[MessageText_Error](#) (p. 74)

[MessageText_Fatal](#) (p. 74)

MessageCount_Audit

CA Introscope® 数据类型

IntCounter

单位

1 (点击)

默认

开

MessageCount_Warning

CA Introscope® 数据类型

IntCounter

单位

1 (点击)

默认

开

MessageCount_EventDebug

CA Introscope® 数据类型

IntCounter

单位

1（点击）

默认

开

MessageCount_Error

CA Introscope® 数据类型

IntCounter

单位

1（点击）

默认

开

MessageCount_Fatal

CA Introscope® 数据类型

IntCounter

单位

1（点击）

默认

开

MessageText_Audit

CA Introscope® 数据类型

IntCounter

单位

1 (点击)

默认

开

MessageText_Warning

CA Introscope® 数据类型

IntCounter

单位

1 (点击)

默认

开

MessageText_EventDebug

CA Introscope® 数据类型

IntCounter

单位

1 (点击)

默认

开

MessageText_Error

CA Introscope® 数据类型

IntCounter

单位

1（点击）

默认

开

MessageText_Fatal

CA Introscope® 数据类型

IntCounter

单位

1（点击）

默认

开

CA Introscope® 数据类型

StringEvent

单位

n/a

默认

开

CA Introscope® 数据类型

IntCounter

单位

1（消息）

默认

开

CA Introscope® 数据类型

IntCounter

单位

1 (消息)

默认

开

CA Introscope® 数据类型

IntCounter

单位

1 (消息)

默认

开

CA Introscope® 数据类型

IntCounter

单位

1 (消息)

默认

开

CA Introscope® 数据类型

IntCounter

单位

1 (消息)

默认

开

CA Introscope® 数据类型

StringEvent

单位

n/a

默认

关闭

CA Introscope® 数据类型

StringEvent

单位

n/a

默认

关闭

CA Introscope® 数据类型

StringEvent

单位

n/a

默认

关闭

CA Introscope® 数据类型

StringEvent

单位

n/a

默认

关闭

CA Introscope® 数据类型

StringEvent

单位

n/a

默认

关闭

WebLogic 日志度量标准

该表显示 *weblogicLogReader.pl* 插件生成的度量标准。

以下各节介绍 WebLogicApplicationServer <Sub-Resource> 度量标准。

详细信息:

[服务器版本](#) (p. 78)

[MessageCount Warning](#) (p. 78)

[MessageCount Error](#) (p. 78)

[MessageCount Notice](#) (p. 79)

[MessageCount Critical](#) (p. 79)

[MessageCount Alert](#) (p. 79)

[MessageCount Emergency](#) (p. 80)

[MessageText Warning](#) (p. 80)

[MessageText Error](#) (p. 80)

[MessageText Notice](#) (p. 81)

[MessageText Critical](#) (p. 81)

[MessageText Alert](#) (p. 81)

[MessageText Emergency](#) (p. 82)

服务器版本

CA Introscope® 数据类型

StringEvent

单位

n/a

默认

开

MessageCount_Warning

CA Introscope® 数据类型

IntCounter

单位

1（消息）

默认

开

MessageCount_Error

CA Introscope® 数据类型

IntCounter

单位

1（消息）

默认

开

MessageCount_Notice

CA Introscope® 数据类型

IntCounter

单位

1 (消息)

默认

开

MessageCount_Critical

CA Introscope® 数据类型

IntCounter

单位

1 (消息)

默认

开

MessageCount_Alert

CA Introscope® 数据类型

IntCounter

单位

1 (消息)

默认

开

MessageCount_Emergency

CA Introscope® 数据类型

IntCounter

单位

1 (消息)

默认

开

MessageText_Warning

CA Introscope® 数据类型

StringEvent

单位

n/a

默认

关闭

MessageText_Error

CA Introscope® 数据类型

StringEvent

单位

n/a

默认

关闭

MessageText_Notice

CA Introscope® 数据类型

StringEvent

单位

n/a

默认

关闭

MessageText_Critical

CA Introscope® 数据类型

StringEvent

单位

n/a

默认

关闭

MessageText_Alert

CA Introscope® 数据类型

StringEvent

单位

n/a

默认

关闭

MessageText_Emergency

CA Introscope® 数据类型

StringEvent

单位

n/a

默认

关闭

第 4 章：EPAgent 属性

本附录详细介绍了用来自定义环境性能代理的属性文件和设置。您可以直接阅读文件以获取其他信息（位于 <EPAgent 主目录>/epagent/introscopeEPAgent.properties）。

注意：在 Windows 计算机上的这些文件中添加路径时，反斜杠 (\) 必须用另一个反斜杠转义（每次使用双反斜杠），例如

C:\\Introscope\\lib\\Agent.jar。

此部分包含以下主题：

[IntroscopeEPAgent.properties 文件](#) (p. 83)

[代理度量标准老化](#) (p. 84)

[区分大小写的度量标准](#) (p. 88)

[ChangeDetector 配置](#) (p. 89)

[企业管理器连接](#) (p. 92)

[EPAgent 配置](#) (p. 92)

[扩展目录](#) (p. 93)

[日志记录](#) (p. 94)

[流程](#) (p. 95)

[远程配置](#) (p. 96)

[有状态插件](#) (p. 96)

[无状态插件](#) (p. 98)

IntroscopeEPAgent.properties 文件

EPAgent 设置的类型与 CA Introscope® 代理配置文件中的设置类型相同。

代理和进程名称设置

introscope.agent.agentName 和 *introscope.agent.customProcessName* 这两个属性定义 EPAgent 的名称和进程的名称，二者均用作完全符合条件的代理名称（域 / 主机 / 进程 / 代理名称）的一部分，目的是跟踪多个代理会话的单个 EPAgent 的信息。

这些属性都是可选的。如果代理名称尚未定义，则称为 UnknownAgent。

进程名

CA Introscope® 根据这些条件（优先顺序同显示顺序）来命名 EPAgent。

- 由属性 `introscope.agent.customProcessName` 定义的名称。
- 从应用程序主类的名称推断出的名称。
- 由 `introscope.agent.defaultProcessName` 设置的名称。

详细信息：

[流程](#) (p. 95)

代理名称

`introscope.agent.agentName` 属性定义 EPAgent 的名称。

CA Introscope® 根据这些条件（优先顺序同显示顺序）来命名环境性能代理。

- 由属性 [introscope.agent.agentName](#) (p. 96) 定义的名称。
- 如果 `introscope.agent.agentName` 中未定义唯一的名称，则会将环境性能代理命名为 `UnknownAgent`。

代理度量标准老化

代理度量标准老化功能可定期从代理内存缓存中删除死度量标准。死度量标准是指在所配置的一段时间内没有报告新数据的度量标准。删除旧的度量标准有助于提高代理性能，避免潜在的度量标准爆发。

注意：如果无意中将代理设置为报告的度量标准多于系统可以处理的度量标准，将发生度量标准爆发。如果报告了太多度量标准，代理会影响应用程序服务器的性能，在极个别情况下，还会使服务器完全无法运行。

仅当组中的所有度量标准均被视为删除候选项时，才会删除组中的度量标准。目前，只有 `BlamePointTracer` 和 `MetricRecordingAdministrator` 度量标准可按组删除。其他度量标准均单独删除。

MetricRecordingAdministrator 具有以下用于创建、检索或删除度量标准组的接口：

- *getAgent().IAgent_getMetricRecordingAdministrator.addMetricGroup*
字符串组件，收集度量标准。组件名称是度量标准组的度量标准资源名称。度量标准必须位于同一个度量标准节点下，才能称之为一个组。度量标准是 *com.wily.introscope.spec.metric.AgentMetric* 数据结构的集合。您只能将 *AgentMetric* 数据结构添加到此集合中。
- *getAgent().IAgent_getMetricRecordingAdministrator.getMetricGroup*
字符串组件。根据组件名称（即度量标准资源名称），您可以获得度量标准集合。
- *getAgent().IAgent_getMetricRecordingAdministrator.removeMetricGroup*
字符串组件。度量标准组基于组建名称（即度量标准资源名称）进行删除。
- *getAgent().IAgent_getDataAccumulatorFactory.isRemoved*
检查度量标准是否已删除。如果在扩展中保留一个累积器实例，则使用此接口。如果累积器由于度量标准老化而被删除，可使用此接口防止保留死引用。

重要信息！ 如果要创建使用 *MetricRecordingAdministrator* 界面的扩展（例如，用于与其他 CA Technologies 产品配合使用），请确保删除自己的累积器实例。如果度量标准因未被调用而过期，并且稍后数据可用于该度量标准，则旧的累积器实例将不会创建新的度量标准数据点。要避免这种情况，请不要删除您自己的累积器实例，并改用 *getDataAccumulatorFactory* 接口。

配置代理度量标准老化

默认情况下，代理度量标准老化处于打开状态。您可以选择使用属性 *introscope.agent.metricAging.turnOn* (see page 86) 关闭此功能。如果从 *IntroscopeAgent.profile* 中删除此属性，默认情况下将关闭代理度量标准老化。

代理度量标准老化将针对代理中的心跳运行。心跳使用属性 *introscope.agent.metricAging.heartbeatInterval* (see page 87) 进行配置。请务必将心跳的频率维持在较低状态。较高的心跳频率将影响代理和 CA Introscope® 的性能。

在每个心跳期间，会检查一组特定的度量标准。这可通过属性 `introscope.agent.metricAging.dataChunk` (see page 87) 进行配置。将此值维持在较低状态也很重要，因为较高的值将影响性能。默认值是每个心跳检查 500 个度量标准。将逐个检查这 500 个度量标准以查看是否是删除候选项。例如，如果将此属性设置为每个心跳检查 500 个度量标准区块，并且代理内存中共有 10,000 个度量标准，则它将在对性能造成较小影响的前提下花费更长时间来检查所有 10,000 个度量标准。但是，如果将此属性设置为较高的值，您可以更快地检查这 10,000 个度量标准，但是开销可能会很大。

如果度量标准在特定时间段之后没有收到新数据，该度量标准将成为删除候选项。您可以使用属性 `introscope.agent.metricAging.numberTimeslices` (see page 87) 来配置此时间段。默认情况下，此属性设置为 3000。如果某个度量标准满足删除以上条件，将执行检查以查看其所在组中的所有度量标准是否是删除候选项。如果也满足此项要求，则删除该度量标准。

introscope.agent.metricAging.turnOn

打开或关闭代理度量标准老化。

属性设置

True 或 False

默认

True

示例

```
introscope.agent.metricAging.turnOn=true
```

注释

对此属性所做的更改可立即生效，不需要重新启动托管应用程序。

introscope.agent.metricAging.heartbeatInterval

指定检查待删除度量标准的时间间隔（以秒为单位）。

默认

1800

示例

```
introscope.agent.metricAging.heartbeatInterval=1800
```

注释

您必须重新启动托管应用程序，对此属性所做的更改才能生效。

introscope.agent.metricAging.dataChunk

指定在每个时间间隔内检查的度量标准数。

默认

500

示例

```
introscope.agent.metricAging.dataChunk=500
```

注释

对此属性所做的更改可立即生效，不需要重新启动托管应用程序。

introscope.agent.metricAging.numberTimeslices

在成为删除候选项之前，此属性会指定要检查的时间间隔数（没有任何新数据）。

默认

3000

示例

```
introscope.agent.metricAging.numberTimeslices=3000
```

注释

对此属性所做的更改会立即生效，不需要重新启动托管应用程序。

introscope.agent.metricAging.metricExclude.ignore.0

使指定的度量标准不被删除。要防止一个或多个度量标准老化，请将度量标准名称或度量标准筛选添加到列表中。

属性设置

以逗号分隔的度量标准列表。可以在度量标准名称中使用星号 (*) 作为通配符。

默认

默认情况下，度量标准名称以 **Threads** 开头 (*Threads**)。

示例

```
introscope.agent.metricAging.metricExclude.ignore.0=Threads*
```

注释

对此属性所做的更改可立即生效，不需要重新启动托管应用程序。

区分大小写的度量标准

以下属性可对区分大小写进行配置。

introscope.epagent.metricscasesensitive

使度量标准名称区分大小写。

属性设置

True 或 False

默认

true

示例

```
introscope.epagent.metricscasesensitive=true
```

注释

- 默认情况下注释掉。
- 设置为 **true**，此属性使度量标准区分大小写。
- 设置为 **false**，该属性关闭区分大小写的功能。

ChangeDetector 配置

以下属性使用 ChangeDetector 配置 EPAgent 交互。

introscope.changeDetector.enable

指定是启用还是禁用 ChangeDetector。将该属性设置为 true 可启用 ChangeDetector。默认情况下该属性会注释掉，并设置为 false。如果启用 ChangeDetector，还应设置与 ChangeDetector 相关的其他属性。

属性设置

True 或 False

默认

False

示例

```
introscope.changeDetector.enable=false
```

注释

您必须重新启动托管应用程序，对此属性所做的更改才能生效。

introscope.changeDetector.rootDir

为 ChangeDetector 文件指定根目录。根目录是 ChangeDetector 在其中创建本地缓存文件的文件夹。

属性设置

文本字符串形式的 ChangeDetector 文件根目录完整路径。

默认

默认路径为 `c://sw//AppServer//wily//change_detector`。

示例

```
introscope.changeDetector.rootDir=c://sw//AppServer//wily//change_detector
```

注释

请使用反斜线转义反斜线字符，如示例中所示。

introscope.changeDetector.isengardStartupWaitTimeInSec

指定启动代理后等待 ChangeDetector 尝试连接到企业管理器的时间。默认情况下，此属性注释掉。

默认

默认值为 15 秒。

示例

```
introscope.changeDetector.isengardStartupWaitTimeInSec=15
```

introscope.changeDetector.waitTimeBetweenReconnectInSec

指定 ChangeDetector 在重新尝试连接到企业管理器之前等待的秒数。默认情况下，此属性注释掉。

默认

默认值为 10 秒。

示例

```
introscope.changeDetector.waitTimeBetweenReconnectInSec=10
```

introscope.changeDetector.enableEPA

启用 ChangeDetector EPAgent 后，可将 EPAgent 插件用作 XML 格式更改数据的数据源。

属性设置

True 或 False

默认

true

示例

```
introscope.changeDetector.enableEPA=true
```

注释

默认情况下注释掉。

introscope.changeDetector.agentID

指定 ChangeDetector 用来标识 EPAgent 的文本字符串。

示例

```
introscope.changeDetector.agentID=SampleApplicationName
```

注释

默认情况下注释掉。

introscope.changeDetector.profile

指定 ChangeDetector 数据源配置文件的绝对路径或相对路径。默认情况下，此属性注释掉。

默认

默认值为 ChangeDetector-config.xml。

示例

```
introscope.changeDetector.profile=CDConfig\\changedetector-config.xml
```

注释

请使用反斜线转义反斜线字符，如示例中所示。

introscope.changeDetector.profileDir

指定包含数据源配置文件的目录的绝对路径或相对路径。如果设置了此属性，除了使用 *introscope.changeDetector.profile* 属性指定的任何文件以外，还会使用此目录中的所有数据源配置文件。默认情况下，此属性注释掉。

默认

默认值为 changeDetector_profiles。

示例

```
introscope.changeDetector.profileDir=c:\\CDconfig\\changeDetector_profiles
```

注释

使用反斜杠来转义反斜杠字符。

企业管理器连接

以下属性用于配置企业管理器的连接。

introscope.agent.enterprisemanager.connectionorder

指定在与主企业管理器断开连接的情况下，EPAgent 使用的备份企业管理器的连接顺序。列表中的项必须以逗号分隔。

默认

DEFAULT

示例

```
introscope.agent.enterprisemanager.connectionorder=DEFAULT
```

introscope.agent.enterprisemanager.transport.tcp.host.DEFAULT

指定运行代理默认与之连接的企业管理员的计算机主机名。

默认

localhost

示例

```
introscope.agent.enterprisemanager.transport.tcp.host.DEFAULT=localhost
```

注释

您必须重新启动托管应用程序，对此属性所做的更改才能生效。

EPAgent 配置

以下属性用于配置 EPAgent 本身。

introscope.epagent.config.networkDataPort

指定 EPAgent 侦听网络连接以获取传入 XML 数据的端口。

默认

8000

示例

```
introscope.epagent.config.networkDataPort=8000
```

注释

默认情况下，此属性注释掉。

introscope.epagent.config.httpServerPort

指定 EPAgent 侦听 HTTP GET 命令的端口。

默认

8080

示例

```
introscope.epagent.config.httpServerPort=8080
```

注释

默认情况下，此属性注释掉。

扩展目录

以下度量标准指定了扩展目录。

introscope.agent.extensions.directory

指定代理加载的所有扩展的位置。您可以指定该目录的绝对路径或相对路径。如果您不指定绝对路径，则您指定的值将相对于 *IntroscopeAgent.properties* 文件位置进行解析。

默认

默认位置为 `<EPAgent_Home>/epagent/ext` 目录中的 `ext` 目录。

示例

```
introscope.agent.extensions.directory=./ext
```

注释

您必须重新启动托管应用程序，对此属性所做的更改才能生效。

日志记录

以下属性用于配置日志记录。

log4j.logger.EPAgent

指定 EPAgent 日志的日志记录详细信息量和日志文件的输出位置。

默认

INFO, console

示例

```
log4j.logger.EPAgent=INFO, console
```

注释

- 将 *INFO* 替换为 *VERBOSE#com.wily.util.feedback.Log4JSeverityLevel* 可提高日志中详细信息的程度。
- 将 *console* 替换为 *logfile* 会将输出发送至日志文件而非控制台。

log4j.appender.logfile.File

此属性用于控制代理日志记录的日志记录级别和输出位置。

默认

IntroscopeEPA.log

示例

```
log4j.appender.logfile.File=IntroscopeEPA.log
```

注释

可使用日志文件位置和名称的完整路径名称。

流程

以下属性用于配置进程。

introscope.agent.customProcessName

指定受监控进程的名称。

默认

EPAgentProcess

示例

```
introscope.agent.customProcessName=EPAgentProcess
```

注释

对于应用程序服务器特定的安装程序，将其设置为应用程序服务器的名称。

introscope.agent.defaultProcessName

用于在未设置 *introscope.Agent.customProcessName* 且无法从应用程序主类推断出名称的情况下，为代理指定默认名称。

默认

UnknownProcess

示例

```
introscope.agent.defaultProcessName=UnknownProcess
```

注释

默认值可能随安装程序的不同而不同。

introscope.agent.agentName

指定 EPAgent 在企业管理器和 Workstation 中的显示名称。

默认

EPAgent

示例

```
introscope.agent.agentName=EPAgent
```

注释

对于应用程序特定的安装程序，将其设置为 UnknownAgent 或应用程序服务器的名称，与 <appservername> 代理中一样。

远程配置

以下属性可用于远程配置 EPAgent。

introscope.agent.remoteagentconfiguration.enabled

启用或禁用 EPAgent 的远程配置。

属性设置

True 或 False

默认

False

示例

```
introscope.agent.remoteagentconfiguration.enabled=false
```

注释

对此属性所做的更改可立即生效，不需要重新启动托管应用程序。

有状态插件

以下属性用于配置有状态插件。

introscope.epagent.plugins.stateful.names

指定要运行的有状态插件的名称。此属性用于启用 EPAgent 插件。

属性设置

以逗号分隔的插件名称列表。

默认

APACHE,WEBLOGIC,WEBSPPHERE,IIS

示例

```
introscope.epagent.plugins.stateful.names=APACHE,WEBLOGIC,WEBSPPHERE,IIS
```

注释

- 默认情况下，此属性注释掉。
- 这些名称可以由用户定义，只需与以下 *introscope.epagent.plugins.stateful.<pluginname>.command* 属性中所用名称匹配即可。

introscope.epagent.stateful.<pluginname>.command

指定命令来运行 *introscope.epagent.plugins.stateful.names* 属性中指定的插件；每个指定的插件名称均需使用一个属性行。

示例

```
introscope.epagent.stateful.APACHE.command=perl apacheLogReader.pl  
introscope.epagent.stateful.WEBLOGIC.command=perl webLogicLogReader.pl
```

注释

默认情况下，此属性注释掉。

introscope.epagent.stateful.<pluginname>.class

插件可以是“无状态”插件，也可以是“有状态”插件。无状态插件由 EPAgent 根据类似 cron 的日程表或延迟定期执行。有状态插件只由 EPAgent 启动一次，便会永久运行。如果有状态插件退出，系统将重新启动它。

插件可以是使用“command”子项指定的外部进程，也可以是在 EPAgent 虚拟机中执行的 java 类。Java 类插件使用“class”子项以及其后跟任何可选参数的完全限定类名称来指定。例如：*my.package.path.Class arg1 arg2 arg3*。

示例

```
introscope.epagent.stateful.mypugin.class=my.package.path.Class arg1 arg2 arg3
```

注释

默认情况下，此属性注释掉。

无状态插件

以下属性用于配置无状态插件。

introscope.epagent.plugins.stateless.names

指定要运行的无状态插件的名称。

默认

DISK,ORACLE,HTTP,KSTAT

示例

```
introscope.epagent.plugins.stateless.names=DISK,ORACLE,HTTP,KSTAT
```

注释

- 默认情况下，此属性注释掉。
- 这些名称可以由用户定义，只需与以下 *introscope.epagent.plugins.stateless.<pluginname>.command* 属性中所用名称匹配即可。

introscope.epagent.stateless.<pluginname>.command

指定命令来运行 *introscope.epagent.plugins.stateless.names* 属性中指定的插件；每个指定的插件名称均需使用一个属性行。

示例

```
introscope.epagent.stateless.DISK.command=perl diskStats.pl
```

注释

默认情况下，此属性注释掉。

introscope.epagent.stateless.<pluginname>.class

指定要在 EAgent VM 中作为无状态插件来运行的类。

注释

默认情况下，此属性注释掉。

introscope.epagent.stateless.<pluginname>.delayInSeconds

指定 EAgent 运行指定插件的频率（秒）；每个指定的插件名称均需使用一个属性行。

默认

900

示例

```
introscope.epagent.stateless.DISK.delayInSeconds=900
```

注释

默认情况下，此属性注释掉。

introscope.epagent.stateless.<pluginname>.schedule

指定 *introscope.epagent.stateless.<pluginname>.command* 属性中指定的插件的运行日程表。输入采用 CronTab 日程表语法（或者 UNIX 系统上的 "man 5 crontab"）。

注释

- 默认情况下，此属性注释掉。
- 使用该属性，而不使用 *introscope.epagent.stateless.<pluginname>.delayInSeconds* 属性。如果两个属性均取消注释，EPAgent 将退出。

introscope.epagent.config.stalledStatelessPluginTimeoutInSeconds

指定删除停顿的无状态插件之前等待的时间。

默认

60 秒

示例

```
#introscope.epagent.config.stalledStatelessPluginTimeoutInSeconds=60
```

注释

默认情况下，此属性注释掉。

introscope.epagent.stateless.<pluginname>.metricNotReportedAction

此属性用于确定无状态插件停止报告值时执行的操作。

属性设置

此属性的有效值有：

- **same**—持续报告同一个值，直到度量标准过期为止。
- **zero**—报告值为 0 的度量标准（如果度量标准的类型为字符串事件或时间戳，则执行“same”操作）。
- **stop**—停止报告度量标准值

默认

same

示例

下面的 MATCHEDPROCS 示例使用 processAvailability 插件为每个正在运行的进程返回值 1，这些进程的名称与由匹配参数指定的正则表达式匹配（例如，在此例中名称以“f”开头的进程）。当进程不再运行时，*metricNotReportedAction* 参数值“zero”可用于将度量标准值重置为 0。

```
introscope.epagent.stateless.MATCHEDPROCS.command=perl processAvailability.pl
-match ^f -distinctmatch
introscope.epagent.stateless.MATCHEDPROCS.delayInSeconds=900
introscope.epagent.stateless.MATCHEDPROCS.metricNotReportedAction=zero
```

