

# CA Application Performance Management

Environment Performance Agent 구현  
안내서  
릴리스 9.5



포함된 도움말 시스템 및 전자적으로 배포된 매체를 포함하는 이 문서(이하 "문서")는 정보 제공의 목적으로만 제공되며 CA에 의해 언제든 변경 또는 취소될 수 있습니다.

CA의 사전 서면 동의 없이 본건 문서의 전체 혹은 일부를 복사, 전송, 재생, 공개, 수정 또는 복제할 수 없습니다. 이 문서는 CA의 기밀 및 독점 정보이며, 귀하는 이 문서를 공개하거나 다음에 의해 허용된 경우를 제외한 다른 용도로 사용할 수 없습니다: (i) 귀하가 이 문서와 관련된 CA 소프트웨어를 사용함에 있어 귀하와 CA 사이에 별도 동의가 있는 경우, 또는 (ii) 귀하와 CA 사이에 별도 기밀 유지 동의가 있는 경우.

상기 사항에도 불구하고, 본건 문서에 기술된 라이선스가 있는 사용자는 귀하 및 귀하 직원들의 해당 소프트웨어와 관련된 내부적인 사용을 위해 합당한 수의 문서 복사본을 인쇄 또는 제작할 수 있습니다. 단, 이 경우 각 복사본에는 전체 CA 저작권 정보와 범례가 첨부되어야 합니다.

본건 문서의 사본 인쇄 또는 제작 권한은 해당 소프트웨어의 라이선스가 전체 효력을 가지고 유효한 상태를 유지하는 기간으로 제한됩니다. 어떤 사유로 인해 라이선스가 종료되는 경우, 귀하는 서면으로 문서의 전체 또는 일부 복사본이 CA에 반환되거나 폐기되었음을 입증할 책임이 있습니다.

CA는 관련법의 허용 범위 내에서, 상품성에 대한 묵시적 보증, 특정 목적에 대한 적합성 또는 권리 위반 보호를 비롯하여(이에 제한되지 않음) 어떤 종류의 보증 없이 본 문서를 "있는 그대로" 제공합니다. CA는 본 시스템의 사용으로 인해 발생되는 직, 간접 손실이나 손해(수익의 손실, 사업 중단, 영업권 또는 데이터 손실 포함)에 대해서는(상기 손실이나 손해에 대해 사전에 명시적으로 통지를 받은 경우라 하더라도) 귀하나 제 3자에게 책임을 지지 않습니다.

본건 문서에 언급된 모든 소프트웨어 제품의 사용 조건은 해당 라이선스 계약을 따르며 어떠한 경우에도 이 문서에서 언급된 조건에 의해 라이선스 계약이 수정되지 않습니다.

본 문서는 CA에서 제작되었습니다.

본 시스템은 "제한적 권리"와 함께 제공됩니다. 미합중국 정부에 의한 사용, 복제 또는 공개는 연방조달규정(FAR) 제 12.212 조, 제 52.227-14 조, 제 52.227-19(c)(1)호 - 제(2)호 및 국방연방구매규정(DFARS) 제 252.227-7014(b)(3)호 또는 해당하는 경우 후속 조항에 명시된 제한 사항을 따릅니다.

Copyright © 2013 CA. All rights reserved. 본 시스템에서 언급된 모든 상표, 상호, 서비스 표시 및 로고는 각 해당 회사의 소유입니다.

## CA Technologies 제품 참조

이 문서에서는 다음과 같은 CA Technologies 제품과 기능을 참조합니다.

- 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)

## CA 에 문의

### 기술 지원팀에 문의

온라인 기술 지원 및 지사 목록, 기본 서비스 시간, 전화 번호에 대해서는  
<http://www.ca.com/worldwide>에서 기술 지원팀에 문의하십시오.



# 목차

---

## 제 1 장: EPAgent 설치, 구성 및 사용 13

Environment Performance Agent 개요.....	13
EPAgent 가 프로세스에서 입력을 받을 수 있음 .....	13
EPAgent 및 스크립팅 환경 .....	15
EPAgent 는 네트워크 원본에서의 입력을 허용할 수 있음 .....	16
EPAgent 및 CA Introscope® .....	16
지원되는 플러그 인 .....	16
Environment Performance Agent 설치.....	18
EPAgent 독립 실행형 설치 관리자.....	19
EPAgent 구성 .....	20
EPAgent 속성 구성 .....	20
EPAgent 로깅 옵션 구성 .....	21
EPAgent 플러그 인 구성 .....	22
네트워크 데이터 원본을 위한 EPAgent 구성 .....	25
EPAgent 실행 .....	26
UNIX에서 EPAgent 실행 .....	27
EPAgent에서 사용자 지정 플러그 인 생성 및 사용 .....	28
메트릭 데이터 형식 .....	28
사용자 지정 플러그 인 형식의 유형에 따른 주의 사항 .....	30
오류 또는 이벤트 데이터 형식 .....	30
XML 형식의 오류 스냅숏 생성 .....	31
EPAgent 이벤트 및 트랜잭션 추적 .....	32
오류 또는 이벤트 데이터에 대한 XML 스키마 .....	34

## 제 2 장: EPAgent 기본 플러그 인 35

solaris/diskStats.pl .....	35
생성되는 메트릭 .....	35
명령줄 인수 .....	35
windows/processAvailability.pl AND solaris/processAvailability.pl .....	36
생성되는 메트릭 .....	36
명령줄 인수 .....	36
windows/diskStats.pl .....	37
생성되는 메트릭 .....	37
명령줄 인수 .....	37

---

apacheLogReader.pl, weblogicLogReader.pl 및 websphereLogReader.pl.....	37
생성되는 메트릭 .....	38
명령줄 인수 .....	38
httpSvcAvailability.pl.....	38
생성되는 메트릭 .....	38
명령줄 인수 .....	39
Perl 지원 문제 .....	40
kstat.pl.....	40
생성되는 메트릭 .....	40
명령줄 인수 .....	40
Perl 지원 문제 .....	41
iisLogReader.pl .....	41
생성되는 메트릭 .....	42
명령줄 인수 .....	42
iisLogReader.cfg 구성 파일.....	42
IIS4.0 과 IIS5.0 에 로그인하는 IIS 웹 서버 구성 .....	51

## 제 3 장: **EPAgent** 기본 플러그 인 메트릭 53

잘못된 문자 바꾸기 .....	53
IIS 로그 판독기 메트릭 .....	54
Total Page Hits(총 페이지 방문 횟수).....	54
Total Visitors Count(총 방문자 수).....	54
Page Hits For <URL>(<URL> 페이지 방문 횟수).....	55
Bytes Transferred(전송된 바이트).....	55
ErrorCount_400_BadRequest .....	55
ErrorCount_401_Unauthorized .....	56
ErrorCount_403_FORBIDDEN .....	56
ErrorCount_404_ServerConfiguration .....	56
ErrorCount_500_InternalServerError .....	57
ErrorMessage_400_BadRequest.....	57
ErrorMessage_401_Unauthorized .....	57
ErrorMessage_403_FORBIDDEN .....	58
ErrorMessage_404_PageNotFound .....	58
ErrorMessage_500_InternalServerError .....	58
HTTP/HTTPS 서비스 가용성 메트릭 .....	59
사용 가능 .....	59
응답 시간 .....	59
Average Throughput(평균 처리량) .....	60
Solaris 커널 통계(Kstat) 메트릭 .....	60
Solaris 커널 GHOST <module> <name> 메트릭 .....	60

---

프로세스 가용성 메트릭 .....	61
수 .....	61
디스크 통계(Windows) 메트릭 .....	61
Disk Reads/sec(초당 디스크에서 읽기 작업 횟수).....	62
Disk Writes/sec(초당 디스크에서 쓰기 작업 횟수).....	62
Disk Read Bytes/sec(초당 디스크에서 읽기 바이트 수).....	63
Disk Write Bytes/sec(초당 디스크에서 쓰기 바이트 수).....	63
Current Disk Queue Length(현재 디스크 큐 길이).....	63
Average Disk Read Time(평균 디스크에서 읽기 시간) (ms).....	64
Average Disk Write Time(평균 디스크에서 쓰기 시간) (ms).....	64
Average Disk Bytes/Read(쓰기당 평균 디스크에 쓴 바이트 수).....	64
Average Disk Bytes/Write(쓰기당 평균 디스크에 쓴 바이트 수).....	65
사용 가능한 공간(mb) .....	65
사용 가능한 공간(%).....	65
디스크 통계(Solaris) 메트릭 .....	66
Disk Time(디스크 시간) (%).....	66
Transaction Waiting Time(트랜잭션 대기 시간) (%) .....	67
Average # Transaction Active(평균 활성 트랜잭션 수).....	67
Average # Transaction Waiting(평균 트랜잭션 대기 수) .....	67
Disk Reads/sec(초당 디스크에서 읽기 작업 횟수).....	68
Disk Writes/sec(초당 디스크에서 쓰기 작업 횟수).....	68
KB Read per Second(초당 읽은 KB) .....	68
KB Written per Second(초당 쓴 KB) .....	69
Average Service Time in Waiting Queue(대기 큐에서의 평균 서비스 시간) (ms) .....	69
Average Service Time for Active Transactions(활성 트랜잭션을 위한 평균 서비스 시간) (ms).....	69
사용 가능한 공간(mb) .....	70
사용 가능한 공간(%).....	70
Apache 로그 메트릭 .....	70
PageHitCount .....	71
ErrorCount_400_BadRequest .....	71
ErrorCount_401_Unauthorized .....	72
ErrorCount_403_Forbidden .....	72
ErrorCount_404_PageNotFound.....	72
ErrorCount_500_ServerConfiguration .....	73
ErrorMessage_400_BadRequest.....	73
ErrorMessage_401_Unauthorized .....	73
ErrorMessage_403_Forbidden.....	74
ErrorMessage_404_PageNotFound .....	74
ErrorMessage_500_ServerConfiguration.....	74
WebSphere 로그 메트릭 .....	75

---

MessageCount_Audit .....	75
MessageCount_Warning.....	76
MessageCount_EventDebug .....	76
MessageCount_Error .....	76
MessageCount_Fatal.....	77
MessageText_Audit.....	77
MessageText_Warning.....	77
MessageText_EventDebug.....	78
MessageText_Error .....	78
MessageText_Fatal .....	78
WebLogic 로그 메트릭 .....	82
ServerVersion.....	83
MessageCount_Warning.....	83
MessageCount_Error .....	83
MessageCount_Notify .....	84
MessageCount_Critical .....	84
MessageCount_Alert.....	84
MessageCount_Emergency.....	85
MessageText_Warning.....	85
MessageText_Error .....	85
MessageText_Notify.....	86
MessageText_Critical .....	86
MessageText_Alert .....	86
MessageText_Emergency .....	87

---

<b>제 4 장: EAgent 속성</b>	<b>89</b>
IntroscopeEAgent.properties 파일 .....	89
에이전트 및 프로세스 이름 설정 .....	90
에이전트 메트릭 만료 처리 .....	91
에이전트 메트릭 만료 처리 구성 .....	92
introscope.agent.metricAging.turnOn .....	93
introscope.agent.metricAging.heartbeatInterval .....	93
introscope.agent.metricAging.dataChunk .....	94
introscope.agent.metricAging.numberTimeslices .....	94
introscope.agent.metricAging.metricExclude.ignore.0 .....	95
Case-Sensitive 메트릭 .....	95
introscope.epagent.metricscasesensitive .....	95
ChangeDetector 구성 .....	96
introscope.changeDetector.enable.....	96
introscope.changeDetector.rootDir.....	97
introscope.changeDetector.isengardStartupWaitTimeInSec.....	97

---

introscope.changeDetector.waitTimeBetweenReconnectInSec.....	98
introscope.changeDetector.enableEPA .....	98
introscope.changeDetector.agentID .....	99
introscope.changeDetector.profile .....	99
introscope.changeDetector.profileDir .....	100
Enterprise Manager 연결 .....	100
introscope.agent.enterpriseManager.connectionorder .....	100
introscope.agent.enterpriseManager.transport.tcp.host.DEFAULT .....	101
EPAgent 구성 .....	101
introscope.epagent.config.networkDataPort .....	101
introscope.epagent.config.httpServerPort .....	102
화장 디렉터리 .....	102
introscope.agent.extensions.directory .....	102
Logging(로깅) .....	103
log4j.logger.EPAgent .....	103
log4j.appenders.logfile.File.....	103
프로세스.....	103
introscope.agent.customProcessName .....	104
introscope.agent.defaultProcessName .....	104
introscope.agent.agentName .....	105
원격 구성 .....	105
introscope.agent.remoteagentconfiguration.enabled .....	105
상태 저장 플러그 인 .....	106
introscope.epagent.plugins.stateful.names.....	106
introscope.epagent.stateful.<pluginname>.command .....	106
introscope.epagent.stateful.<pluginname>.class .....	107
상태 비저장 플러그 인 .....	107
introscope.epagent.plugins.stateless.names.....	108
introscope.epagent.stateless.<pluginname>.command .....	108
introscope.epagent.stateless.<pluginname>.class .....	108
introscope.epagent.stateless.<pluginname>.delayInSeconds .....	109
introscope.epagent.stateless.<pluginname>.schedule.....	109
introscope.epagent.config.stalledStatelessPluginTimeoutInSeconds .....	110
introscope.epagent.stateless.<pluginname>.metricNotReportedAction .....	111



# 제 1 장: EPAgent 설치, 구성 및 사용

---

이 단원에서는 CA APM Environment Performance Agent(EPAgent)를 설치하고 구성하고 사용하는 방법을 다룹니다.

이 섹션은 다음 항목을 포함하고 있습니다.

[Environment Performance Agent 개요](#) (페이지 13)

[Environment Performance Agent 설치](#) (페이지 18)

[EPAgent 구성](#) (페이지 20)

[EPAgent 실행](#) (페이지 26)

[UNIX에서 EPAgent 실행](#) (페이지 27)

[EPAgent에서 사용자 지정 플러그인 생성 및 사용](#) (페이지 28)

## Environment Performance Agent 개요

CA APM Environment Performance Agent(EPAgent)는 비 Java 소스로부터의 메트릭 데이터를 CA Introscope로 통합하는 기능을 합니다.

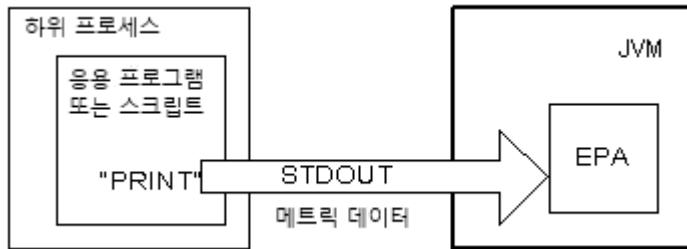
많은 운영 체제가 프로세스에서 하위 프로세스를 생성하는 기능을 제공합니다. EPAgent는 이러한 기능을 활용하여 운영 체제 파이프를 통해(표준 IPC 메커니즘) 하위 프로세스로부터 표준 출력을 수신합니다. Perl과 같은 유연한 스크립팅 환경과 하위 프로세스를 실행하는 기능을 결합할 경우 융통성과 막강한 기능이 발휘되어 사실상 모든 출처에서 응용 프로그램 성능 정보를 가져올 수 있습니다.

## EPAgent가 프로세스에서 입력을 받을 수 있음

EPAgent는 Java에서 구현되었으며 JVM(Java Virtual Machine)에서 실행됩니다. EPAgent는 JVM 기능을 사용하여 하위 프로세스를 호출하고 해당 프로세스에서 STDOUT 및 STDERR를 수신합니다. 하위 프로세스는 명령줄에서 호출할 수 있는 스크립트, 실행 파일 또는 기타 항목일 수 있습니다. 예:

- 사용자가 C로 작성한 컴파일된 프로그램
- Perl 스크립트

하위 프로세스의 응용 프로그램 또는 스크립트에서는 간단한 PRINT 함수를 실행하여 EPAGent 와 통신할 수 있습니다. 다음 그림에서는 응용 프로그램이나 스크립트에서 STDOUT 에 인쇄하여 EPAGent 에 메트릭 데이터를 전달하는 방법을 보여 줍니다.



EPAGent 에서는 STDOUT 를 사용하여 수신되는 텍스트가 다음 특정 형식 중 하나를 준수해야 합니다.

- 단순 이름/값 쌍

또는

- XML

EPAGent 텍스트 형식 요구 사항 및 옵션에 대해서는 이 안내서 뒷부분에서 자세히 설명합니다.

이 안내서에서는 EPAGent 에 데이터를 전송하는 응용 프로그램이나 스크립트를 EPAGent 플러그 인이라고 합니다. EPAGent 에는 자세한 메트릭 데이터를 제공하는 여러 기본 플러그 인이 포함되어 있습니다. 사용자 지정 플러그 인을 만들 수도 있습니다.

**추가 정보:**

[EPAGent 플러그 인 구성 \(페이지 22\)](#)

[EPAGent에서 사용자 지정 플러그 인 생성 및 사용 \(페이지 28\)](#)

[EPAGent 기본 플러그 인 \(페이지 35\)](#)

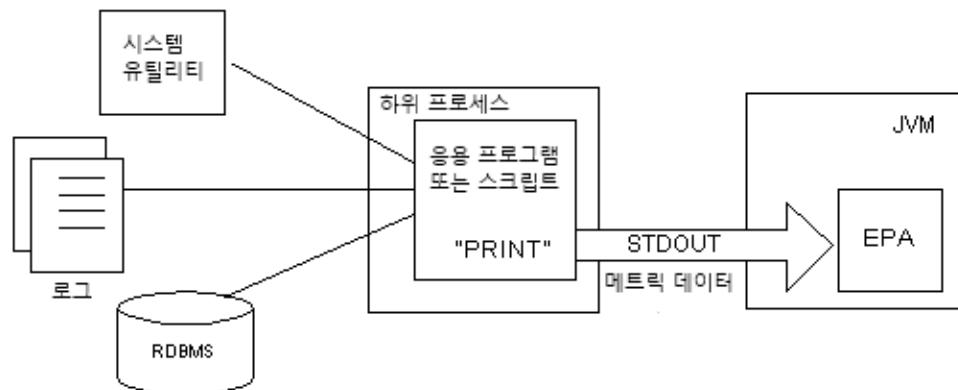
## EPAgent 및 스크립팅 환경

EPAgent 는 데이터를 수신할 때 하위 프로세스의 실행 가능한 엔터티(컴파일된 응용 프로그램 포함)로부터 텍스트를 수신할 수도 있지만, Perl, Korn Shell 또는 rexx 와 같은 스크립팅 환경을 사용해 접근하는 방법이 가장 유연합니다. 인터프리터식 스크립팅 환경은 구현이 쉬운 장점을 가지며, 관계형 데이터베이스와 같은 실제 데이터 원본, OS 하위 시스템(프로세스 테이블, 파일 시스템 등) 및 타사 제품에 대한 인터페이스나 라이브러리를 제공하기도 합니다.

스크립트를 사용하면 다양한 출처에서 정보를 수집할 수 있습니다. 예를 들어 스크립트는 다음과 같은 기능을 합니다.

- 데이터베이스와 같은 중요한 응용 프로그램의 존재를 확인하기 위해 라이브러리 함수나 시스템 유ти리티를 호출합니다.
- 응용 프로그램 오류를 감지하기 위해 응용 프로그램 로그 파일을 검색하고 구문 분석합니다.

스크립팅 환경은 유연하기 때문에 EPAgent 는 사실상 모든 출처에서 성능 및 관리 정보를 수집할 수 있습니다. 이 그림은 스크립트가 다양한 출처에서 정보를 액세스하는 방법을 보여 줍니다.



Perl 인터프리터는 광범위한 플랫폼에서 사용할 수 있고 운영 체제, 미들웨어 및 타사 제품에 대한 데이터 API 를 폭넓게 지원하므로 EPAgent 를 Perl 과 함께 사용하는 것이 바람직합니다. CA Introscope?EPAgent 에는 샘플 Perl 스크립트 집합이 포함되어 있어 프로세스 가용성을 확인하거나 디스크 성능 통계를 가져오거나 HTTP 로그를 읽어 오는 등 다양한 기능을 수행할 수 있습니다. 이러한 스크립트를 확장해 다른 기능을 수행할 수도 있습니다.

**참고:** PBD, EPAgent 스크립트 또는 JavaScript 계산기의 사용자 지정과 관련된 지원을 받으려면 CA Technologies 전문 서비스에 문의하십시오.

## EPAgent 는 네트워크 원본에서의 입력을 허용할 수 있음

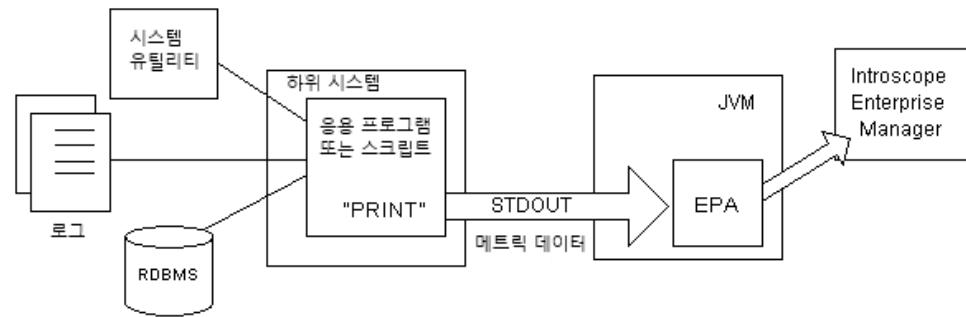
EPAgent 는 로컬 프로세스에서의 입력뿐 아니라 네트워크 원본에서의 입력도 허용할 수 있습니다. 이 기능은 네트워크 포트를 통해 모니터링 정보를 제공하거나 웹 URL에 정기적으로 데이터를 게시하는 C++ 백엔드 시스템의 경우처럼 EPAgent에서 실행할 수 없는 네트워크로 연결된 프로세스에서 정보를 수집하는 데 유용합니다.

추가 정보:

[네트워크 데이터 원본을 위한 EPAgent 구성](#) (페이지 25)

## EPAgent 및 CA Introscope®

EPAgent는 독립 실행형 프로세스로 실행되며 Java Agent나 .NET Agent처럼 관리 및 성능 정보를 제공하기 위해 Enterprise Manager에 연결됩니다. 이 그림은 EPAgent가 Enterprise Manager로 데이터를 전달하는 방법을 보여 줍니다.



EPAgent를 사용하면 응용 프로그램 환경의 전반적인 모습을 더욱 잘 파악할 수 있습니다. 분산 응용 프로그램 아키텍처는 복잡하며 Java와 비 Java 응용 프로그램 하위 시스템을 혼용하여 구성하기도 합니다. EPAgent를 사용하면 응용 프로그램 환경의 비 Java 및 비 .NET 부분에서 정보를 쉽게 수집해 CA Introscope?Workstation에서 Java 및 .NET 메트릭과 함께 나타낼 수 있습니다.

## 지원되는 플러그인

EPAgent는 상태 저장 및 상태 비저장 플러그인을 모두 지원합니다.

참고: 모든 기본 플러그인에는 Perl 버전 5.00503 이상이 필요합니다.

## 플러그 인 유형

사용 가능한 플러그 인 유형은 다음과 같습니다.

- **상태 저장 플러그 인-**상태 저장 플러그 인은 EPAgent 가 시작되면 작동하기 시작해 일반적으로 무한정 실행되는 스크립트(데몬)입니다. 이러한 상태 저장 플러그 인은 플러그 인의 표준 출력 채널을 통해 데이터를 CA Introscope®로 언제든 되돌려 보낼 수 있습니다. 상태 저장 플러그 인은 종료될 경우 EPAgent 에 의해 다시 시작됩니다.
- **상태 비저장 플러그 인-**상태 비저장 플러그 인은 반복적인 일정에 따라 실행되도록 설계되었으며 실행될 빈도(실행 간 지연 시간을 지정)를 설정해 구성할 수 있습니다. 이 플러그 인은 일반적으로 단기간 실행되는 스크립트로서 일부 데이터를 수집하여 표준 출력 채널을 통해 EPAgent 로 전송한 후 종료할 목적으로 사용됩니다.

**참고:** EPAgent 는 상태 비저장 플러그 인의 인스턴스를 한 번에 하나만 실행하므로 특별한 오류 확인은 하지 않습니다. 따라서 플러그 인 개발자는 상태 비저장 플러그 인이 비교적 짧은 시간 내에 실행을 마치고 완료되도록 설계해야 합니다.

## 오류 로깅

플러그 인의 표준 오류 채널을 통해 오류를 기록할 수 있으므로 EPAgent 는 다른 오류와 마찬가지로 이 오류 출력을 기록할 수 있습니다.

## 기본 플러그 인

EPAgent 에는 최소 구성으로 사용할 수 있는 기본 플러그 인 라이브러리가 포함되어 있습니다. 다음 표에서는 각 플러그 인의 기능과 이들이 지원되는 운영 체제에 대해 간략히 설명합니다.

플러그 인	기능	지원되는 운영 체제
프로세스 가용성 확인	지정된 이름으로 실행 중인 프로세스 찾기	Solaris
OS 수준 디스크 통계	읽기/쓰기, 트랜잭션 시간, % 대기 시간, 디스크 공간 등의 보고	Solaris
Apache 로그 판독기	HTTP 오류 수, 트랜잭션 수, 평균 트랜잭션 크기 보고	Solaris

플러그 인	기능	지원되는 운영 체제
WebLogic 로그 판독기	오류 수, 경고 수 보고	Solaris
WebSphere 로그 판독기	오류 수, 경고 수 보고	Solaris
IIS 로그 판독기	유형별 오류 수, 제공된 페이지 수, IP 주소별 사용자 수 보고	Windows
HTTP 서비스 가용성 HTTPS 서비스 가용성	연결 성공/실패, 연결 왕복 시간, 평균 처리량/대역폭 보고	Solaris
Solaris Kstat 통계	Kstat 유ти리티를 통해 제공되는 모든 통계(I/O 통계, 모든 장치에 대한 네트워크 통계, 가상 메모리 통계 등) 보고	Solaris

### 사용자 지정 플러그 인

EPAgent에서는 지원되는 두 가지 형식 중 하나로 출력을 생성하는 임의의 프로그램 및 플러그 인에서 데이터를 수집할 수 있습니다.

추가 정보:

[EPAgent에서 사용자 지정 플러그 인 생성 및 사용 \(페이지 28\)](#)

## Environment Performance Agent 설치

EPAgent 설치에는 Standalone Agent 설치 관리자가 사용됩니다.

참고: 자세한 내용은 *CA APM 설치 및 업그레이드 안내서*를 참조하십시오.

## EPAgent 독립 실행형 설치 관리자

다음과 같은 Standalone Agent 설치 관리자 중 하나를 사용하여 다른 컴퓨터에 EPAgent 를 설치할 수 있습니다. [CA Support](#)의 CA APM 소프트웨어 다운로드 영역에서 CA APM 제품을 다운로드할 수 있습니다.

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

EPAgent 설치 관리자를 추출하면 다음 파일이 <EPAgent\_Home>/epagent 디렉터리에 설치됩니다.

참고: <EPAgent\_Home>은 EPAgent 가 설치된 컴퓨터입니다.

- IntroscopeEPAgent.properties
- lib/EPAgent.jar
- epaplugins/ (플러그 인을 위한 최상위 디렉터리)
  - 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 파일을 위한 최상위 디렉터리)
  - ChangeDetectorAgent.jar
  - ChangeDetector-Agent\_Server.jar
  - ChangeDetector-CommonAll.jar
  - ChangeDetector-config.xml
  - Windows: cdnativefile.dll
  - UNIX: libcdnativefile.so

## EPAgent 구성

EPAgent에는 Enterprise Manager 위치 및 포트 설정과 같은 에이전트 구성 속성 집합이 포함되어 있습니다.

**참고:** 속성 파일 설정에서 잘못된 파일 이름 경로와 같은 오류가 발생하는 경우 EPAgent가 시작하지 못합니다. 시작할 때 EPAgent를 자동으로 시작하려는 경우 속성이 올바로 설정되어 있는지 확인하십시오.

추가 정보:

[EPAgent 속성 구성](#) (페이지 20)

[EPAgent 로깅 옵션 구성](#) (페이지 21)

[EPAgent 플러그인 구성](#) (페이지 22)

[네트워크 데이터 원본을 위한 EPAgent 구성](#) (페이지 25)

## EPAgent 속성 구성

EPAgent가 사용하는 속성은 Java Agent가 사용하는 속성과 동일합니다.

다음 단계를 수행하십시오.

1. *IntroscopeEPAgent.properties* 파일에서 EPAgent 설정을 구성합니다. EPAgent의 설정 유형은 CA Introscope® 에이전트 프로필의 유형과 동일합니다.

**참고:** 속성에 대한 자세한 내용은 *CA APM Java Agent 구현 안내서*를 참조하십시오.

2. *IntroscopeEPAgent.properties* 파일의 이름이나 위치를 변경하는 경우 다음과 같이 Java 시스템 속성을 사용하여 파일을 설정할 수 있습니다.

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

**참고:** 명령줄에서 이 시스템 속성은 "java" 바로 다음에 나와야 합니다.  
명령줄에서 나중에 오게 되면(예를 들어 *-jar* 다음) 작동하지 않습니다.

## EPAgent 로깅 옵션 구성

기본적으로 EPAgent 는 명령 콘솔에 메시지 및 오류 출력을 보냅니다.  
EPAgent에서 메시지 및 오류 출력을 로그 파일로도 전송하도록 구성할 수 있습니다.

다음 단계를 수행하십시오.

1. <EPAgent\_Home>/epagent/IntroscopeEPAgent.properties 파일을 엽니다.
2. 속성을 수정합니다.

## EPAgent 를 세부 정보 표시 모드로 실행하고 출력 파일을 리디렉션

EPAgent 를 세부 정보 표시 모드로 실행하면 세부 정보가 로그에 기록되어 디버그할 때 도움을 받을 수 있습니다. 세부 정보 표시 모드를 제어하는 속성에 따라 출력 로그 파일의 배치도 결정됩니다.

다음 단계를 수행하십시오.

1. <EPAgent\_Home>/epagent/IntroscopeEPAgent.properties 파일을 엽니다.

2. 다음 예와 같이 log4j.logger.EPAgent 속성을 주석으로 처리합니다.  
`#log4j.logger.EPAgent=INFO, console`
3. log4j.logger.IntroscopeAgent 속성의 주석 처리를 제거하고 INFO 를  
VERBOS#com.wily.util.feedback.Log4JSeverityLevel 로 교체하고  
console 을 logfile 로 교체하면 출력이 콘솔 대신 로그 파일로 전송됩니다.  
예를 들어 EPAgent 가 세부 정보 표시 모드로 로그 파일에 로그를  
보고하도록 하려면 속성을 다음과 같이 설정해야 합니다.  
`log4j.logger.IntroscopeAgent=VERBOS#com.wily.util.feedback.Log4JSeverityLeve  
l,logfile`
4. 변경 사항을 저장하고 IntroscopeEPAgent.properties 파일을 닫습니다.

**중요!** IntroscopeEPAgent.properties 파일을 저장한 후에도 properties 파일이  
확인되어 구현할 때까지는 EPAgent 로그가 명령 콘솔에 계속 표시됩니다.  
properties 파일이 구현되면 그 시점 이후부터는 로그 파일로 로그가  
기록됩니다.

## EPAgent 플러그 인 구성

기본 플러그 인을 사용하여 EPAgent 를 실행하려면  
IntroscopeEPAgent.properties 파일에서 특정 속성의 주석 처리를 제거하면  
됩니다.

그러나 IntroscopeEPAgent.properties 파일의 기본 플러그 인에서 필요하지  
않은 플러그 인을 제거하거나 다른 플러그 인을 추가할 수도 있습니다.

EPAgent 플러그 인은 IntroscopeEPAgent.properties 파일에서 상태 저장 및  
상태 비저장의 두 가지 섹션으로 구분되어 있습니다.

### 상태 저장 플러그 인 추가

EPAgent 가 사용하는 플러그 인은 IntroscopeEPAgent.properties 파일에  
구성됩니다.

다음 단계를 수행하십시오.

1. <EPAgent\_Home>/epagent/IntroscopeEPAgent.properties 파일을 엽니다.

2. *introscope.epagent.plugins.stateful.names* 속성의 주석 처리를 제거하고 모든 상태 저장 플러그인의 이름을 추가하고 다음 예처럼 쉼표로 구분합니다.  
*introscope.epagent.plugins.stateful.names=APACHE,WEBLOGIC,WEBSPHHERE,IIS*
3. 이전 단계에서 추가된 상태 저장 플러그인 이름 각각에 대해 행을 하나씩 추가해 상태 저장 플러그인을 실행하거나 시작하는 데 필요한 전체 명령(명령줄 인수도 포함할 수 있음)을 지정합니다.

**참고:** 파일 내에서 기본 속성의 주석 처리를 제거해 사용할 수 있습니다. 그러나 이 경우에도 플러그인을 실행하거나 시작하는 데 필요한 전체 명령줄을 추가해 주어야 합니다.

이전 단계에서 지정된 상태 저장 이름 속성을 위해서는 다음과 같은 네 행을 추가해 주어야 합니다.

```
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.WEBSPHHERE.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
```

**참고:** Windows 컴퓨터에서 경로를 추가할 경우 백슬래시(\)는 백슬래시를 하나 더 사용해 중첩하여 이스케이프해야 합니다. 예를 들어 다음과 같습니다.

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

**추가 정보:**

[EPAgent 기본 플러그인](#) (페이지 35)

## 다음 단계

[상태 비저장 플러그인을 추가](#) (페이지 24) 할 수 있습니다. 상태 비저장 플러그인을 추가하지 않으려면 IntroscopeEPAgent.properties 파일에 변경 사항을 저장하고 EPAgent를 다시 시작하십시오.

## 상태 비저장 플러그 인 추가

상태 비저장 플러그 인을 추가할 수 있습니다.

다음 단계를 수행하십시오.

1. <EPAgent\_Home>/epagent/IntroscopeEPAgent.properties 파일을 엽니다.
2. *introscope.epagent.plugins.stateless.names* 속성의 주석 처리를 제거하고 모든 상태 비저장 플러그 인의 논리적 이름을 추가하고 다음 예처럼 첨표로 구분합니다.

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

3. 이전 단계에서 추가된 상태 비저장 플러그 인의 이름에 대해 두 개의 행을 추가해 다음을 지정합니다.

- 상태 비저장 플러그 인을 실행하거나 시작하는 데 필요한 전체 명령줄을 다음과 같은 구문으로 지정합니다. 필요한 경우 "EPAgent 기본 플러그 인"을 참조해 명령줄 인수를 포함하도록 합니다.

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

- EPAgent 가 플러그 인을 실행할 빈도(지연 시간)를 다음과 같은 구문을 이용해 초 단위로 지정합니다.

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

예:

```
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
```

참고: 위에 나온 예는 이러한 속성의 기본값입니다. 이러한 값을 사용하려면 각 속성의 주석 처리를 제거해야 합니다.

4. 마쳤으면 변경 사항을 저장하고 EPAgent 를 다시 시작합니다.

추가 정보:

[EPAgent 기본 플러그 인](#) (페이지 35)

## 상태 저장 및 상태 비저장 플러그 인 제거

언제든지 플러그 인을 제거할 수 있습니다.

다음 단계를 수행하십시오.

1. <EPAgent\_Home>/epagent/IntroscopeEPAgent.properties 파일을 엽니다.
2. *introscope.epagent.plugins.stateful.names* 및 *introscope.epagent.plugins.stateless.names* 속성에서 제거할 플러그 인의 이름을 삭제합니다.
3. 삭제하는 플러그 인과 연결된 명령줄을 주석 처리하거나 삭제합니다.
4. 마쳤으면 변경 사항을 저장하고 EPAgent 를 다시 시작합니다.

## Java 클래스 플러그 인 구성

EPAgent JVM 내에서 실행되는 Java 클래스인 플러그 인을 구성하려면 EPAgent 속성 파일에 다음 속성을 구성하십시오.

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

참고: 정규화된 클래스 이름 뒤에 다음 예와 같은 선택적 인수를 지정하십시오.

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

## 네트워크 데이터 원본을 위한 EPAgent 구성

EPAgent 가 네트워크 원본에서 데이터를 수신하도록 구성할 수 있습니다.

- [단순 입력 또는 XML 네트워크 입력을 위한 EPAgent 구성](#) (페이지 25)
- [HTTP GET 입력을 위한 EPAgent 구성](#) (페이지 26)

### 단순 입력 또는 XML 네트워크 입력을 위한 EPAgent 구성

EPAgent 가 네트워크를 통해 데이터를 수신할 수 있도록 하려면 EPAgent 속성 파일에서 다음 행의 주석 처리를 제거하십시오.

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

## HTTP GET 입력을 위한 EPAgent 구성

EPAgent에서 HTTP GET 요청을 수신 대기하는 단순 HTTP 서버를 생성할 수 있습니다. HTTP GET 요청에 따라 Enterprise Manager에 전송되는 메트릭이 결정됩니다.

HTTP GET 요청에서는 다음과 같은 구문을 사용합니다.

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

단순 HTTP 서버를 사용하도록 설정하고 테스트하려면

1. EPAgent 속성 파일에서 다음 속성의 주석 처리를 제거하여 단순 HTTP 서버를 사용하도록 설정합니다.

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

2. EPAgent를 다시 시작합니다.

변경 내용이 적용됩니다.

3. 브라우저를 통해 HTTP GET 요청을 테스트합니다. 예:

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

15초 후 MyHTTPMetric|Test:MyCounter 메트릭이 EPAgent 아래에 표시됩니다.

메트릭을 업데이트하려면 EPAgent가 새 HTTP GET 요청을 수신해야 합니다. 따라서 단순 HTTP 서버 기능은 여러 HTTP GET 요청을 생성할 수 있는 도구와 함께 사용하기 위한 것입니다. 예를 들어 오픈 소스 도구 Wget로 여러 wget 명령을 실행하여 여러 HTTP GET 요청을 생성할 수 있습니다. Wget에 대한 자세한 내용은 다음 사이트를 참조하십시오.

```
http://www.gnu.org/software/wget/wget.html
```

## EPAgent 실행

EPAgent는 독립 실행형 .jar 파일이나 Java 응용 프로그램으로 실행될 수 있습니다.

EPAgent를 독립 실행형 .jar 파일로 실행하려면 다음과 같이 하십시오.

- 다음 예처럼 적절한 -jar 플래그를 사용하여 Java 명령줄을 실행합니다.  

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

EAgent를 Java 응용 프로그램으로 실행하려면 다음과 같이 하십시오.

- 다음 예처럼 EAgent 파일을 적절한 classpath에 추가합니다.

```
java -classpath "<EAgent_Home>/epagent/lib/EAgent.jar"  
-Dcom.wily.introscope.epagent.properties=<EAgent_Home>/epagent/IntroscopeEP  
Agent.properties"
```

## UNIX에서 EAgent 실행

UNIX 운영 체제에서 제어 스크립트(셸 스크립트)를 사용하여 Introscope EAgent를 실행할 수 있습니다.

다음 단계를 수행하십시오.

1. 명령 프롬프트를 엽니다.
2. 제어 스크립트가 있는 디렉터리로 이동합니다. 예:  
`cd Introscope<version_number>/bin`
3. 원하는 작업에 해당하는 명령을 실행합니다.

### EPACtrl.sh start

EAgent를 시작합니다.

### EPACtrl.sh status

EAgent 프로세스의 현재 상태("실행 중" 또는 "중지됨")를 표시합니다.

### EPACtrl.sh stop

EAgent 프로세스를 중지합니다.

### EPACtrl.sh help

도움말 메뉴를 표시합니다.

## EPAgent에서 사용자 지정 플러그인 생성 및 사용

기본 플러그인과 함께 EPAgent를 실행하면 일반적으로 대부분의 사용자 요구를 충족하는 정보가 수집됩니다. 그러나 EPAgent에서 사용자 지정 플러그인을 사용해야 할 수도 있습니다. 이렇게 하려면 다음 두 가지 치침을 따르십시오.

- 사용자 지정 플러그인의 출력을 EPAgent에서 구문 분석할 수 있는지 확인하십시오([메트릭 데이터 형식](#)(페이지 28) 및 [오류 또는 이벤트 데이터 형식](#)(페이지 30) 참조).
- [EPAgent 플러그인 구성](#)(페이지 22) 단원에 설명된 대로 *IntroscopeEPAgent.properties* 파일에 사용자 지정 플러그인을 추가하십시오.

### 메트릭 데이터 형식

EPAgent는 플러그인(또는 EPAgent에 연결된 다른 메트릭 생성 프로그램)에서 제공하는 다음과 같은 두 가지 형식의 메트릭 데이터를 구문 분석할 수 있습니다.

- 단순
- XML

#### 메트릭 데이터를 위한 단순 형식

행별로 다음 형식에 따라 하나의 메트릭 이름과 값을 지정합니다.

<metric\_name>=<value>

예:

diskWrites=37

리소스 세그먼트에 대한 참조를 포함할 수도 있습니다.

<resource\_segment>:<metric\_name>=<value>

예를 들면 다음과 같습니다(모두 한 행임).

Resource Usage|File IO:diskWrites=37

Apache Errors:LastErrorString=ERROR: Apache shutdown unexpectedly

단순 형식에 대한 지침은 다음과 같습니다.

- "단순" 형식에서는 메트릭 이름에 같음 기호(=)를 포함해서는 안 됩니다. 같음 기호를 이름에 포함해야 하는 경우는 XML 형식을 사용하십시오.
- 값에는 같음 기호(문자열)가 포함될 수 있습니다. 이 경우 EPAgent는 왼쪽에서 오른쪽으로 구문 분석할 때 첫 번째 같음 기호까지의 문자는 모두 메트릭 이름으로 간주하고 같음 기호 다음에 나오는 문자는 모두 값으로 간주합니다.
- 숫자로 구성된 값은 모두 숫자 데이터로 해석되며 CA Introscope® "IntCounter" 유형으로 보고됩니다.
- 숫자가 아닌 값으로 구성된 모든 값은 문자열 데이터로 해석되며 CA Introscope® "string event" 유형으로 보고됩니다.

## 메트릭 데이터를 위한 XML 형식

"단순" 형식에서는 Introscope 메트릭 유형을 제한하지만 XML 스타일 형식에서는 다음 예와 같이 플러그인에서 Introscope 메트릭 이름, Introscope 메트릭 유형 및 값과 같은 추가 정보를 보고할 수 있습니다.

```
<metric type="LongCounter" name="Resource Usage|File IO:diskWrites" value="37" />
<metric type="StringEvent" name="Apache Errors:LastErrorString" value="ERROR:
Apache shutdown unexpectedly" />
```

XML 형식 지침:

- 이 형식을 사용하면 메트릭 이름과 값 모두에서 Introscope 데이터 유형 및 같음 기호를 완벽히 지원할 수 있습니다.
- 메트릭의 "type" 특성은 다음 중 하나여야 합니다.
  - PerIntervalCounter - 값은 간격이 변경될 수 있는 "간격당" 비율입니다. 이러한 메트릭은 값을 합산하는 방식으로 시간에 따라 집계됩니다. 예를 들어 15 초당 메서드 호출 10 개가 발생한 다음 15 초당 메서드 호출 15 개가 발생한 경우 30 초로 집계하면 "30 초당 메서드 호출 25 개"가 됩니다.
  - IntCounter - 증가하거나 감소할 수 있는 int 값입니다.
  - IntAverage - 일정 시간 동안의 평균을 구한 int 값입니다.
  - IntRate - 값은 초당 비율입니다. 이러한 메트릭은 값의 평균을 구하는 방식으로 시간에 따라 집계됩니다.
  - LongCounter - 증가하거나 감소할 수 있는 long 값입니다.

- LongAverage - 일정 시간 동안의 평균을 구한 long 값입니다.
- StringEvent - 주기적으로 문자열을 생성하는 유형을 나타냅니다. 이 레코더는 현재 값이라는 개념이 없고 단순히 이벤트를 보고 받은 순서로 보고합니다.
- Timestamp - 연속적으로 증가하는 타임스탬프를 생성하는 유형입니다.
- 비교에서는 대/소문자를 구분하지 않으므로 플러그인 작성기가 쉽게 비교할 수 있습니다. 숫자 유형을 지정했지만 값이 숫자가 아니면 Introscope에 아무것도 보고되지 않고 EPAgent에서 오류를 기록합니다.

## 사용자 지정 플러그인 형식의 유형에 따른 주의 사항

사용자 지정 플러그인 형식의 유형 중 하나를 사용할 때 다음 사항에 주의하십시오.

- 두 형식을 모두 지원하면 시스템에서 보다 작은 기호로 시작하는 메트릭 이름을 인식하지 못하는 상호 작용이 발생합니다.
- 두 형식 모두에서 형식의 구문을 분석할 수 없으면(가비지 또는 잘못된 구문) EPAgent는 행을 무시하고 오류를 기록합니다.
- 플러그인이 여러 행을 반환하는 경우 구문 분석이 다음 행에서 계속됩니다.
- 각 메트릭 이름에 대해 메트릭 유형을 하나만 지정할 수 있습니다. 여러 유형을 지정하면 다음 오류가 발생합니다.  
`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`

## 오류 또는 이벤트 데이터 형식

EPAgent는 플러그인이 제공하는 다음 두 가지 형식의 오류 또는 이벤트 데이터를 구문 분석할 수 있습니다.

- 단순
- XML

## 오류 또는 이벤트 데이터를 위한 단순 형식

일반적으로 단순 형식 스크립트는 다음과 같은 고정 문자열로 시작합니다.

event:

콜론 다음의 모든 항목은 다음 중 하나입니다.

- "name=value" 쌍의 일부. 각 쌍은 앤페센드 문자(&)로 구분됩니다.
- 이벤트의 선택적 매개 변수

아래 예는 Firefox 브라우저 프로세스를 모니터링하고 브라우저가 종료될 때 알림을 보내는 가상스크립트의 출력입니다.

event:type=processWentAway&processName=firefox

## 오류 또는 이벤트 데이터를 위한 단순 XML 형식

이벤트를 XML 형식으로 지정할 수도 있습니다. 그러면 에이전트에서 이벤트에 대한 전체 표현 기능을 사용할 수 있습니다. 가장 단순한 XML 형식 이벤트에서는 이벤트를 생성한 리소스의 이름("Connection Pool", "Java VM" 등)을 제공합니다. 아래 예에서는 *Some Resource*에서 이벤트가 발생했다는 알림을 제공합니다

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

타임스탬프는 이벤트가 생성된 시간이고 이벤트의 기간은 0입니다.

## 매개 변수 및 시간 데이터가 포함된 오류 또는 이벤트 데이터를 위한 XML 형식

명시적 타임스탬프와 명시적 기간을 포함하는 이벤트 알림을 구성할 수 있습니다. 타임스탬프 형식은 구문 분석 가능한 Java 형식입니다. 기간은 밀리초입니다. 아래 예는 기간이 1 분(60,000 밀리초)인 이벤트입니다.

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

## XML 형식의 오류 스냅숏 생성

오류 스냅숏은 다음과 같이 매개 변수에 유형을 오류 스냅숏으로 지정해야 합니다.

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

## 중첩된 구성 요소

아래 예에서는 중첩된 하위 구성 요소가 있는 이벤트를 보여 줍니다.  
이벤트는 하위 구성 요소를 0 개에서 무한개까지 포함할 수 있으며 각 하위  
구성 요소도 하위 구성 요소를 0 개에서 무한개까지 포함할 수 있습니다.  
실제로는 중첩 수준이 작거나 0 입니다.

```
<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 이벤트 및 트랜잭션 추적

이벤트 뷰어에서 "추적 뷰" 탭을 선택해 트랜잭션 추적으로 EPAgent  
이벤트를 볼 수 있습니다.

EPAgent 가 보낸 이벤트에 시간 정보가 포함되어 있으면 더 쉽게 추적 뷰를  
이해할 수 있습니다. 시간 정보를 포함하려면 <event>와 <calledComponent>  
태그에서 startTime 과 offset 특성을 사용하십시오.

startTime 특성은 절대 시간을 나타냅니다. 특성의 형식은  
java.util.Date.parse()가 구문 분석 할 수만 있다면 어떤 형식이든 가능합니다.  
<event> 요소에서 startTime 을 반드시 지정해야 하는 것은 아닙니다.  
지정하지 않은 경우에는 Java 메서드인 System.currentTimeMillis()나 new  
Date().getTime()에서 지정된 현재 시간으로 기본값이 설정됩니다.  
<calledComponent> 요소에서 startTime 을 누락하면 포함하는 요소의  
시간으로 시간 기본값이 설정됩니다. 따라서 startTime 특성이 지정된  
요소가 어디에도 없다면 모두 기본값인 현재 시간으로 설정됩니다.

offset 특성은 정수 값입니다. 오프셋의 시간은 밀리초로 해석되며 startTime  
특성에 더해져 <event> 또는 <calledComponent>에 보고할 실제 시간이  
계산됩니다(startTime 은 기본값이거나 명시적일 수 있음).

### 예제 1

```
<event resource="Customized Web Server" startTime="123456789" duration="500">
<calledComponent resource="Web Server Module" offset="300" duration="100"/>
</event>
```

이 이벤트의 추적 뷰에는 1234567899에 시작하는 "Customized Web Server"와 123457089(123456789 + 300)에 시작하는 "Web Server Module"이 나타납니다. 각 요소에 기간을 지정하면 다음과 같은 유용한 추적 뷰를 얻을 수 있습니다.

- "Customized Web Server"를 300 밀리초 동안 실행
- "Customized Web Server"가 호출한 "Web Server Module"을 100 밀리초 동안 실행
- "Web Server Module"이 반환된 후 또 다른 100 밀리초 동안 "Customized Web Server"를 실행

### 예제 2

```
<event resource="Customized Web Server" duration="500">
<calledComponent resource="Web Server Module" offset="300" duration="100"/>
</event>
```

이 예제는 예제 1과 유사하지만 "Customized Web Server"가 현재 시간에 시작되고 "Web Server Module"이 300 밀리초 후에 시작된다는 점이 다릅니다. 이 예제의 어디에서도 EPAgent 스크립트가 현재 시간을 알 필요가 없다는 점에 주의하십시오.

### 예제 3

```
<event resource="Customized Web Server" startTime="123000000" offset="1000"
duration="5000">
<calledComponent resource="Web Server Module" startTime="123003000"
duration="200"/>
</event>
```

여기에서는 "Customized Web Server"가 123001000(123000000 + 1000)에 시작하고 "Web Server Module"은 123003000에 시작합니다.

위 예제에서는 가독성과 사용성 향상을 위해 기간이 정의되어 있습니다. `startTime`, 오프셋 및 기간을 잘못 지정하면 추적 뷰를 읽기 어려워집니다. 따라서 이 특성을 사용할 때는 주의가 필요합니다. 특히 `<calledComponent>` 요소에서의 `startTime`과 `offset`을 더해 계산되는 시작 시간의 경우에는 항상 자신이 포함되는 `<event>`나 `<calledComponent>`의 시작 시간 이후여야 합니다. `<calledComponent>`의 (시작 시간 + 기간)은 자신이 포함되는 `<event>`나 `<calledComponent>`의 (시작 시간 + 기간)보다 항상 작아야 합니다.

## 오류 또는 이벤트 데이터에 대한 XML 스키마

지원되는 공식 XSD 스키마는 다음과 같습니다.

```
<?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: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: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:complexType>
</xs:schema>
```

## 제 2 장: EPAGent 기본 플러그 인

---

이 단원에서는 CA APM Environment Performance Agent에서 제공하는 기본 플러그 인과 해당하는 선택적 명령줄 인수에 대해 설명합니다.

이 섹션은 다음 항목을 포함하고 있습니다.

[solaris/diskStats.pl](#) (페이지 35)

[windows/processAvailability.pl AND solaris/processAvailability.pl](#) (페이지 36)

[windows/diskStats.pl](#) (페이지 37)

[apacheLogReader.pl, weblogicLogReader.pl 및 websphereLogReader.pl](#) (페이지 37)

[httpSvcAvailability.pl](#) (페이지 38)

[kstat.pl](#) (페이지 40)

[iisLogReader.pl](#) (페이지 41)

### solaris/diskStats.pl

이 플러그 인은 시스템 하드 디스크 관련 정보가 들어 있는 메트릭을 반환합니다.

#### 생성되는 메트릭

*/solaris/diskStats.pl* 플러그 인에 의해 생성되는 메트릭은 Solaris 커널 통계(Kstat) 메트릭에서 자세히 설명합니다.

#### 명령줄 인수

이 플러그 인은 명령줄 인수로 디스크 이름을 사용합니다. 디스크 이름을 지정하지 않으면 모든 디스크에 대한 정보가 보고됩니다.

- 예를 들어 다음 명령은 모든 로컬 디스크에 대한 메트릭을 보고합니다.  
`perl diskStats.pl`
- 다음 명령은 '/' 및 '/usr' 디스크에 대한 메트릭만 보고합니다.  
`perl diskStats.pl / /usr`

**참고:** 이 인수를 사용하여 결과를 좁힐 수 있습니다.

## windows/processAvailability.pl AND solaris/processAvailability.pl

이 플러그 인은 전달된 인수 값과 일치하는 프로세스 이름의 수에 해당하는 값이 포함된 메트릭을 반환합니다. 반환되는 메트릭 유형은 'IntCounter'입니다.

### 생성되는 메트릭

*/windows/processAvailability.pl and solaris/processAvailability.pl* 플러그 인에 의해 생성되는 메트릭은 프로세스 가용성 메트릭에서 자세히 설명합니다.

### 명령줄 인수

이러한 각 플러그 인에서는 다음과 같은 다섯 가지 선택적 명령줄 인수를 사용합니다.

#### **match**

검색할 정규 Perl 표현식

#### **resource**

메트릭 이름의 리소스 부분

#### **subResource**

메트릭 이름의 하위 리소스 부분

#### **metric**

메트릭 이름의 주요 부분

#### **distinctMatch**

값이 없으며 각 정규식 일치 수를 개별적으로 유지 및 보고할지 여부를 결정합니다. 이 옵션은 **subResource** 인수의 값(지정된 경우)을 무시합니다.

다음 예에서는 문자 'sh'로 시작하는 프로세스 수를 보고합니다.

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

## windows/diskStats.pl

이 플러그 인은 Windows 성능 모니터에서 사용할 수 있는 통계를 바탕으로 디스크 사용량/용량 통계를 보고합니다.

이 플러그 인이 모든 통계를 적절하게 보고하도록 'diskperf -y' 명령을 실행해야 합니다. 그런 다음 새 설정이 활성화되도록 컴퓨터를 다시 부팅해야 합니다.

### 생성되는 메트릭

/windows/diskStats.pl 플러그 인에 의해 생성되는 메트릭은 디스크 통계(Windows) 메트릭에서 자세히 설명합니다.

### 명령줄 인수

이 플러그 인은 명령줄 인수로 디스크 이름을 사용합니다. 디스크 이름을 지정하지 않으면 모든 디스크에 대한 정보가 보고됩니다.

- 예를 들어 다음 명령은 모든 로컬 디스크에 대한 메트릭을 보고합니다.  
perl diskStats.pl
- 다음 명령은 c: 및 d: 디스크에 대한 메트릭만 보고합니다.  
perl diskStats.pl c: d:

**참고:** 이 인수를 사용하여 결과를 좁힐 수 있습니다.

## apacheLogReader.pl, weblogicLogReader.pl 및 websphereLogReader.pl

이러한 플러그 인은 <webserver>LogReader.cfg 파일에 구성된 메트릭을 기반으로 메트릭을 보고합니다.

웹 서버 응용 프로그램 로그 파일 파서마다 대기 지연, 하위 리소스 이름, 리소스 이름, 로그 파일 디렉터리 및 로그 파일 이름 패턴에 대한 기본값을 포함하는 자체의 구성 파일이 있습니다.

**참고:** 이러한 로그 판독기 플러그 인을 사용할 경우 로그 파일에 새 항목이 추가될 때까지 EAgent 가 Enterprise Manager 에 보고하지 않습니다.

## 생성되는 메트릭

이러한 플러그 인에 의해 생성되는 메트릭은 다음 각 항목에서 자세히 설명합니다.

- Apache 로그 메트릭
- WebSphere 로그 메트릭
- WebLogic 로그 메트릭

## 명령줄 인수

이러한 각 플러그 인에서는 다음과 같은 네 가지 선택적 인수를 사용합니다.

### **logfileDir**

로그 파일이 있는 디렉터리

### **sleepTime**

로그 파일에 새 메시지가 있는지 확인하는 지연 간격(초)

### **logfile**

생성되는 로그 파일의 이름에 해당하는 정규식

### **subResource**

메트릭 이름의 하위 리소스 부분

#### **예제:**

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

## httpSvcAvailability.pl

이 플러그 인은 지정된 URL의 HTTP 서비스 가용성 및 성능을 바탕으로 메트릭을 보고합니다.

## 생성되는 메트릭

*httpSvcAvailability.pl* 플러그 인에 의해 생성되는 메트릭은 HTTP/HTTPS 서비스 가용성 메트릭에서 자세히 설명합니다.

## 명령줄 인수

이 플러그인에서는 다음과 같은 세 가지 선택적 명령줄 인수를 사용합니다.

### url

쉼표로 구분한 URL 집합 및 선택적 포트 번호. URL은 포트 번호(기본값과 다른 경우) 및 HTTP 와 HTTPS 를 구분하는 http: 또는 https:를 포함할 수 있습니다.

### get

서버에 대해 생성할 HTTP 요청의 쉼표로 구분된 목록(공백 없음)을 요청합니다.

일부 웹 사이트에서 HEAD 요청을 허용하지 않거나 일부 응용 프로그램 서버에서 GET 을 지원하지 않을 수 있습니다.

GET 또는 HEAD 중 적어도 하나는 지정해야 합니다.

GET 과 HEAD 를 모두 지정하면 HEAD 가 먼저

실행되고 GET 이 다음에 실행됩니다.

**기본값:** -request HEAD

### metric

쉼표로 구분된 반환할 메트릭 목록

- av = 가용성/연결할 수 있음(0=아니요, 1=예)
- rt = 연결에 대한 왕복 응답 시간
- tp = 처리량(KB. 반환된 KB/왕복 응답)
- all = 모든 통계

**기본값:** all

### 예제:

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

## Perl 지원 문제

*httpSvcAvailability.pl* 플러그 인(HTTP)의 경우:

*httpSvcAvailability.pl* 플러그 인(HTTP)을 사용하려면 기본 http 서비스 확인을 위해 Perl 라이브러리 libwww-perl(v5.65 이상)이 설치되어 있어야 합니다. www.activestate.com 에서는 libwww-perl 을 설치의 일부로 포함하는 Perl v5.6.1 을 제공합니다.

*httpSvcAvailability.pl* 플러그 인(HTTPS)의 경우:

- Windows에서 https 서비스 확인을 위해서는 Crypt-SSLeay(v0.17.1 이상)가 설치되어 있어야 합니다.
- Solaris에서 https 서비스 확인을 위해서는 Net-SSLeay(v1.05 이상)가 설치되어 있어야 합니다. 둘 다 ActiveState 의 Perl에 포함된 ppm Package Manager 도구를 사용하면 가장 쉽게 설치할 수 있습니다.

## kstat.pl

이 플러그 인은 Solaris의 Kstat 유틸리티에서 사용할 수 있는 커널 통계를 바탕으로 메트릭을 보고합니다.

## 생성되는 메트릭

*kstat.pl* 플러그 인에 의해 생성되는 메트릭은 Solaris 커널 통계(Kstat) 메트릭에서 자세히 설명합니다.

## 명령줄 인수

이 플러그 인에서는 다음과 같은 네 가지 명령줄 인수를 사용합니다.

### class

kstat 클래스와 일치시킬 정규식

### module

kstat 모듈과 일치시킬 정규식

## 이름

kstat 이름과 일치시킬 정규식

### statistic

kstat 통계와 일치시킬 정규식

**참고:** 이러한 인수를 사용하여 결과를 좁힐 수 있습니다. 명령줄 스위치를 지정하지 않으면 플러그인은 클래스, 모듈, 이름 및 통계에 대한 모든 Solaris 커널 정보를 반환합니다.

### 예제:

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

Introscope 메트릭은 노드 VM, 리소스 UNIX 및 segmap 을 차례로 생성한 다음 "free"를 확인한 후 값을 확인합니다.

**참고:** kstat에서 리소스 이름으로 null 값을 보고하는 경우 메트릭이 생성되지 않습니다.

## Perl 지원 문제

최상의 성능을 얻으려면 Solaris 제공 Perl(일반적으로 /usr/perl5 에 설치된 Perl 라이브러리에 의해 /usr/bin/perl에 있음)을 사용하여 kstat.pl 플러그인을 실행하는 것이 좋습니다. 이렇게 하면 포함된 Perl 라이브러리를 통해 Solaris 커널 하위 시스템에 직접 연결할 수 있는 인터페이스가 지원됩니다. 다른 Perl 설치를 사용하여 kstat.pl 플러그인을 실행하는 경우 kstat.pl 명령줄 유ти리티가 사용되므로 오버헤드가 약간 커질 수 있습니다.

## iisLogReader.pl

이 플러그인은 iisLogReader.cfg 파일의 구성을 기반으로 메트릭을 보고합니다.

**참고:** 이 로그 판독기 플러그인을 사용할 경우 로그 파일에 새 항목이 추가될 때까지 EAgent 가 Enterprise Manager로 보고하지 않습니다.

## 생성되는 메트릭

*iisLogReader.pl* 플러그인에 의해 생성되는 메트릭은 IIS 로그 판독기 메트릭에서 자세히 설명합니다.

## 명령줄 인수

이 플러그인에서는 다음과 같은 네 가지 선택적 명령줄 인수를 사용합니다.

### **logfileDir**

로그 파일이 있는 디렉터리

### **sleepTime**

로그 파일에 새 메시지가 있는지 확인하는 지연 간격(초)

### **logfile**

생성되는 로그 파일의 이름에 해당하는 정규식

### **subResource**

메트릭 이름의 하위 리소스 부분

### **예제:**

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

## iisLogReader.cfg 구성 파일

*iisLogReader.cfg* 파일은 *iisLogReader.pl* 파일의 명령줄 인수에 대한 값을 포함하고 있습니다.

웹 서버 응용 프로그램 로그 파일 파서마다 대기 지연, 하위 리소스 이름, 리소스 이름, 로그 파일 디렉토리 및 로그 파일 이름 패턴에 대한 기본값을 포함하는 자체의 구성 파일이 있습니다. 구성 파일에는 일치시키고 보고할 메트릭 목록도 포함됩니다. 각 메트릭 구성에는 **match**(정규식), **metricType**, **metricName** 및 **metricValue** 가 있습니다.

## LOGFILEDIR

연관된 웹 서버에 대한 로그 파일을 포함하는 전체 디렉터리 경로 이름입니다.

**참고:** Windows 경로의 유효한 경로 구분 기호는 슬래시입니다(백슬래시도 사용할 수 있음).

### 기본값

C:/WINNT/system32/LogFiles/W3SVC1

## LOGFILE

읽을 로그 파일의 이름입니다. 시간에 따라 변경되는 경우(예: 순환하는 로그 파일) 정규식이어야 합니다. 대/소문자를 구분하여 일치시킵니다.

**참고:** 파일의 전체 이름(경로 제외)과 일치해야 합니다.

**참고:** IIS 가 *iisLogReader.pl* 플러그인에서 지원하는 유일한 형식인 W3C 확장 로그 파일 형식으로 구성된 경우 기본값은 로그 파일 이름과 일치하도록 설정됩니다.

### 기본값

ex(tend)?\d+\.\log

## SLEEPTIME

로그 파일에 새 메시지가 있는지 확인하는 지연 간격(초)입니다.

카운터 메트릭은 "간격당"으로 정의되므로(메트릭 테이블 참조)  
일반적으로 SLEEPTIME 은 Enterprise Manager 시간 조각(즉, 15 초)보다  
크거나 같게 설정해야 합니다. 그렇지 않으면 카운터 메트릭이  
부정확해집니다.

### 기본값

15

## RESOURCE

Introscope 메트릭 이름의 리소스 부분입니다. Workstation Investigator에서 에이전트 노드 아래의 첫 번째 노드에 해당합니다.

기본값

IISWebServer

## SUBRESOURCE

Introscope 메트릭 이름의 하위 리소스 부분입니다. 위에 지정된 리소스 노드 아래의 하위 노드에 해당합니다.

기본값

<empty>

## LINE\_FORMAT

IIS 웹 서버에 의해 기록되는 정확한 행 형식을 정의합니다.

행 형식의 하위 속성은 LINE\_FORMAT 하위 속성 테이블에서 설명합니다.

기본값

해당 없음

## LINE\_FORMAT 하위 속성

다음 테이블에서는 LINE\_FORMAT 속성과 모든 하위 속성에 대해 설명합니다.

LINE\_FORMAT에 대한 다음 속성은 IIS 웹 서버에 의해 로그 파일에 기록되는 필드와 정확히 일치해야 합니다. 속성이 일치하지 않으면 iisLogReader.pl 플러그인은 일부 또는 전체 메트릭을 보고하지 못합니다.

IIS 웹 서버의 정확한 필드는 IIS 관리 도구(또는 이전 IIS 버전의 웹 페이지) 내에서 구성합니다. [IIS4.0 및 IIS5.0에서 IIS 웹 서버 로깅에 대한 로깅 필드를 구성](#)(페이지 51)할 수 있습니다.

기본적으로 IIS 웹 서버는 다음 필드만 기록합니다:

- 시간
- ClientIPAddress
- Method
- UriStem
- ProtocolStatus

일치시키기 위해 기본 iisLogReader.cfg 파일에서도 이러한 다섯 가지 LINE\_FORMAT 하위 속성만 사용할 수 있습니다.

설정 변경 없는 원래 상태의 iisLogReader.pl 플러그인은 기본 IIS 웹 서버 로깅 구성을 기준으로 합니다. 이 플러그인은 전송된 바이트를 제외한 모든 메트릭을 보고할 수 있습니다. 전송된 바이트를 보고하려면 IIS 웹 서버가 BytesSent 데이터 필드를 기록해야 하며 iisLogReader.cfg 하위 속성인 BytesSent의 주석 처리를 제거하여 활성화해야 합니다. 사용자가 IIS 웹 서버 및 iisLogReader.cfg에서 BytesSent를 활성화하지 못하면 CA Introscope®에는 전송된 바이트 메트릭이 보고되지 않습니다.

하위 속성 행의 주석 처리를 제거하는 경우 iisLogReader.pl 플러그인은 IIS 웹 서버 로그 파일(LOGFILEDIR 및 LOGFILE 속성으로 지정)에서 해당 필드를 찾습니다. 반대로 하위 속성 행을 주석으로 처리하는 경우 iisLogReader.pl 플러그인은 로그 파일에서 필드를 찾지 않습니다. 하위 속성 행의 순서는 IIS 웹 서버가 데이터를 기록하는 순서인 왼쪽에서 오른쪽으로의 순서와 이미 일치되어 있습니다(IIS 웹 서버에서 필드를 사용하도록 설정한 경우). 이 순서는 변경하면 안 됩니다.

## DELIMITER

연속 공백(공백, 탭)입니다.

## 기본값

\s+

## FIELDS

해당 없음

기본값

해당 없음

## FIELDS->Date

동작이 수행된 날짜입니다. YYYY-MM-DD.

기본값

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

## FIELDS->Time

기본적으로 사용하도록 설정됩니다.

동작이 수행된 시간입니다. 24HH:MM:SS GMT.

기본값

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

## FIELDS->ClientIPAddress

기본적으로 사용하도록 설정됩니다.

서버에 액세스한 클라이언트의 IP 주소입니다. 마침표로 구분한 숫자 네 개입니다.

기본값

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

## FIELDS->UserName

서버에 액세스한 사용자의 이름입니다. 하이픈(-), 백슬래시 및 영숫자를 임의로 조합한 값입니다.

기본값

[-\\w]+

**FIELDS->ServiceName**

클라이언트 컴퓨터에서 실행 중이던 인터넷 서비스입니다. 영숫자를 임의로 조합한 값입니다.

**기본값**

\w+

**FIELDS->ServerName**

로그 항목이 생성된 서버의 이름입니다. 마침표, 하이픈, 영숫자를 임의로 조합한 값입니다.

**기본값**

[.-\w]+

**FIELDS->ServerIPAddress**

로그 항목이 생성된 서버의 IP 주소입니다. 마침표로 구분한 숫자 네 개입니다.

**기본값**

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

**FIELDS->ServerPort**

클라이언트가 연결된 포트 번호입니다. 하나 이상의 숫자입니다.

**기본값**

\d+

**FIELDS->Method**

기본적으로 사용하도록 설정됩니다.

클라이언트가 수행하려고 한 작업입니다(예:GET 명령).

**기본값**

GET|HEAD|PUT|POST|DELETE

## FIELDS->URIStem

기본적으로 사용하도록 설정됩니다.

액세스한 리소스(예: HTML 페이지, CGI 프로그램 또는 스크립트)입니다.  
공백 아닌 문자의 시퀀스입니다.

기본값

\S+

## FIELDS->URIQuery

클라이언트가 수행하려고 한 쿼리(있는 경우), 즉 클라이언트가 일치 항목을  
찾던 하나 이상의 검색 문자열입니다. 공백 아닌 문자의 시퀀스입니다.

기본값

\S+

## FIELDS->ProtocolStatus

기본적으로 사용하도록 설정됩니다.

HTTP 용어로 나타낸 작업의 상태입니다.

기본값

\d+

## FIELDS->Win32Status

Windows에서 사용하는 용어로 나타낸 작업의 상태입니다. 하나 이상의  
숫자입니다.

기본값

\d+

**FIELDS->BytesSent**

서버에서 전송한 바이트 수입니다. 하나 이상의 숫자입니다.

**참고:** IIS에서 이 필드는 기본적으로 사용할 수 없습니다. "전송된 바이트" 메트릭을 수집하려면 이 필드를 설정해야 합니다.

**기본값**

\d+

**FIELDS->BytesReceived**

서버가 수신한 바이트 수입니다. 하나 이상의 숫자입니다.

**기본값**

\d+

**FIELDS->TimeTaken**

작업에 걸린 시간(밀리초)입니다. 하나 이상의 숫자입니다.

**기본값**

\d+

**FIELDS->ProtocolVersion**

클라이언트에서 사용되는 프로토콜(HTTP, FTP) 버전입니다. HTTP인 경우 HTTP 1.0 또는 HTTP 1.1입니다. 영숫자, 슬래시 및 마침표로 구성되는 시퀀스입니다.

**기본값**

[\\w/.]+

**FIELDS->Host**

클라이언트의 호스트입니다. 영숫자, 마침표, 콜론으로 구성되는 시퀀스입니다.

**기본값**

[\\w.:]+

## FIELDS->UserAgent

클라이언트에서 사용되는 브라우저입니다. 공백 아닌 문자의 시퀀스입니다.

기본값

\S+

## FIELDS->Cookie

보내거나 받은 쿠키의 콘텐츠입니다. 공백 아닌 문자의 시퀀스입니다.

기본값

\S+

## FIELDS->Referer

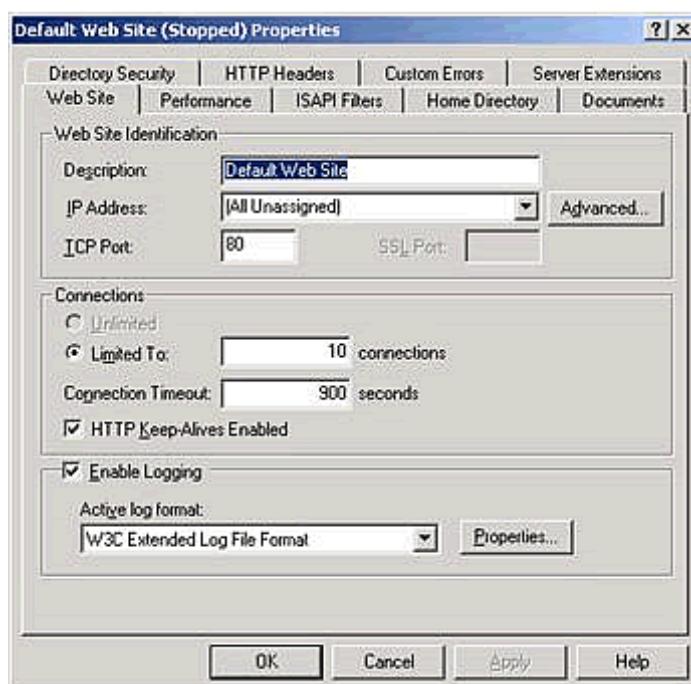
현재 사이트로 사용자를 리디렉션한 사이트입니다. 공백 아닌 문자의 시퀀스입니다.

기본값

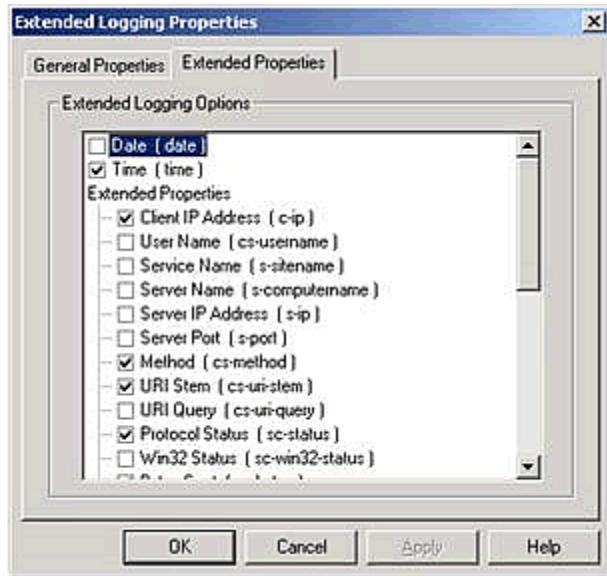
\S+

## IIS4.0 과 IIS5.0 에 로그인하는 IIS 웹 서버 구성

1. "인터넷 서비스 관리자"를 시작합니다.
2. 왼쪽에서 컴퓨터의 트리 노드를 확장합니다.  
다음과 같은 세 개의 트리 노드가 나타납니다.
  - FTP 사이트
  - 웹 사이트
  - SMTP 가상 서버
3. "웹 사이트" 노드를 마우스 오른쪽 단추로 클릭하고 "속성"을 선택합니다.  
"속성" 대화 상자가 나타납니다.



4. "웹 사이트" 탭을 선택하고 다음과 같이 설정합니다.
  - "로깅 사용" 확인란을 선택합니다.
  - "활성 로그 형식" 드롭다운 목록 상자에서 "W3C 확장 로그 파일 형식"을 선택합니다.
5. "속성"을 클릭합니다.  
"확장된 로깅 속성" 대화 상자가 나타납니다.



6. "일반 속성" 탭에서 "로그 파일 디렉터리"를 *iisLogReader.cfg* 파일에 있는 LOGFILEDIR 속성과 일치하도록 지정합니다.
7. "확장 속성" 탭에서 다음 필드를 반드시 선택해야 *iisLogReader.pl* 플러그인이 모든 메트릭을 보고할 수 있습니다.
  - 시간
  - 클라이언트 IP 주소
  - Method
  - URIStem
  - 프로토콜 상태
  - 보낸 바이트 수(전송된 바이트 확인을 위해 필요함)

여기서 선택한 모든 필드는 *iisLogReader.cfg* 파일의 LINE\_FORMAT->FIELDS 섹션 아래에서 주석 처리가 제거되어 사용 가능해진 필드와 일치해야 합니다. *iisLogReader.cfg* 파일의 이 부분을 통해 *iisLogReader.pl* 플러그인은 로그 파일에 들어가게 될 정확한 행 형식을 알게 됩니다. 만일 불일치가 발생하면 *iisLogReader.pl* 플러그인은 메트릭 보고에 실패하거나 올바르지 않은 메트릭을 보고하게 됩니다.

# 제 3 장: EAgent 기본 플러그 인 메트릭

---

이 장에서는 EAgent에 포함된 기본 플러그 인에 의해 생성되는 메트릭에 대해 설명합니다.

모든 메트릭은 시스템 클록에 설정된 시간을 사용하여 계산됩니다.  
트랜잭션 도중 시스템 클록이 재설정되면 해당 트랜잭션에 대해 보고된  
경과 시간이 잘못될 수 있습니다.

이 섹션은 다음 항목을 포함하고 있습니다.

[잘못된 문자 바꾸기](#) (페이지 53)

[IIS 로그 판독기 메트릭](#) (페이지 54)

[HTTP/HTTPS 서비스 가용성 메트릭](#) (페이지 59)

[Solaris 커널 통계\(Kstat\) 메트릭](#) (페이지 60)

[프로세스 가용성 메트릭](#) (페이지 61)

[디스크 통계\(Windows\) 메트릭](#) (페이지 61)

[디스크 통계\(Solaris\) 메트릭](#) (페이지 66)

[Apache 로그 메트릭](#) (페이지 70)

[WebSphere 로그 메트릭](#) (페이지 75)

[WebLogic 로그 메트릭](#) (페이지 82)

## 잘못된 문자 바꾸기

모든 리소스, 하위 리소스 및 메트릭 이름에서 Introscope 메트릭 이름에 허용되지 않는 잘못된 문자가 있는지 확인합니다. 특히 파이프 문자(|) 및 콜론(:)은 예약된 문자입니다. 사용자 또는 시스템이 생성한 이름에 이러한 문자가 포함되어 있으면 자동으로 이러한 문자가 다음과 같이 바뀝니다.

원래 문자	바뀌는 문자	설명
	-	파이프는 밑줄로 바뀜
:	^	콜론은 캐럿으로 바뀜

## IIS 로그 판독기 메트릭

다음 단원에서는 *iisLogReader.pl* 플러그인에 의해 생성되는 메트릭에 대해 설명합니다. 이러한 메트릭의 수는 플러그인이 로그 파일을 마지막으로 확인한 이후 간격당 개수이며 IISWebServer <Sub-Resource> 메트릭을 설명합니다.

### Total Page Hits(총 페이지 방문 횟수)

CA Introscope 데이터 유형

IntCounter

단위

1 (회)

기본값

켜짐

### Total Visitors Count(총 방문자 수)

CA Introscope 데이터 유형

IntCounter

단위

1 (회)

기본값

켜짐

## Page Hits For <URL>(<URL> 페이지 방문 횟수)

CA Introscope 데이터 유형

IntCounter

단위

1 (회)

기본값

켜짐

## Bytes Transferred(전송된 바이트)

CA Introscope 데이터 유형

IntCounter

단위

1 (MB)

기본값

켜짐

## ErrorCount\_400\_BadRequest

CA Introscope 데이터 유형

IntCounter

단위

1 (오류)

기본값

켜짐

## ErrorCount\_401\_Unauthorized

CA Introscope 데이터 유형

IntCounter

단위

1 (오류)

기본값

켜짐

## ErrorCount\_403\_Forbidden

CA Introscope 데이터 유형

IntCounter

단위

1 (오류)

기본값

켜짐

## ErrorCount\_404\_ServerConfiguration

CA Introscope 데이터 유형

IntCounter

단위

1 (오류)

기본값

켜짐

## ErrorCount\_500\_InternalServerError

CA Introscope 데이터 유형

IntCounter

단위

1 (오류)

기본값

켜짐

## ErrorMessage\_400\_BadRequest

CA Introscope 데이터 유형

StringEvent

단위

해당 없음

기본값

off

## ErrorMessage\_401\_Unauthorized

CA Introscope 데이터 유형

StringEvent

단위

해당 없음

기본값

off

### ErrorMessage\_403\_Forbidden

CA Introscope 데이터 유형

StringEvent

단위

해당 없음

기본값

off

### ErrorMessage\_404\_PageNotFound

CA Introscope 데이터 유형

StringEvent

단위

해당 없음

기본값

off

### ErrorMessage\_500\_InternalServerError

CA Introscope 데이터 유형

StringEvent

단위

해당 없음

기본값

off

## HTTP/HTTPS 서비스 가용성 메트릭

다음 단원에서는 *httpSvcAvailability.pl* 플러그인에 의해 생성되는 메트릭에 대해 설명하며 Remote Machine <hostnameOrIPAddr>|Service|HTTP<Sub-Resource> 메트릭을 설명합니다.

### 사용 가능

**CA Introscope** 데이터 유형

IntCounter

단위

0-1

기본값

켜짐

### 응답 시간

**CA Introscope** 데이터 유형

IntCounter

단위

초

기본값

켜짐

## Average Throughput(평균 처리량)

CA Introscope 데이터 유형

IntCounter

단위

KB/msec

기본값

켜짐

## Solaris 커널 통계(Kstat) 메트릭

다음 메트릭은 *kstat.pl* 플러그 인에 의해 생성됩니다.

*kstat* 플러그 인에 의해 생성되는 특정 메트릭의 이름은 모니터링되는 Solaris 시스템, Solaris 이름 지정 개념 및 Environment Performance Agent 를 통한 *kstat.pl* 플러그 인의 명령줄 호출(*IntroscopeEPAgent.properties*에 구성)에 따라 달라집니다.

## Solaris 커널 GHOST|<module>|<name> 메트릭

다음 단원에서는 Solaris 커널 GHOST|<module>|<name> 메트릭에 대해 설명합니다.

### <stat>

CA Introscope 데이터 유형

IntCounter

단위

다양함

기본값

켜짐

## 프로세스 가용성 메트릭

이 메트릭은 *processAvailability.pl* 플러그인에 의해 생성됩니다.

다음 단원에서는 RunningProcesses <Sub-Resource> 메트릭을 설명합니다.

### 수

**CA Introscope** 데이터 유형

IntCounter

단위

1 (프로세스)

기본값

켜짐

## 디스크 통계(Windows) 메트릭

이러한 메트릭은 <*EPA\_Home*>/windows/diskStats.pl 플러그인에 의해 생성됩니다.

다음 단원에서는 RunningProcesses <Sub-Resource> 메트릭에 대해 설명합니다.

추가 정보:

[Disk Reads/sec\(초당 디스크에서 읽기 작업 횟수\)](#) (페이지 62)  
[Disk Writes/sec\(초당 디스크에서 쓰기 작업 횟수\)](#) (페이지 62)  
[Disk Read Bytes/sec\(초당 디스크에서 읽기 바이트 수\)](#) (페이지 63)  
[Disk Write Bytes/sec\(초당 디스크에서 쓰기 바이트 수\)](#) (페이지 63)  
[Current Disk Queue Length\(현재 디스크 큐 길이\)](#) (페이지 63)  
[Average Disk Read Time\(평균 디스크에서 읽기 시간\) \(ms\)](#) (페이지 64)  
[Average Disk Write Time\(평균 디스크에서 쓰기 시간\) \(ms\)](#) (페이지 64)  
[Average Disk Bytes/Read\(쓰기당 평균 디스크에 쓴 바이트 수\)](#) (페이지 64)  
[Average Disk Bytes/Write\(쓰기당 평균 디스크에 쓴 바이트 수\)](#) (페이지 65)  
[사용 가능한 공간\(mb\)](#) (페이지 65)  
[사용 가능한 공간\(%\)](#) (페이지 65)

## Disk Reads/sec(초당 디스크에서 읽기 작업 횟수)

CA Introscope 데이터 유형

IntCounter

단위

1 (번)

기본값

켜짐

## Disk Writes/sec(초당 디스크에서 쓰기 작업 횟수)

CA Introscope 데이터 유형

IntCounter

단위

1 (번)

기본값

켜짐

## Disk Read Bytes/sec(초당 디스크에서 읽기 바이트 수)

CA Introscope 데이터 유형

IntCounter

단위

1 바이트

기본값

켜짐

## Disk Write Bytes/sec(초당 디스크에서 쓰기 바이트 수)

CA Introscope 데이터 유형

IntCounter

단위

1 바이트

기본값

켜짐

## Current Disk Queue Length(현재 디스크 큐 길이)

CA Introscope 데이터 유형

IntCounter

단위

1 (요청)

기본값

켜짐

## Average Disk Read Time(평균 디스크에서 읽기 시간) (ms)

CA Introscope 데이터 유형

IntCounter

단위

1 ms

기본값

켜짐

## Average Disk Write Time(평균 디스크에서 쓰기 시간) (ms)

CA Introscope 데이터 유형

IntCounter

단위

1 ms

기본값

켜짐

## Average Disk Bytes/Read(쓰기당 평균 디스크에 쓴 바이트 수)

CA Introscope 데이터 유형

IntCounter

단위

1 바이트

기본값

켜짐

## Average Disk Bytes/Write(쓰기당 평균 디스크에 쓴 바이트 수)

CA Introscope 데이터 유형

IntCounter

단위

1 바이트

기본값

켜짐

## 사용 가능한 공간(mb)

CA Introscope 데이터 유형

IntCounter

단위

1 mb

기본값

켜짐

## 사용 가능한 공간(%)

CA Introscope 데이터 유형

IntCounter

단위

1 (프로세스)

기본값

켜짐

## 디스크 통계(Solaris) 메트릭

다음 표에서는 <EPAgent\_Home>/solaris/diskStats.pl 플러그 인에 의해 생성되는 메트릭을 보여 줍니다.

다음 단원에서는 Disk <Sub-Resource> 메트릭을 설명합니다.

추가 정보:

[Disk Time\(디스크 시간\) \(%\)](#) (페이지 66)

[Transaction Waiting Time\(트랜잭션 대기 시간\) \(%\)](#) (페이지 67)

[Average # Transaction Active\(평균 활성 트랜잭션 수\)](#) (페이지 67)

[Average # Transaction Waiting\(평균 트랜잭션 대기 수\)](#) (페이지 67)

[Disk Reads/sec\(초당 디스크에서 읽기 작업 횟수\)](#) (페이지 68)

[Disk Writes/sec\(초당 디스크에서 쓰기 작업 횟수\)](#) (페이지 68)

[KB Read per Second\(초당 읽은 KB\)](#) (페이지 68)

[KB Written per Second\(초당 쓴 KB\)](#) (페이지 69)

[Average Service Time in Waiting Queue\(대기 큐에서의 평균 서비스 시간\) \(ms\)](#)

(페이지 69)

[Average Service Time for Active Transactions\(활성 트랜잭션을 위한 평균](#)

[서비스 시간\) \(ms\)](#) (페이지 69)

[사용 가능한 공간\(mb\)](#) (페이지 70)

[사용 가능한 공간\(%\)](#) (페이지 70)

### Disk Time(디스크 시간) (%)

CA Introscope 데이터 유형

IntCounter

단위

1 퍼센트

기본값

켜짐

## Transaction Waiting Time(트랜잭션 대기 시간) (%)

CA Introscope 데이터 유형

IntCounter

단위

1 퍼센트

기본값

켜짐

## Average # Transaction Active(평균 활성 트랜잭션 수)

CA Introscope 데이터 유형

IntCounter

단위

1 트랜잭션

기본값

켜짐

## Average # Transaction Waiting(평균 트랜잭션 대기 수)

CA Introscope 데이터 유형

IntCounter

단위

1 트랜잭션

기본값

켜짐

## Disk Reads/sec(초당 디스크에서 읽기 작업 횟수)

CA Introscope 데이터 유형

IntCounter

단위

1 (번)

기본값

켜짐

## Disk Writes/sec(초당 디스크에서 쓰기 작업 횟수)

CA Introscope 데이터 유형

IntCounter

단위

1 (번)

기본값

켜짐

## KB Read per Second(초당 읽은 KB)

CA Introscope 데이터 유형

IntCounter

단위

1 바이트

기본값

켜짐

## KB Written per Second(초당 쓴 KB)

CA Introscope 데이터 유형

IntCounter

단위

1 바이트

기본값

켜짐

## Average Service Time in Waiting Queue(대기 큐에서의 평균 서비스 시간) (ms)

CA Introscope 데이터 유형

IntCounter

단위

1 ms

기본값

켜짐

## Average Service Time for Active Transactions(활성 트랜잭션을 위한 평균 서비스 시간) (ms)

CA Introscope 데이터 유형

IntCounter

단위

1 ms

기본값

켜짐

## 사용 가능한 공간(mb)

CA Introscope 데이터 유형

IntCounter

단위

1 mb

기본값

켜짐

## 사용 가능한 공간(%)

CA Introscope 데이터 유형

IntCounter

단위

1 퍼센트

기본값

켜짐

## Apache 로그 메트릭

이 표는 *apacheLogReader.pl* 플러그인에 의해 생성되는 메트릭을 보여줍니다.

다음 단원에서는 ApacheWebServer <Sub-Resource> 메트릭에 대해 설명합니다.

추가 정보:

[PageHitCount](#) (페이지 71)  
[ErrorCount\\_400\\_BadRequest](#) (페이지 71)  
[ErrorCount\\_401\\_Unauthorized](#) (페이지 72)  
[ErrorCount\\_403\\_Forbidden](#) (페이지 72)  
[ErrorCount\\_404\\_PageNotFound](#) (페이지 72)  
[ErrorCount\\_500\\_ServerConfiguration](#) (페이지 73)  
[ErrorMessage\\_400\\_BadRequest](#) (페이지 73)  
[ErrorMessage\\_401\\_Unauthorized](#) (페이지 73)  
[ErrorMessage\\_403\\_Forbidden](#) (페이지 74)  
[ErrorMessage\\_404\\_PageNotFound](#) (페이지 74)  
[ErrorMessage\\_500\\_ServerConfiguration](#) (페이지 74)

## PageHitCount

CA Introscope 데이터 유형

IntCounter

단위

1 (회)

기본값

켜짐

## ErrorCount\_400\_BadRequest

CA Introscope 데이터 유형

IntCounter

단위

1 (오류)

기본값

켜짐

### ErrorCount\_401\_Unauthorized

CA Introscope 데이터 유형

IntCounter

단위

1 (오류)

기본값

켜짐

### ErrorCount\_403\_Forbidden

CA Introscope 데이터 유형

IntCounter

단위

1 (오류)

기본값

켜짐

### ErrorCount\_404\_PageNotFound

CA Introscope 데이터 유형

IntCounter

단위

1 (오류)

기본값

켜짐

## ErrorCount\_500\_ServerConfiguration

CA Introscope 데이터 유형

IntCounter

단위

1 (오류)

기본값

켜짐

## ErrorMessage\_400\_BadRequest

CA Introscope 데이터 유형

StringEvent

단위

1 (오류)

기본값

켜짐

## ErrorMessage\_401\_Unauthorized

CA Introscope 데이터 유형

StringEvent

단위

1 (오류)

기본값

켜짐

## ErrorMessage\_403\_Forgotten

CA Introscope 데이터 유형

StringEvent

단위

1 (오류)

기본값

켜짐

## ErrorMessage\_404\_PageNotFound

CA Introscope 데이터 유형

StringEvent

단위

1 (오류)

기본값

켜짐

## ErrorMessage\_500\_ServerConfiguration

CA Introscope 데이터 유형

StringEvent

단위

1 (오류)

기본값

켜짐

## WebSphere 로그 메트릭

이 메트릭은 `websphereLogReader.pl` 플러그인에 의해 생성됩니다.

다음 단원에서는 WebSphereApplicationServer <Sub-Resource> 메트릭에 대해 설명합니다.

추가 정보:

[MessageCount\\_Audit](#) (페이지 75)

[MessageCount\\_Warning](#) (페이지 76)

[MessageCount\\_EventDebug](#) (페이지 76)

[MessageCount\\_Error](#) (페이지 76)

[MessageCount\\_Fatal](#) (페이지 77)

[MessageText\\_Audit](#) (페이지 77)

[MessageText\\_Warning](#) (페이지 77)

[MessageText\\_EventDebug](#) (페이지 78)

[MessageText\\_Error](#) (페이지 78)

[MessageText\\_Fatal](#) (페이지 78)

### MessageCount\_Audit

CA Introscope 데이터 유형

IntCounter

단위

1 (회)

기본값

켜짐

## MessageCount\_Warning

CA Introscope 데이터 유형

IntCounter

단위

1 (회)

기본값

켜짐

## MessageCount\_EventDebug

CA Introscope 데이터 유형

IntCounter

단위

1 (회)

기본값

켜짐

## MessageCount\_Error

CA Introscope 데이터 유형

IntCounter

단위

1 (회)

기본값

켜짐

## MessageCount\_Fatal

CA Introscope 데이터 유형

IntCounter

단위

1 (회)

기본값

켜짐

## MessageText\_Audit

CA Introscope 데이터 유형

IntCounter

단위

1 (회)

기본값

켜짐

## MessageText\_Warning

CA Introscope 데이터 유형

IntCounter

단위

1 (회)

기본값

켜짐

## MessageText\_EventDebug

CA Introscope 데이터 유형

IntCounter

단위

1 (회)

기본값

켜짐

## MessageText\_Error

CA Introscope 데이터 유형

IntCounter

단위

1 (회)

기본값

켜짐

## MessageText\_Fatal

CA Introscope 데이터 유형

IntCounter

단위

1 (회)

기본값

켜짐

**CA Introscope** 데이터 유형

StringEvent

단위

해당 없음

기본값

켜짐

**CA Introscope** 데이터 유형

IntCounter

단위

1 (메시지)

기본값

켜짐

**CA Introscope** 데이터 유형

IntCounter

단위

1 (메시지)

기본값

켜짐

**CA Introscope** 데이터 유형

IntCounter

단위

1 (메시지)

기본값

켜짐

**CA Introscope** 데이터 유형

IntCounter

단위

1 (메시지)

기본값

켜짐

**CA Introscope** 데이터 유형

IntCounter

단위

1 (메시지)

기본값

켜짐

**CA Introscope** 데이터 유형

StringEvent

단위

해당 없음

기본값

off

**CA Introscope 데이터 유형**

StringEvent

단위

해당 없음

기본값

off

**CA Introscope 데이터 유형**

StringEvent

단위

해당 없음

기본값

off

**CA Introscope 데이터 유형**

StringEvent

단위

해당 없음

기본값

off

**CA Introscope 데이터 유형**

StringEvent

단위

해당 없음

기본값

off

## WebLogic 로그 메트릭

이 표는 `weblogicLogReader.pl` 플러그인에 의해 생성되는 메트릭을 보여줍니다.

다음 단원에서는 WebLogicApplicationServer <Sub-Resource> 메트릭에 대해 설명합니다.

추가 정보:

[ServerVersion](#) (페이지 83)  
[MessageCount\\_Warning](#) (페이지 83)  
[MessageCount\\_Error](#) (페이지 83)  
[MessageCount\\_Notify](#) (페이지 84)  
[MessageCount\\_Critical](#) (페이지 84)  
[MessageCount\\_Alert](#) (페이지 84)  
[MessageCount\\_Emergency](#) (페이지 85)  
[MessageText\\_Warning](#) (페이지 85)  
[MessageText\\_Error](#) (페이지 85)  
[MessageText\\_Notify](#) (페이지 86)  
[MessageText\\_Critical](#) (페이지 86)  
[MessageText\\_Alert](#) (페이지 86)  
[MessageText\\_Emergency](#) (페이지 87)

## ServerVersion

CA Introscope 데이터 유형

StringEvent

단위

해당 없음

기본값

켜짐

## MessageCount\_Warning

CA Introscope 데이터 유형

IntCounter

단위

1 (메시지)

기본값

켜짐

## MessageCount\_Error

CA Introscope 데이터 유형

IntCounter

단위

1 (메시지)

기본값

켜짐

## MessageCount\_Notify

CA Introscope 데이터 유형

IntCounter

단위

1 (메시지)

기본값

켜짐

## MessageCount\_Critical

CA Introscope 데이터 유형

IntCounter

단위

1 (메시지)

기본값

켜짐

## MessageCount\_Alert

CA Introscope 데이터 유형

IntCounter

단위

1 (메시지)

기본값

켜짐

## MessageCount\_Emergency

CA Introscope 데이터 유형

IntCounter

단위

1 (메시지)

기본값

켜짐

## MessageText\_Warning

CA Introscope 데이터 유형

StringEvent

단위

해당 없음

기본값

off

## MessageText\_Error

CA Introscope 데이터 유형

StringEvent

단위

해당 없음

기본값

off

## MessageText\_Notify

CA Introscope 데이터 유형

StringEvent

단위

해당 없음

기본값

off

## MessageText\_Critical

CA Introscope 데이터 유형

StringEvent

단위

해당 없음

기본값

off

## MessageText\_Alert

CA Introscope 데이터 유형

StringEvent

단위

해당 없음

기본값

off

## MessageText\_Emergency

CA Introscope 데이터 유형

StringEvent

단위

해당 없음

기본값

off



# 제 4 장: EPAGent 속성

---

이 부록에서는 Environment Performance Agent 를 사용자 지정하기 위해 사용하는 속성 파일 및 설정에 대해 자세히 설명합니다.

<EPAGent\_Home>/epagent/introscopeEPAGent.properties 에 있는 그 밖의 다른 정보를 위해 파일을 읽을 수도 있습니다.

**참고:** Windows 컴퓨터에서 이러한 파일에 경로를 추가할 경우 백슬래시(\)는 백슬래시를 하나 더 사용해 중첩하여 이스케이프해야 합니다(예: C:\\\\Introscope\\\\lib\\\\Agent.jar).

이 섹션은 다음 항목을 포함하고 있습니다.

[IntroscopeEPAGent.properties 파일](#) (페이지 89)

[에이전트 메트릭 만료 처리](#) (페이지 91)

[Case-Sensitive 메트릭](#) (페이지 95)

[ChangeDetector 구성](#) (페이지 96)

[Enterprise Manager 연결](#) (페이지 100)

[EPAGent 구성](#) (페이지 101)

[확장 디렉터리](#) (페이지 102)

[Logging\(로깅\)](#) (페이지 103)

[프로세스](#) (페이지 103)

[원격 구성](#) (페이지 105)

[상태 저장 플러그인](#) (페이지 106)

[상태 비저장 플러그인](#) (페이지 107)

## IntroscopeEPAGent.properties 파일

EPAGent 의 설정 유형은 CA Introscope 에이전트 프로필의 유형과 동일합니다.

## 에이전트 및 프로세스 이름 설정

두 속성 `introscope.agent.agentName` 및 `introscope.agent.customProcessName`은 EPAgent의 이름과 프로세스의 이름을 정의하며 둘 다 여러 에이전트 세션에서 단일 EPAgent에 대한 정보를 추적하기 위해 정규화된 에이전트 이름(*Domain / host / process / Agentname*)의 일부로 사용됩니다.

이러한 속성은 선택 사항입니다. 에이전트 이름을 정의하지 않으면 `UnknownAgent`로 간주됩니다.

### 프로세스 이름

CA Introscope는 90 다음 조건에 따라 EPAgent를 명명하며 표시되는 순서에 따라 우선 순위가 부여됩니다.

- `introscope.agent.customProcessName` 속성에 의해 정의된 이름
- 응용 프로그램의 주요 클래스 이름을 딴 이름
- `introscope.agent.defaultProcessName`에 의해 설정된 이름

추가 정보:

[프로세스](#) (페이지 103)

### 에이전트 이름

`introscope.agent.agentName` 속성에 따라 EPAgent의 이름이 결정됩니다.

CA Introscope®는 다음 조건에 따라 Environment Performance Agent를 명명하며 표시되는 순서에 따라 우선 순위가 부여됩니다.

- [introscope.agent.agentName](#) (페이지 105) 속성에 의해 정의된 이름.
- `introscope.agent.agentName`에서 고유한 이름이 지정되지 않으면 Environment Performance Agent는 `UnknownAgent`로 명명됩니다.

## 에이전트 메트릭 만료 처리

에이전트 메트릭 만료 처리는 사용되지 않는 메트릭을 주기적으로 제거합니다. 사용되지 않는 메트릭은 구성된 시간 동안 새 데이터를 보고하지 않은 메트릭입니다. 오래된 메트릭을 제거하면 에이전트 성능을 향상하고 발생할 수 있는 메트릭 급증을 방지할 수 있습니다.

**참고:** 시스템이 처리할 수 있는 것보다 많은 메트릭을 보고하도록 부주의하게 에이전트를 설정하는 경우 메트릭이 급증합니다. 너무 많은 메트릭이 보고되면 응용 프로그램 서버의 성능에 영향을 미칠 수 있고 심한 경우 서버가 전혀 작동하지 않게 될 수도 있습니다.

그룹에 속한 메트릭은 그룹의 모든 메트릭이 제거 후보에 해당하는 경우에만 제거됩니다. 현재 *BlamePointTracer* 및 *MetricRecordingAdministrator* 메트릭만 그룹으로 제거됩니다. 다른 메트릭은 개별적으로 제거됩니다.

*MetricRecordingAdministrator*에는 메트릭 그룹을 생성, 검색 또는 제거하기 위한 다음과 같은 인터페이스가 있습니다.

- `getAgent().IAgent_getMetricRecordingAdministrator.addMetricGroup`  
문자열 구성 요소, 수집 메트릭. 구성 요소 이름은 메트릭 그룹의 메트릭 리소스 이름입니다. 동일한 메트릭 노드 아래에 메트릭이 있어야만 그룹으로 간주됩니다. 메트릭은 `com.wily.introscope.spec.metric.AgentMetric` 데이터 구조의 모음입니다. 이 모음에는 `AgentMetric` 데이터 구조만 추가할 수 있습니다.
- `getAgent().IAgent_getMetricRecordingAdministrator.getMetricGroup`  
문자열 구성 요소. 메트릭 리소스 이름인 구성 요소 이름을 기준으로 메트릭의 모음을 가져올 수 있습니다.
- `getAgent().IAgent_getMetricRecordingAdministrator.removeMetricGroup`  
문자열 구성 요소. 메트릭 그룹은 메트릭 리소스 이름인 구성 요소를 기반으로 제거됩니다.
- `getAgent().IAgent_getDataAccumulatorFactory.isRemoved`  
메트릭이 제거되었는지 확인합니다. 확장에 있는 누산기의 인스턴스를 유지하려는 경우 이 인터페이스를 사용합니다. 메트릭 만료 처리로 인해 누산기가 제거되면 이 인터페이스를 통해 사용되지 않는 참조를 유지하지 않도록 합니다.

**중요!** 다른 CA Technologies 제품에서 사용하기 위한 용도 등으로 *MetricRecordingAdministrator* 인터페이스를 사용하는 확장을 생성하는 경우 사용자 고유의 누산기 인스턴스를 제거해야 합니다. 호출되지 않았다는 이유로 메트릭이 만료 처리되면 나중에 해당 메트릭의 데이터를 사용할 수 있게 되어도 이전 누산기 인스턴스가 새 메트릭 데이터 포인트를 생성하지 않습니다. 이러한 상황을 방지하려면 사용자 고유의 누산기 인스턴스를 삭제하지 말고 대신 *getDataAccumulatorFactory* 인스턴스를 사용하십시오.

## 에이전트 메트릭 만료 처리 구성

에이전트 메트릭 만료 처리는 기본적으로 사용됩니다. 이 기능을 해제하려면 *introscope.agent.metricAging.turnOn* (see page 93) 속성을 사용합니다. 이 속성을 *IntroscopeAgent.profile*에서 제거하면 에이전트 메트릭 만료 처리가 기본적으로 해제됩니다.

에이전트 메트릭 만료 처리는 에이전트의 하트비트에 실행됩니다. 하트비트는 *introscope.agent.metricAging.heartbeatInterval* (see page 93) 속성을 사용하여 구성됩니다. 하트비트 빈도를 낮게 유지해야 합니다. 하트비트가 높으면 에이전트와 CA Introscope®의 성능에 영향을 미칩니다.

각 하트비트 동안 특정 메트릭 집합이 확인됩니다. 이 집합은 *introscope.agent.metricAging.dataChunk* (see page 94) 속성을 사용하여 구성할 수 있습니다. 이 값도 낮게 유지해야 합니다. 이 값이 높으면 성능에 영향을 줍니다. 기본값은 하트비트당 메트릭을 500 개 확인합니다. 메트릭 500 개 각각을 확인하여 제거 후보에 해당하는지 확인합니다. 예를 들어 이 속성을 하트비트당 메트릭 500 개 단위의 청크를 확인하도록 설정하는 경우 에이전트 메모리의 메트릭이 총 10,000 개 있으면 메트릭 10,000 개를 모두 확인하느라 성능이 저하되고 시간이 오래 걸립니다. 그러나 이 속성을 더 높은 값으로 설정하면 모든 10,000 개 메트릭을 더 빨리 확인하지만 오버헤드가 높아집니다.

특정 기간 후 메트릭에 새 데이터가 수신되지 않은 경우 메트릭은 제거 후보입니다. 이 기간은 *introscope.agent.metricAging.numberTimeslices* (see page 94) 속성을 사용하여 구성할 수 있습니다. 이 속성은 기본적으로 3000 으로 설정되어 있습니다. 메트릭이 제거 조건을 충족하면 해당 그룹의 모든 메트릭이 메트릭 제거 후보인지 확인합니다. 이 요구 사항도 충족하면 메트릭이 제거됩니다.

## introscope.agent.metricAging.turnOn

에이전트 메트릭 만료 처리를 설정하거나 해제합니다.

### 속성 설정

True 또는 False

### 기본값

True

### 예

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

### 참고

이 속성의 변경 사항은 즉시 적용되며 관리되는 응용 프로그램을 다시 시작할 필요가 없습니다.

## introscope.agent.metricAging.heartbeatInterval

제거할 메트릭을 확인하는 시간 간격을 초 단위로 지정합니다.

### 기본값

1800

### 예

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

### 참고

이 속성의 변경 내용을 적용하려면 관리되는 응용 프로그램을 다시 시작해야 합니다.

## introscope.agent.metricAging.dataChunk

각 간격에서 확인할 메트릭의 수를 지정합니다.

### 기본값

500

예

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

### 참고

이 속성의 변경 사항은 즉시 적용되며 관리되는 응용 프로그램을 다시 시작할 필요가 없습니다.

## introscope.agent.metricAging.numberTimeslices

이 속성은 데이터를 제거 대상으로 만들기 전에 새 데이터가 없는지 확인하는 간격의 수를 지정합니다.

### 기본값

3000

예

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

### 참고

이 속성에 대한 변경 내용은 즉시 적용되므로 관리되는 응용 프로그램을 다시 시작할 필요가 없습니다.

## introscope.agent.metricAging.metricExclude.ignore.0

지정한 메트릭을 제거하지 않도록 제외합니다. 만료 처리에서 하나 이상의 메트릭을 제외하려면 목록에 메트릭 이름이나 메트릭 필터를 추가합니다.

### 속성 설정

메트릭을 쉼표로 구분한 목록입니다. 메트릭 이름에서 별표(\*)를 와일드카드로 사용할 수 있습니다.

### 기본값

기본값은 Threads로 시작하는 메트릭 이름(*Threads\**)입니다.

#### 예

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

### 참고

이 속성의 변경 사항은 즉시 적용되며 관리되는 응용 프로그램을 다시 시작할 필요가 없습니다.

## Case-Sensitive 메트릭

다음 속성은 대/소문자 구분 여부를 구성합니다.

## introscope.epagent.metricscasesensitive

메트릭 이름의 대/소문자를 구분하도록 설정합니다.

### 속성 설정

True 또는 False

### 기본값

true

#### 예

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

### 참고

- 기본적으로 주석 처리됩니다.
- 이 속성을 `true`로 설정하면 메트릭의 대/소문자를 구분하게 됩니다.
- 이 속성을 `false`로 설정하면 대/소문자를 구분하지 않게 됩니다.

## ChangeDetector 구성

다음 속성은 EPAgent 와 ChangeDetector 의 상호 작용을 구성합니다.

### introscope.changeDetector.enable

ChangeDetector 사용 여부를 지정합니다. ChangeDetector 가 사용되도록 설정하려면 이 속성을 `true`로 설정합니다. 주석 처리되어 기본적으로 `false`로 설정되어 있습니다. ChangeDetector 가 사용되도록 설정한 경우 ChangeDetector 관련 추가 속성도 설정해야 합니다.

#### 속성 설정

True 또는 False

#### 기본값

False

#### 예

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

#### 참고

이 속성의 변경 내용을 적용하려면 관리되는 응용 프로그램을 다시 시작해야 합니다.

## introscope.changeDetector.rootDir

ChangeDetector 파일의 루트 디렉터리를 지정합니다. 루트 디렉터리는 ChangeDetector 가 해당 로컬 캐시 파일을 생성하는 폴더입니다.

### 속성 설정

ChangeDetector 파일의 루트 디렉터리에 대한 전체 경로를 나타내는 텍스트 문자열입니다.

### 기본값

기본 경로는 *c://sw//AppServer//wily//change\_detector* 입니다.

#### 예

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

### 참고

예제에서처럼 백슬래시를 사용하여 백슬래시 문자를 이스케이프하십시오.

## introscope.changeDetector.isengardStartupWaitTimeInSec

에이전트가 시작된 후 ChangeDetector 가 Enterprise Manager 연결을 시도하기 전에 대기하는 기간을 초 단위로 지정합니다. 이 속성은 기본적으로 주석으로 처리되어 있습니다.

### 기본값

기본값은 15 초입니다.

#### 예

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

## introscope.changeDetector.waitTimeBetweenReconnectInSec

Enterprise Manager 연결을 다시 시도하기 전에 ChangeDetector 가 대기하는 기간을 초 단위로 지정합니다. 이 속성은 기본적으로 주석으로 처리되어 있습니다.

### 기본값

기본값은 10 초입니다.

### 예

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

## introscope.changeDetector.enableEPA

ChangeDetector EAgent 를 사용하도록 설정하면 EAgent 플러그인을 XML 형식의 변경 데이터에 대한 데이터 원본으로 사용할 수 있습니다.

### 속성 설정

True 또는 False

### 기본값

true

### 예

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

### 참고

기본적으로 주석 처리됩니다.

## introscope.changeDetector.agentID

EPAgent 를 식별하기 위해 ChangeDetector 가 사용하는 텍스트 문자열을 지정합니다.

예

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

참고

기본적으로 주석 처리됩니다.

## introscope.changeDetector.profile

ChangeDetector 데이터 원본 구성 파일의 절대 또는 상대 경로를 지정합니다. 이 속성은 기본적으로 주석으로 처리되어 있습니다.

기본값

기본값은 ChangeDetector-config.xml 입니다.

예

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

참고

예제에서처럼 백슬래시를 사용하여 백슬래시 문자를 이스케이프하십시오.

## introscope.changeDetector.profileDir

데이터 원본 구성 파일이 들어 있는 디렉터리에 대한 절대 또는 상대 경로를 지정합니다. 이 속성이 설정된 경우 *introscope.changeDetector.profile* 속성으로 지정된 모든 파일과 함께 이 디렉터리의 모든 데이터 원본 구성 파일이 사용됩니다. 이 속성은 기본적으로 주석으로 처리되어 있습니다.

### 기본값

기본값은 `changeDetector_profiles`입니다.

예

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

### 참고

백슬래시를 사용하여 백슬래시 문자를 이스케이프 처리하십시오.

## Enterprise Manager 연결

다음 속성은 Enterprise Manager에 대한 연결을 구성합니다.

## introscope.agent.enterprisemanager.connectionorder

EPAgent와 기본 Enterprise Manager 간의 연결이 끊어질 경우 EPAgent에서 사용하는 백업 Enterprise Manager의 연결 순서를 지정합니다. 목록의 항목은 쉼표로 구분해야 합니다.

### 기본값

DEFAULT

예

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

## introscope.agent.enterpriseManager.transport.tcp.host.DEFAULT

에이전트가 기본적으로 연결되는 Enterprise Manager 를 실행하는 컴퓨터의 호스트 이름을 지정합니다.

### 기본값

localhost

### 예

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

### 참고

이 속성의 변경 내용을 적용하려면 관리되는 응용 프로그램을 다시 시작해야 합니다.

## EPAgent 구성

다음 속성은 EPAgent 자체를 구성합니다.

## introscope.epagent.config.networkDataPort

EPAgent 가 들어오는 XML 데이터에 대한 네트워크 연결을 수신 대기하는 포트를 지정합니다.

### 기본값

8000

### 예

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

### 참고

이 속성은 기본적으로 주석으로 처리되어 있습니다.

## introscope.epagent.config.httpServerPort

EPAgent 가 HTTP GET 명령을 수신 대기하는 포트를 지정합니다.

### 기본값

8080

### 예

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

### 참고

이 속성은 기본적으로 주석으로 처리되어 있습니다.

## 확장 디렉터리

다음 메트릭은 확장 디렉터리를 지정합니다.

## introscope.agent.extensions.directory

에이전트가 로드할 모든 확장의 위치를 지정합니다. 디렉터리의 절대 또는 상대 경로를 지정할 수 있습니다. 절대 경로를 지정하지 않을 경우, 지정한 값은 *IntroscopeAgent.profiles* 파일의 위치를 기준으로 확인됩니다.

### 기본값

기본 위치는 <EPAgent\_Home>/epagent/ext 디렉터리 내의 ext 디렉터리입니다.

### 예

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

### 참고

이 속성의 변경 내용을 적용하려면 관리되는 응용 프로그램을 다시 시작해야 합니다.

## Logging(로깅)

다음과 같은 속성이 로깅을 구성합니다.

### log4j.logger.EPAGent

EPAGent 로그의 로깅 세부 정보 양과 로그 파일의 출력 위치를 지정합니다.

#### 기본값

INFO, console

#### 예

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

#### 참고

- 로그 파일의 세부 정보 수준을 높이려면 *INFO* 를 *VERBOSE#com.wily.util.feedback.Log4JSeverityLevel* 로 바꾸십시오.
- 출력을 콘솔 대신 로그 파일에 보내려면 *console* 을 *logfile* 로 바꿉니다.

### log4j.appenders.logfile.File

이 속성은 에이전트 로깅에 대한 로깅 수준과 출력 위치를 제어합니다.

#### 기본값

IntroscopeEPA.log

#### 예

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

#### 참고

로그 파일 위치 및 이름에 대한 전체 경로 이름을 사용할 수 있습니다.

## 프로세스

다음과 같은 속성이 프로세스를 구성합니다.

## introscope.agent.customProcessName

모니터링되는 프로세스의 이름을 지정합니다.

### 기본값

EPAgentProcess

### 예

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

### 참고

응용 프로그램 서버별 설치 관리자를 위한 응용 프로그램 서버 이름으로 설정합니다.

## introscope.agent.defaultProcessName

*introscope.Agent.customProcessName*이 설정되어 있지 않고 응용 프로그램의 기본 클래스에서 이름을 유추할 수 없는 경우 에이전트에서 사용할 기본 이름을 지정합니다.

### 기본값

UnknownProcess

### 예

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

### 참고

설치 관리자에 따라 기본값이 다를 수 있습니다.

## introscope.agent.agentName

Enterprise Manager 및 Workstation에 나타나는 EPAgent 이름을 지정합니다.

### 기본값

EPAgent

#### 예

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

### 참고

UnknownAgent로 설정하거나 <appservername>Agent와 같이 응용 프로그램별 설치 관리자의 응용 프로그램 서버 이름으로 지정합니다.

## 원격 구성

다음 속성을 사용하여 EPAgent를 원격으로 구성할 수 있습니다.

## introscope.agent.remoteagentconfiguration.enabled

EPAgent의 원격 구성을 사용하거나 사용하지 않도록 설정합니다.

### 속성 설정

True 또는 False

### 기본값

False

#### 예

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

### 참고

이 속성의 변경 사항은 즉시 적용되며 관리되는 응용 프로그램을 다시 시작할 필요가 없습니다.

## 상태 저장 플러그 인

상태 저장 플러그 인은 다음 속성으로 구성됩니다.

### introscope.epagent.plugins.stateful.names

실행할 상태 저장 플러그 인의 이름을 지정합니다. 이 속성은 EPAgent 플러그 인을 사용하는 데 필요합니다.

#### 속성 설정

플러그 인 이름이 쉼표로 구분된 목록.

#### 기본값

APACHE,WEBLOGIC,WEBSPHHERE,IIS

#### 예

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

#### 참고

- 이 속성은 기본적으로 주석으로 처리되어 있습니다.
- 이러한 이름은 사용자 정의될 수 있습니다. 단, 이름은 다음 `introscope.epagent.plugins.stateful.<pluginname>.command` 속성에 사용된 이름과 일치해야 합니다.

### introscope.epagent.stateful.<pluginname>.command

`introscope.epagent.plugins.stateful.names` 속성에서 지정된 플러그 인을 실행할 명령을 지정합니다. 플러그 인 이름별로 속성이 한 행씩 지정되어야 합니다.

#### 예

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

#### 참고

이 속성은 기본적으로 주석으로 처리되어 있습니다.

## introscope.epagent.stateful.<pluginname>.class

플러그 인은 "상태 비저장"이거나 "상태 저장"입니다. 상태 비저장 플러그 인은 cron 같은 일정에 기반하거나 지연을 기반으로 하여 EPAGent 에 의해 주기적으로 실행됩니다. 상태 저장 플러그 인은 EPAGent 에 의해 일단 시작되며 계속 실행할 목적으로 사용됩니다. 상태 저장 플러그 인은 종료되면 다시 시작됩니다.

플러그 인은 "명령" 하위 키와 함께 지정된 외부 프로세스이거나 EPAGent 가상 컴퓨터에서 실행될 수 있는 Java 클래스일 수 있습니다. Java 클래스 플러그 인은 "클래스" 하위 키, 정규화된 클래스 이름 및 인수 옵션으로 지정되며 열거한 순서대로 구성됩니다. 예를 들면 *my.package.path.Class arg1 arg2 arg3* 과 같습니다.

### 예

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

### 참고

이 속성은 기본적으로 주석으로 처리되어 있습니다.

## 상태 비저장 플러그 인

상태 비저장 플러그 인은 다음 속성으로 구성됩니다.

## introscope.epagent.plugins.stateless.names

실행 할 상태 비저장 플러그 인의 이름을 지정합니다.

### 기본값

DISK,ORACLE,HTTP,KSTAT

### 예

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

### 참고

- 이 속성은 기본적으로 주석으로 처리되어 있습니다.
- 이러한 이름은 사용자 정의될 수 있습니다. 단, 이름은 다음 *introscope.epagent.plugins.stateless.<pluginname>.command* 속성에 사용된 이름과 일치해야 합니다.

## introscope.epagent.stateless.<pluginname>.command

*introscope.epagent.plugins.stateless.names* 속성에서 지정된 플러그 인을 실행 할 명령을 지정합니다. 플러그 인 이름별로 속성을 한 행씩 지정해야 합니다.

### 예

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

### 참고

이 속성은 기본적으로 주석으로 처리되어 있습니다.

## introscope.epagent.stateless.<pluginname>.class

EPAgent VM 내에서 상태 비저장 플러그 인으로 실행 할 클래스를 지정합니다.

### 참고

이 속성은 기본적으로 주석으로 처리되어 있습니다.

## introscope.epagent.stateless.<pluginname>.delayInSeconds

EPAgent 가 지정된 플러그 인을 실행하는 빈도를 초 단위로 지정합니다.  
플러그 인 이름별로 속성이 한 행씩 지정되어야 합니다.

### 기본값

900

### 예

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

### 참고

이 속성은 기본적으로 주석으로 처리되어 있습니다.

## introscope.epagent.stateless.<pluginname>.schedule

*introscope.epagent.stateless.<pluginname>.command* 속성에 지정된 플러그  
인을 실행할 일정을 지정합니다. 입력은 CronTab 일정 구문(UNIX 시스템의  
경우 "man 5 crontab")입니다.

### 참고

- 이 속성은 기본적으로 주석으로 처리되어 있습니다.
- 이 속성은 *introscope.epagent.stateless.<pluginname>.delayInSeconds* 속성  
대신 사용됩니다. 두 속성의 주석 처리를 모두 제거하면 EPAgent 가  
종료됩니다.

## **introscope.epagent.config.stalledStatelessPluginTimeoutInSeconds**

중단된 상태 비저장 플러그 인을 종료하기 전까지 대기할 시간을  
지정합니다.

**기본값**

60 초

**예**

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

**참고**

이 속성은 기본적으로 주석으로 처리되어 있습니다.

## introscope.epagent.stateless.<pluginname>.metricNotReportedAction

상태 비저장 플러그 인이 값을 보고하는 작업을 중단하는 경우 이 속성에 따라 후속 작업이 결정됩니다.

### 속성 설정

이 속성에 유효한 값은 다음과 같습니다.

- **same(동일함)** - 메트릭이 만료 처리될 때까지 동일한 값을 계속 보고합니다.
- **zero(0)** - 0 값으로 메트릭을 보고합니다. 메트릭 유형이 stringevent 이거나 timestamp 인 경우에는 'same'(동일함) 작업이 수행됩니다.
- **stop(중지)** - 메트릭 값 보고를 중지합니다.

### 기본값

same(동일함)

### 예

다음 MATCHEDPROCS 예에서는 processAvailability 플러그 인을 사용하여 일치 매개 변수로 지정한 정규식과 일치하는 이름의 실행 프로세스에 대해 1 값을 반환합니다. 즉, 이 경우에는 'f'로 시작하는 이름의 프로세스가 일치합니다. 프로세스가 더 이상 실행되지 않을 때 메트릭 값을 0 으로 다시 설정하기 위해 *metricNotReportedAction* 매개 변수 값 'zero'가 사용되었습니다.

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