

# CA Application Performance Management

Environment Performance Agent 実装ガイド

リリース 9.5



このドキュメント（組み込みヘルプシステムおよび電子的に配布される資料を含む、以下「本ドキュメント」）は、お客様への情報提供のみを目的としたもので、日本 CA 株式会社（以下「CA」）により随時、変更または撤回されることがあります。

CA の事前の書面による承諾を受けずに本ドキュメントの全部または一部を複製、譲渡、開示、変更、複製することはできません。本ドキュメントは、CA が知的財産権を有する機密情報です。ユーザは本ドキュメントを開示したり、  
(i) 本ドキュメントが関係する CA ソフトウェアの使用について CA とユーザとの間で別途締結される契約または (ii) CA とユーザとの間で別途締結される機密保持契約により許可された目的以外に、本ドキュメントを使用することはできません。

上記にかかわらず、本ドキュメントで言及されている CA ソフトウェア製品のライセンスを受けたユーザは、社内でユーザおよび従業員が使用する場合に限り、当該ソフトウェアに関連する本ドキュメントのコピーを妥当な部数だけ作成できます。ただし CA のすべての著作権表示およびその説明を当該複製に添付することを条件とします。

本ドキュメントを印刷するまたはコピーを作成する上記の権利は、当該ソフトウェアのライセンスが完全に有効となっている期間内に限定されます。いかなる理由であれ、上記のライセンスが終了した場合には、お客様は本ドキュメントの全部または一部と、それらを複製したコピーのすべてを破棄したことを、CA に文書で証明する責任を負いません。

準拠法により認められる限り、CA は本ドキュメントを現状有姿のまま提供し、商品性、特定の使用目的に対する適合性、他者の権利に対して侵害のないことについて、黙示の保証も含めいかなる保証もしません。また、本ドキュメントの使用に起因して、逸失利益、投資損失、業務の中断、営業権の喪失、情報の喪失等、いかなる損害（直接損害か間接損害かを問いません）が発生しても、CA はお客様または第三者に対し責任を負いません。CA がかかる損害の発生の可能性について事前に明示に通告されていた場合も同様とします。

本ドキュメントで参照されているすべてのソフトウェア製品の使用には、該当するライセンス契約が適用され、当該ライセンス契約はこの通知の条件によっていかなる変更も行われません。

本ドキュメントの制作者は CA です。

「制限された権利」のもとの提供: アメリカ合衆国政府が使用、複製、開示する場合は、FAR Sections 12.212、52.227-14 及び 52.227-19(c)(1)及び(2)、ならびに DFARS Section 252.227-7014(b)(3) または、これらの後継の条項に規定される該当する制限に従うものとします。

Copyright © 2013 CA. All rights reserved. 本書に記載された全ての製品名、サービス名、商号およびロゴは各社のそれぞれの商標またはサービスマークです。

## CA Technologies 製品リファレンス

このドキュメントは、以下の CA Technologies 製品および機能に関するものです。

- 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 への連絡先

テクニカルサポートの詳細については、弊社テクニカルサポートの Web サイト (<http://www.ca.com/jp/support/>) をご覧ください。



# 目次

---

## 第 1 章: EPAgent のインストール、構成、および使用 13

|  |    |
|--|----|
| Environment Performance Agent の概要.....     | 13 |
| EPAgent によるプロセスからの入力の受信 .....              | 14 |
| EPAgent およびスクリプト環境.....                    | 15 |
| EPAgent でのネットワーク ソースからの入力.....             | 16 |
| EPAgent および CA Introscope®.....            | 17 |
| サポートされるプラグイン.....                          | 17 |
| Environment Performance Agent のインストール..... | 19 |
| EPAgent スタンドアロンインストーラ .....                | 20 |
| EPAgent の構成 .....                          | 21 |
| EPAgent プロパティの設定 .....                     | 21 |
| EPAgent ログ記録オプションの設定 .....                 | 22 |
| EPAgent プラグインの構成 .....                     | 23 |
| ネットワーク データ ソースに関する EPAgent の構成.....        | 26 |
| EPAgent の実行 .....                          | 28 |
| UNIX での EPAgent の実行 .....                  | 28 |
| EPAgent でのカスタム プラグインの作成および使用 .....         | 29 |
| メトリック データ形式.....                           | 29 |
| 2つのカスタム プラグイン形式に関する注意事項 .....              | 32 |
| エラーまたはイベント データ形式.....                      | 32 |
| XML 形式でのエラー スナップショットの作成.....               | 33 |
| EPAgent イベントとトランザクション追跡 .....              | 34 |
| エラーまたはイベント データの XML スキーマ .....             | 36 |

## 第 2 章: EPAgent のデフォルト プラグイン 37

|   |    |
|---|----|
| solaris/diskStats.pl .....  | 37 |
| 生成されるメトリック .....  | 37 |
| コマンドライン引数.....  | 38 |
| windows/processAvailability.pl、および solaris/processAvailability.pl ..... | 38 |
| 生成されるメトリック .....  | 38 |
| コマンドライン引数.....  | 39 |
| windows/diskStats.pl .....  | 39 |
| 生成されるメトリック .....  | 39 |
| コマンドライン引数.....  | 40 |

---

|  |    |
|--|----|
| apacheLogReader.pl、weblogicLogReader.pl、および websphereLogReader.pl..... | 40 |
| 生成されるメトリック .....   | 40 |
| コマンドライン引数.....   | 41 |
| httpSvcAvailability.pl.....  | 41 |
| 生成されるメトリック .....   | 41 |
| コマンドライン引数.....   | 42 |
| Perl サポートの問題 .....   | 43 |
| kstat.pl.....  | 43 |
| 生成されるメトリック .....   | 43 |
| コマンドライン引数.....   | 43 |
| Perl サポートの問題 .....   | 44 |
| iisLogReader.pl .....  | 44 |
| 生成されるメトリック .....   | 45 |
| コマンドライン引数.....   | 45 |
| iisLogReader.cfg 環境設定ファイル.....   | 45 |
| IIS4.0 および IIS5.0 での IIS Web サーバ ログ記録の設定.....                          | 54 |

### 第 3 章: EPAgent デフォルト プラグイン メトリック 57

|   |    |
|---|----|
| 無効な文字の置き換え .....                              | 57 |
| IIS ログリーダーメトリック .....                         | 58 |
| Total Page Hits .....                         | 58 |
| Total Visitors Count.....                     | 58 |
| Page Hits For <URL> .....                     | 59 |
| Bytes Transferred.....                        | 59 |
| ErrorCount_400_BadRequest .....               | 59 |
| ErrorCount_401_Unauthorized .....             | 60 |
| ErrorCount_403_Forbidden .....                | 60 |
| ErrorCount_404_ServerConfiguration .....      | 60 |
| ErrorCount_500_InternalServerError .....      | 61 |
| ErrorMessage_400_BadRequest.....              | 61 |
| ErrorMessage_401_Unauthorized .....           | 61 |
| ErrorMessage_403_Forbidden.....               | 62 |
| ErrorMessage_404_PageNotFound .....           | 62 |
| ErrorMessage_500_InternalServerError .....    | 62 |
| HTTP/HTTPS サービス可用性メトリック .....                 | 63 |
| Available.....                                | 63 |
| 応答時間.....                                     | 63 |
| Average Throughput .....                      | 64 |
| Solaris カーネル統計情報 (Kstat) メトリック .....          | 64 |
| Solaris Kernel <クラス> <モジュール> <名前> メトリック ..... | 64 |



---

|   |    |
|---|----|
| プロセスの可用性メトリック .....                                     | 65 |
| Count.....  | 65 |
| ディスク統計情報 (Windows) メトリック .....                          | 65 |
| Disk Reads/sec .....                                    | 66 |
| Disk Writes/sec .....                                   | 66 |
| Disk Read Bytes/sec .....                               | 67 |
| Disk Write Bytes/sec .....                              | 67 |
| Current Disk Queue Length .....                         | 67 |
| Average Disk Read Time (ms).....                        | 68 |
| Average Disk Write Time (ms).....                       | 68 |
| Average Disk Bytes/Read .....                           | 68 |
| Average Disk Bytes/Write .....                          | 69 |
| Free Disk Space (mb).....                               | 69 |
| Free Disk Space (%) .....                               | 69 |
| ディスク統計情報 (Solaris) メトリック .....                          | 70 |
| Disk Time (%).....                                      | 70 |
| Transaction Waiting Time (%) .....                      | 71 |
| Average # Transaction Active.....                       | 71 |
| Average # Transaction Waiting .....                     | 71 |
| Disk Reads/sec .....                                    | 72 |
| Disk Writes/sec .....                                   | 72 |
| KB Read per Second .....                                | 72 |
| KB Written per Second.....                              | 73 |
| Average Service Time in Waiting Queue (ms).....         | 73 |
| Average Service Time for Active Transactions (ms) ..... | 73 |
| Free Disk Space (mb).....                               | 74 |
| Free Disk Space (%) .....                               | 74 |
| Apache ログメトリック .....                                    | 74 |
| PageHitCount .....                                      | 75 |
| ErrorCount_400_BadRequest .....                         | 75 |
| ErrorCount_401_Unauthorized .....                       | 76 |
| ErrorCount_403_Forbidden .....                          | 76 |
| ErrorCount_404_PageNotFound.....                        | 76 |
| ErrorCount_500_ServerConfiguration .....                | 77 |
| ErrorMessage_400_BadRequest.....                        | 77 |
| ErrorMessage_401_Unauthorized .....                     | 77 |
| ErrorMessage_403_Forbidden.....                         | 78 |
| ErrorMessage_404_PageNotFound .....                     | 78 |
| ErrorMessage_500_ServerConfiguration.....               | 78 |
| WebSphere ログメトリック .....                                 | 79 |
| MessageCount_Audit.....                                 | 79 |
| MessageCount_Warning.....                               | 80 |

---

---

|                               |    |
|-------------------------------|----|
| MessageCount_EventDebug ..... | 80 |
| MessageCount_Error .....      | 80 |
| MessageCount_Fatal.....       | 81 |
| MessageText_Audit.....        | 81 |
| MessageText_Warning.....      | 81 |
| MessageText_EventDebug.....   | 82 |
| MessageText_Error .....       | 82 |
| MessageText_Fatal .....       | 82 |
| WebLogic ログ メトリック .....       | 86 |
| ServerVersion.....            | 86 |
| MessageCount_Warning.....     | 87 |
| MessageCount_Error .....      | 87 |
| MessageCount_Notice .....     | 87 |
| MessageCount_Critical .....   | 88 |
| MessageCount_Alert.....       | 88 |
| MessageCount_Emergency.....   | 88 |
| MessageText_Warning.....      | 89 |
| MessageText_Error .....       | 89 |
| MessageText_Notice.....       | 89 |
| MessageText_Critical .....    | 90 |
| MessageText_Alert .....       | 90 |
| MessageText_Emergency .....   | 90 |

## 第 4 章: EPAgent のプロパティ 91

|  |     |
|--|-----|
| IntroscopeEPAgent.properties ファイル .....                      | 91  |
| エージェント名およびプロセス名の設定 .....                                     | 92  |
| Agent メトリックのエイジング .....                                      | 93  |
| エージェント メトリック エイジングの設定 .....                                  | 94  |
| introscope.agent.metricAging.turnOn .....                    | 95  |
| introscope.agent.metricAging.heartbeatInterval .....         | 96  |
| introscope.agent.metricAging.dataChunk .....                 | 96  |
| introscope.agent.metricAging.numberTimeslices .....          | 97  |
| introscope.agent.metricAging.metricExclude.ignore.0 .....    | 97  |
| Case-Sensitive メトリック .....                                   | 98  |
| introscope.epagent.metricscasesensitive .....                | 98  |
| ChangeDetector の設定.....                                      | 98  |
| introscope.changeDetector.enable.....                        | 99  |
| introscope.changeDetector.rootDir .....                      | 100 |
| introscope.changeDetector.isengardStartupWaitTimeInSec.....  | 100 |
| introscope.changeDetector.waitTimeBetweenReconnectInSec..... | 101 |
| introscope.changeDetector.enableEPA .....                    | 101 |

---

|   |     |
|---|-----|
| introscope.changeDetector.agentID .....                                 | 102 |
| introscope.changeDetector.profile .....                                 | 102 |
| introscope.changeDetector.profileDir .....                              | 103 |
| Enterprise Manager 接続.....  | 103 |
| introscope.agent.enterprisemanager.connectionorder .....                | 103 |
| introscope.agent.enterprisemanager.transport.tcp.host.DEFAULT .....     | 104 |
| EPAgent Configuration .....   | 104 |
| introscope.epagent.config.networkDataPort .....                         | 104 |
| introscope.epagent.config.httpServerPort .....                          | 105 |
| 拡張ディレクトリ .....  | 105 |
| introscope.agent.extensions.directory .....                             | 105 |
| ログ .....  | 106 |
| log4j.logger.EPAgent .....  | 106 |
| log4j.appender.logfile.File.....  | 106 |
| プロセス.....   | 107 |
| introscope.agent.customProcessName .....                                | 107 |
| introscope.agent.defaultProcessName.....                                | 107 |
| introscope.agent.agentName .....  | 108 |
| リモート設定.....   | 108 |
| introscope.agent.remoteagentconfiguration.enabled .....                 | 108 |
| ステートフルプラグイン .....   | 109 |
| introscope.epagent.plugins.stateful.names.....                          | 109 |
| introscope.epagent.stateful.<pluginname>.command.....                   | 109 |
| introscope.epagent.stateful.<pluginname>.class .....                    | 110 |
| ステートレス プラグイン .....  | 110 |
| introscope.epagent.plugins.stateless.names.....                         | 111 |
| introscope.epagent.stateless.<pluginname>.command .....                 | 111 |
| introscope.epagent.stateless.<pluginname>.class .....                   | 111 |
| introscope.epagent.stateless.<pluginname>.delayInSeconds .....          | 112 |
| introscope.epagent.stateless.<pluginname>.schedule.....                 | 112 |
| introscope.epagent.config.stalledStatelessPluginTimeoutInSeconds .....  | 113 |
| introscope.epagent.stateless.<pluginname>.metricNotReportedAction ..... | 114 |



# 第 1 章: EPMonitor のインストール、構成、および使用

---

このセクションでは、CA APM Environment Performance Agent (EPMonitor) のインストール、構成、および使用について説明します。

このセクションには、以下のトピックが含まれています。

[Environment Performance Agent の概要](#) (P. 13)

[Environment Performance Agent のインストール](#) (P. 19)

[EPMonitor の構成](#) (P. 21)

[EPMonitor の実行](#) (P. 28)

[UNIX での EPMonitor の実行](#) (P. 28)

[EPMonitor でのカスタムプラグインの作成および使用](#) (P. 29)

## Environment Performance Agent の概要

CA APM Environment Performance Agent (EPMonitor) は、Java 環境以外のソースからのメトリック データを CA Introscope® に統合します。

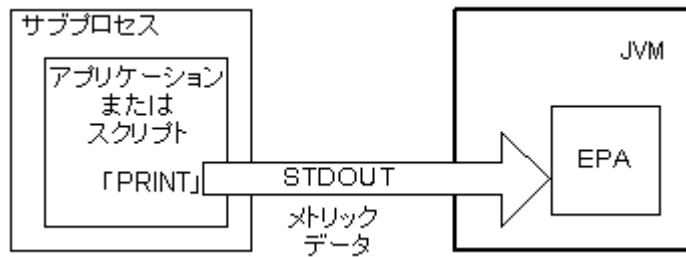
EPMonitor では、1つのプロセスに対しサブプロセスを生成し、そのサブプロセスからオペレーティング システムのパイプ (標準の IPC メカニズム) を介して標準出力を受け取るという、多くのオペレーティング システムで使用できる機能を利用しています。サブプロセスを起動する機能と、Perl のような柔軟なスクリプト環境とを組み合わせることで、事実上どのソースからでもアプリケーションのパフォーマンス情報を収集することができる、柔軟で強力な機能が得られます。

## EPAgent によるプロセスからの入力の受信

EPAgent は、Java で実装され、Java 仮想マシン (JVM) で実行されます。EPAgent は、JVM 機能を利用してサブプロセスを起動し、そのプロセスから **STDOUT** および **STDERR** を受信します。任意のスクリプトや実行可能ファイル、その他コマンドラインで起動可能なものはすべてプロセスとなります。例：

- C 言語で記述されたコンパイル済みプログラム
- Perl スクリプト

簡単な **PRINT** 関数を実行することによって、サブプロセスのアプリケーションまたはスクリプトは EPAgent と通信できます。以下の図は、アプリケーションまたはスクリプトが **STDOUT** (標準出力) に出力することによって、メトリックデータを EPAgent に渡す方法を示しています。



EPAgent は、**STDOUT** (標準出力) を使用して受信するテキストが以下のいずれかの特定の形式に準拠していることが想定されています。

- シンプルな名前/値のペア

または

- XML

EPAgent テキスト形式の要件とオプションは、このマニュアルの後半で詳細に説明しています。

このマニュアルでは、EPAgent にデータを送信するアプリケーションまたはスクリプトを EPAgent プラグインと呼びます。EPAgent には、詳細なメトリックデータを提供するいくつかのデフォルトプラグインが含まれます。また、ユーザはカスタムのプラグインを作成することもできます。

詳細:

[EPAgent プラグインの構成 \(P. 23\)](#)

[EPAgent でのカスタム プラグインの作成および使用 \(P. 29\)](#)

[EPAgent のデフォルトプラグイン \(P. 37\)](#)

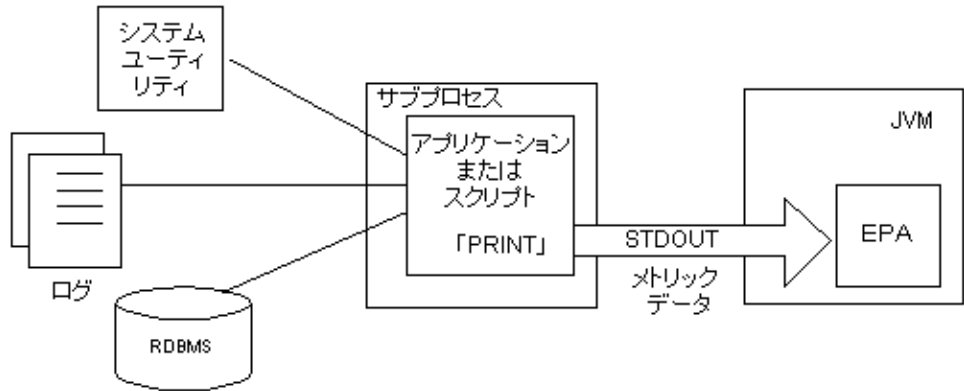
## EPAgent およびスクリプト環境

EPAgent は、コンパイル済みアプリケーションを含め、サブプロセスの任意の実行可能ファイルのエンティティからテキストを受信することができます。もっとも柔軟なアプローチは、Perl、KornShell、または REXX などのスクリプト環境を使用することです。 インタープリタ型のスクリプト環境には実装の容易さという利点があり、ライブラリ、またはリレーショナルデータベース、OS サブシステム（プロセステーブル、ファイルシステムなど）、およびサードパーティ製品などの実際のデータソースへのインターフェースが数多く提供されています。

スクリプトは、幅広いソースから情報を収集することができます。たとえば、スクリプトによって以下のことが可能です。

- ライブラリ関数、またはシステムユーティリティを呼び出し、データベースなどの重要なアプリケーションの存在を確認する。
- アプリケーションログファイルをスキャンおよび解析して、アプリケーションエラーを検出する。

スクリプト環境の柔軟性により、EPAgent が実質的にあらゆるソースからパフォーマンス、および管理情報を収集することが可能になります。以下の図は、スクリプトが多くのソースから情報にどのようにアクセスできるかを示しています。



Perl インタープリタはさまざまなプラットフォーム上で使用でき、オペレーティングシステム、ミドルウェア、およびサードパーティ製品へのデータの API を幅広くサポートしているため、CA Technologies では、EPAgent で Perl を使用することをお勧めします。CA Introscope® EPAgent には、プロセスの可用性の確認、ディスクパフォーマンス統計情報の取得、および HTTP ログの読み取りなどのさまざまな機能を実行するための、Perl スクリプトのサンプルセットが含まれています。これらのスクリプトを拡張して、その他の機能を実行させることも可能です。

**注:** PBD、EPAgent スクリプト、または JavaScript 計算機のカスタマイズに関するサポートについては、CA Technologies プロフェッショナルサービスにお問い合わせください。

## EPAgent でのネットワークソースからの入力

ローカルプロセスからの入力に加えて、EPAgent はネットワークソースからの入力も使用できます。これは、情報を収集するプロセスがネットワーク化されていて、EPAgent が実行できない場合に役立ちます。たとえば、ネットワークポートを使用して監視情報を提供したり Web URL に定期的にデータをポストする C++ バックエンドシステムのようなプロセスが該当します。

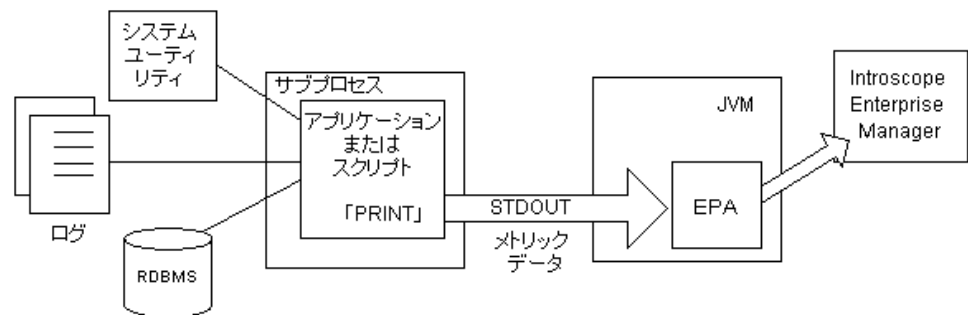


詳細:

[ネットワーク データ ソースに関する EPAgent の構成 \(P. 26\)](#)

## EPAgent および CA Introscope®

EPAgent は、スタンドアロン プロセスとして動作します。Java エージェント、または .NET Agent 同様に、Enterprise Manager に接続し、管理、およびパフォーマンス情報を提供します。以下の図は、Enterprise Manager に、EPAgent がどのようにデータを転送するかを示しています。



EPAgent によって、使用するアプリケーション環境の全体的なビューを取得することができます。分散型アプリケーションアーキテクチャは複雑であり、Java および Java 以外のアプリケーションサブシステムの混在で構成されることがよくあります。EPAgent を使用することで、CA Introscope® Workstation 内の Java および .NET メトリックと併せて、アプリケーション環境内の Java 以外および .NET 以外の部分からの情報を収集し表示することができるようになります。

## サポートされるプラグイン

EPAgent は、ステートフル、およびステートレス プラグインの両方をサポートします。

注: すべてのデフォルトのプラグインで、Perl バージョン 5.00503 以降が必須です。

## プラグインタイプ

利用可能なプラグインタイプは以下の通りです。

- **ステートフルプラグイン** - ステートフルプラグインは、EPAgent が起動する際に起動され無期限に実行する、長期実行スクリプト（デーモン）を想定しています。これらのステートフルプラグインは、プラグインの標準出力チャンネルを介して、データを CA Introscope® に随時取り込むことができます。ステートフルプラグインが終了した場合は、EPAgent によって再起動されます。
- **ステートレスプラグイン** - ステートレスプラグインは、定期的なスケジュールで実行するように設計されており、実行する必要がある周期（実行から次の実行までの遅延時間として指定されます）を設定することができます。収集したデータを標準出力チャンネルを介して EPAgent に送信し、その後で終了するような短期実行スクリプトを想定しています。

**注:** EPAgent では、同時に複数のステートレスプラグインのインスタンスが実行されていないかを確認するためのチェックを特に行っていないので、プラグイン開発者はステートレスプラグインが必要最小限の時間内で実行し、完了するように設計する必要があります。

## エラー ログ

エラーは、プラグインの標準エラーチャンネルを介してログに記録されます。EPAgent は、このエラー出力をその他のエラーと同様にログに記録することができます。

## デフォルト プラグイン

EPAgent には、最小限の構成で使用できるデフォルトのプラグインのライブラリが含まれています。以下の表に、各プラグインの機能とサポートされているオペレーティングシステムについての概要を示します。

| プラグイン         | 機能  | サポートされている OS |
|---------------|---|--------------|
| プロセス可用性チェック   | 指定した名前を持つ実行中のプロセスを検索                      | Solaris      |
| OS レベル ディスク統計 | 読み取り/書き込み、トランザクション時間、待機時間 %、ディスク容量などをレポート | Solaris      |

| プラグイン                         | 機能   | サポートされている OS |
|-------------------------------|--|--------------|
| Apache ログリーダー                 | HTTP エラー数、トランザクション数、平均トランザクションサイズをレポート   | Solaris      |
| WebLogic ログリーダー               | エラー数、警告数をレポート  | Solaris      |
| WebSphere ログリーダー              | エラー数、警告数をレポート  | Solaris      |
| IIS ログリーダー                    | タイプごとのエラー数、使用されたページ数、IP アドレスごとのユーザ数をレポート   | Windows      |
| HTTP サービス可用性<br>HTTPS サービス可用性 | 接続の成功/失敗、接続の往復時間、平均スループット/帯域幅をレポート   | Solaris      |
| Solaris Kstat 統計              | I/O 統計情報など、Kstat ユーティリティを介して使用できるすべての統計情報、すべてのデバイスのネットワーク統計情報、仮想メモリ統計情報をレポートします。 | Solaris      |

## カスタム プラグイン

EPAgent は、任意のプログラムおよびプラグインからデータを収集することができますが、そのためにはサポートされている 2 種類の形式のいずれかでデータが出力される必要があります。

詳細:

[EPAgent でのカスタム プラグインの作成および使用 \(P. 29\)](#)

## Environment Performance Agent のインストール

EPAgent をインストールするには、スタンドアロン エージェント インストーラを使用します。

注: 詳細については、「CA APM インストールおよびアップグレードガイド」を参照してください。

## EPAgent スタンドアロン インストーラ

以下のスタンドアロン エージェント インストーラのいずれかを使用して、EPAgent を別のコンピュータ上にインストールすることができます。CA APM 製品は、[CA サポート](#) の CA APM ソフトウェア ダウンロード セクションからダウンロードできます。

- EPAgent<バージョン>windows.zip
- EPAgent<バージョン>ebcdic.tar
- EPAgent<バージョン>unix.tar

EPAgent インストーラを抽出すると、以下のファイルが <EPAgent\_Home>/epagent ディレクトリにインストールされます。

注: <EPAgent\_Home> は EPAgent をインストールしたコンピュータです。

- IntroscopeEPAgent.properties
- lib/EPAgent.jar
- epaplugins/ (プラグインの最上位ディレクトリ)
  - 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 ファイルのトップレベルのディレクトリ)
  - ChangeDetectorAgent.jar
  - ChangeDetector-Agent\_Server.jar
  - ChangeDetector-CommonAll.jar
  - ChangeDetector-config.xml
  - Windows : cdnativefile.dll
  - UNIX : libcdnativefile.so

## EPAgent の構成

EPAgent には、Enterprise Manager の場所、およびポート設定などの、エージェントの構成プロパティのセットが含まれています。

注: プロパティ ファイルの設定で何らかのエラー（無効なファイル名パスなど）があると、EPAgent は起動に失敗します。ブート時に EPAgent を自動的に起動する場合には、プロパティが正しく設定されていることを確認してください。

詳細:

[EPAgent プロパティの設定](#) (P. 21)

[EPAgent ログ記録オプションの設定](#) (P. 22)

[EPAgent プラグインの構成](#) (P. 23)

[ネットワーク データ ソースに関する EPAgent の構成](#) (P. 26)

## EPAgent プロパティの設定

EPAgent で使用されるプロパティは Java エージェントで使用されるプロパティと同様です。

次の手順に従ってください:

1. *IntroscopeEPAgent.properties* ファイルで EPAgent を設定します。  
EPAgent の設定は、CA Introscope® エージェント プロファイルで定義される設定と同じタイプのものを使用します。

注: プロパティの詳細については、「CA APM Java Agent 実装ガイド」を参照してください。

2. *IntroscopeEPAgent.properties* ファイルの名前または場所を変更する場合、以下の Java システム プロパティを使用して設定できます。

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

注: このシステム プロパティは、コマンドラインで「java」という文字列に続けて入力する必要があります。コマンドラインの後方（たとえば `-jar` の後など）に記述された場合は無効です。

## EPAgent ログ記録オプションの設定

デフォルトでは、EPAgent はメッセージ、およびエラー出力をコマンド コンソールに送信します。メッセージおよびエラー出力がログ ファイルにも送信されるように EPAgent を設定できます。

次の手順に従ってください:

1. `<EPAgent_Home>/epagent/IntroscopeEPAgent.properties` ファイルを開きます。
2. プロパティを変更します。

## 冗長モードでの EPAgent の実行と出力ファイルのリダイレクト

EPAgent を冗長モードで実行すると、ログに詳細な情報が記録されます。これはデバッグ時に役立ちます。冗長モードの有無を設定するプロパティでは、ログ ファイルの出力先も設定できます。

次の手順に従ってください:

1. `<EPAgent_Home>/epagent/IntroscopeEPAgent.properties` ファイルを開きます。

2. プロパティ `log4j.logger.EPAgent` を以下のようにコメント化します。  
`#log4j.logger.EPAgent=INFO, console`
3. 出力をコンソールではなくログ ファイルに送るようになるには、プロパティ `log4j.logger.IntroscopeAgent` のコメント化を解除し、`INFO` を `VERBOSE#com.wily.util.feedback.Log4JSeverityLevel` に、`console` を `logfile` にそれぞれ置き換えます。たとえば、EPAgent から冗長モードでログ ファイルにレポートが出力されるようになるには、プロパティは以下 のようになります。  
`log4j.logger.IntroscopeAgent=VERBOSE#com.wily.util.feedback.Log4JSeverityLevel,logfile`
4. 変更を保存し、`IntroscopeEPAgent.properties` ファイルを閉じます。

**重要:** `IntroscopeEPAgent.properties` ファイルを保存した後も、プロパティ ファイルが読み取られて実装されるまでは、EPAgent ログはコマンド コンソールにそのまま表示されます。プロパティ ファイルが実装されると、以降のログはログ ファイルに書き込まれるようになります。

## EPAgent プラグインの構成

`IntroscopeEPAgent.properties` ファイルの特定のプロパティをコメント化 解除するだけで、デフォルトのプラグインを使用して EPAgent を実行す ることができます。

ただし、`IntroscopeEPAgent.properties` ファイル内のデフォルトのプラグイ ンから必要のないプラグインを削除したり、プラグインを追加することも 可能です。

EPAgent プラグインは、`IntroscopeEPAgent.properties` ファイルで、ステート フルとステートレスの 2 つのセクションに分けられます。

### ステートフル プラグインの追加

EPAgent によって使用されるプラグインは、`IntroscopeEPAgent.properties` ファイルで設定されます。

次の手順に従ってください:

1. `<EPAgent_Home>/epagent/IntroscopeEPAgent.properties` ファイルを開き ます。

2. プロパティ `introscope.epagent.plugins.stateful.names` をコメント化解除し、すべてのステートフルプラグインの名前をカンマ区切りで以下の例のように追加します。  
`introscope.epagent.plugins.stateful.names=APACHE,WEBLOGIC,WEBSPPHERE,IIS`
3. 前の手順で追加されたステートフルプラグインの名前に、必要なすべてのコマンドライン引数を含めて、ステートフルプラグインを開始または起動するのに必要な完全なコマンドを指定する行を追加します。

**注:** コメント化解除することによってファイル内のデフォルトのプロパティを使用できます。ただしプラグインを開始または起動するには、必要なコマンドラインを完全な形で記述する必要があります。

前の手順で指定したステートフル名のプロパティには、以下の 4 行が必要です。

```
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.WEBSPPHERE.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:** Windows マシンの場合、パスを追加する際には以下のようにして、もう 1 つの円記号を使って円記号をエスケープする必要があります。

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

詳細:

[EPAgent のデフォルトプラグイン \(P. 37\)](#)

## 次の手順

[ステートレスプラグインを追加 \(P. 25\)](#) できます。ステートレスプラグインを追加しない場合は、`IntroscopeEPAgent.properties` ファイルへの変更を保存し、EPAgent を再起動します。



## ステートレス プラグインの追加

ステートレス プラグインを追加できます。

次の手順に従ってください:

1. `<EPAgent_Home>/epagent/IntroscopeEPAgent.properties` ファイルを開きます。
2. プロパティ `introscope.epagent.plugins.stateless.names` で、プロパティをコメント化解除し、すべてのステートレス プラグインに対し、カンマ区切りで、以下のようにして論理名を追加します。  
`introscope.epagent.plugins.stateless.names=DISK,ORACLE,HTTP,KSTAT`
3. 前の手順で追加されたステートレス プラグイン名にさらに 2 行を追加します。以下のように指定します。

- ステートレス プラグインを開始または起動するのに必要な、完全コマンドライン（「EPAgent のデフォルト プラグイン」を参照して必要なすべてのコマンドライン引数を含めます）。以下のような構文になります。

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

- EPAgent がプラグインを実行する周期。実行から実行までの遅延時間として秒単位で定義します。以下のような構文になります。  
`introscope.epagent.stateless.<pluginname>.delayInSeconds=<value in seconds>`

例:

```
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: これらの例は、これらのプロパティのデフォルトの値です。使用するには、それぞれプロパティをコメント化解除する必要があります。

4. 作業が終了したら、変更を保存し EPAgent を再起動します。

詳細:

[EPAgent のデフォルトプラグイン \(P. 37\)](#)

### ステートフルおよびステートレス プラグインの削除

任意の理由でプラグインを削除できます。

次の手順に従ってください:

1. <EPAgent\_Home>/epagent/IntroscopeEPAgent.properties ファイルを開きます。
2. プロパティ `introscope.epagent.plugins.stateful.names` および `introscope.epagent.plugins.stateless.names` で、削除対象のプラグインの名前を削除します。
3. 削除するプラグインに関連付けられたすべてのコマンドラインをコメント化するか、または削除します。
4. 作業が終了したら、変更を保存し EPAgent を再起動します。

### Java クラス プラグインの構成

EPAgent JVM 内で実行される Java クラスのプラグインを設定するには、EPAgent プロパティ ファイルで、以下のプロパティを設定します。  
`introscope.epagent.stateful.JAVA-CLASS.class=my.package.path.Class arg1 arg2 arg3`

注: 完全修飾クラス名に続けて任意のオプションを入力します。たとえば、以下のように入力します。

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

### ネットワーク データソースに関する EPAgent の構成

ネットワーク ソースからデータを受信するように EPAgent を構成することができます。

- シンプル形式または XML 形式のネットワーク入力に関する EPAgent の構成
- [HTTP GET 入力に関する EPAgent の構成 \(P. 27\)](#)

## シンプル形式または XML 形式のネットワーク入力に関する EPAgent の構成

ネットワーク経由でデータを受け取れるように EPAgent を設定するには、EPAgent プロパティ ファイルの以下の行のコメント化を解除します。

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

## HTTP GET 入力に関する EPAgent の構成

EPAgent で、簡易 HTTP サーバを作成して HTTP GET 要求をリスンします。HTTP GET 要求は Enterprise Manager に送信するメトリックを定義します。

HTTP GET 要求は、以下の構文を使用します。

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

### 簡易 HTTP サーバを有効化およびテストする方法

1. EPAgent プロパティ ファイル内で、以下のプロパティをコメント化解除することによって、簡易 HTTP サーバを有効にします。

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

2. EPAgent を再起動します。

変更が反映されます。

3. ブラウザを通して、HTTP GET 要求をテストします。例：

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

15 秒後、EPAgent で、MyHTTPMetric|Test:MyCounter メトリックが表示されます。

メトリックを更新するには、EPAgent で新しい HTTP GET 要求を受信する必要があります。したがって、簡易 HTTP サーバ機能は、複数の HTTP GET 要求を発行するツールと併用するよう意図されています。たとえば、複数の `wget` コマンドを発行するオープンソースツール `Wget` を使用して、複数の HTTP GET 要求を作成することができます。Wget の詳細については、以下を参照してください。

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

## EPAgent の実行

EPAgent は、スタンドアロンの .jar ファイル、または Java アプリケーションのいずれかとして実行できます。

### スタンドアロン .jar ファイルとして EPAgent を実行する方法

- 以下の例のように適切な `-jar` フラグを使用して、Java コマンドラインを実行します。

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

### Java アプリケーションとして EPAgent を実行する方法

- 適切なクラスパスに EPAgent ファイルを以下の例のように追加します。

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

## UNIX での EPAgent の実行

制御スクリプト（シェル スクリプト）を使用して、UNIX オペレーティングシステムで Introscope EPAgent を実行できます。

次の手順に従ってください:

1. コマンドプロンプトを開きます。
2. コントロールスクリプトがあるディレクトリに移動します。例:

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

3. 目的のアクションに対応するコマンドを実行します。

EPACtrl.sh start

EPAgent を開始します。

EPACtrl.sh status

EPAgent プロセスのステータス（実行中または停止中）を表示します。

EPACtrl.sh stop

EPAgent プロセスを停止します

EPACtrl.sh help

ヘルプメニューを表示します。

## EPAgent でのカスタム プラグインの作成および使用

デフォルトのプラグインを使用して EPAgent を実行することで、通常ほとんどのユーザのニーズを満たす情報を収集できます。ただし、EPAgent でカスタム プラグインを使用したい場合があります。これを行うには、以下の 2 つのガイドラインに従います。

- カスタム プラグインの出力が EPAgent で解析可能であることを確認します（「[メトリック データ形式 \(P. 29\)](#)」、「[エラーまたはイベント データ形式 \(P. 32\)](#)」を参照してください）。
- 「[EPAgent プラグインの構成 \(P. 23\)](#)」セクションの説明に従って、*IntroscopeEPAgent.properties* ファイルに、カスタム プラグインを追加します。

### メトリック データ形式

EPAgent は、プラグイン（または EPAgent にプラグインとして提供されている、その他のメトリック生成プログラム）から提供された以下の 2 つの異なる形式のメトリックデータを解析できます。

- シンプル
- XML

## シンプル形式のメトリック データ

以下の形式を使用して、行ごとに 1 つのメトリック名と値を指定します。

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

例 :

```
diskWrites=37
```

リソース セグメントへの参照を含めることもできます。

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

たとえば、以下のとおりです (すべて 1 行で記述してください)。

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

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

シンプル形式のガイドラインは、以下のとおりです。

- 「シンプル」形式では、メトリック名に等号 (=) を含めることはできません。名前に等号が必要な場合、XML 形式を使用します。
- 値には等号 (文字として) を含めることができます。EPAgent は常に最初の等号まで (左から右に読む場合) すべての文字をメトリック名として解析し、最初の等号の後の文字をすべて値として解析します。
- 数字から構成される値はすべて、数値データとして解釈され、CA Introscope® の「IntCounter」型としてレポートされます。
- 数字以外で構成された値はすべて、文字列データとして解釈され、CA Introscope® の「string event」型としてレポートされます。

## XML 形式のメトリックデータ

「シンプル」形式では Introscope メトリック タイプに制限があるのに対し、XML スタイル形式では、プラグインが、Introscope メトリック名、Introscope メトリック タイプ、および値などの追加情報を、以下の例に示すようにレポートできるようになります。

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

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

XML 形式のガイドラインは、以下のとおりです。

- この形式により、Introscope データ タイプを完全にサポートし、メトリック名と値の両方で等号が使用できるようになります。
- メトリックの「type」属性は、以下のいずれかである必要があります。
  - **PerIntervalCounter** - 値は「間隔ごと」のレート。間隔は変更可能です。これらのメトリックは、値を合計することによって、一定期間で集約されます。たとえば、15 秒間に 10 のメソッド呼び出しがあり、次の 15 秒間に 15 のメソッド呼び出しがあった場合、30 秒に集約すると、「30 秒間に 25 メソッド呼び出し」になります。
  - **IntCounter** - 上下に変動する int 値
  - **IntAverage** - 一定期間で平均される int 値
  - **IntRate** - 値は 1 秒あたりのレートです。これらのメトリックは、値の平均値を利用することによって一定期間で集約されます。
  - **LongCounter** - 上下に変動する long 値
  - **LongAverage** - 一定期間で平均される long 値
  - **StringEvent** - 定期的に文字列を生成するタイプを表しています。このレコーダは、現在の値の概念を持たず、レポートされた順番にイベントをレポートするのみです。
  - **Timestamp** - 連続して増加するタイプスタンプを生成するタイプ。
- 比較には大文字小文字の区別がありません（プラグインの作成を容易にします）。数値タイプが入力されたが、値が数値以外の場合、Introscope には何もレポートされず、EPAgent によってエラーがログに記録されます。

## 2つのカスタム プラグイン形式に関する注意事項

どちらかのタイプのカスタム プラグイン形式を使用するときは、以下の注意事項を考慮してください。

- 両方の形式をサポートすると、システムが不等号の小なり記号で始まるメトリック名を認識しないという、相互作用の原因になります。
- いずれの形式の場合も、形式が解析できない（無効または不正な構文）場合、EPAgent はその行を無視し、エラーをログに記録します。
- プラグインが複数の行を返す場合、次の行が続けて解析されます。
- 各メトリック名に対し、指定できるメトリックタイプは1つだけです。1つ以上のタイプが指定されると、以下のエラーが発生します。  
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

## エラーまたはイベント データ形式

EPAgent は、プラグインによって提供された以下の2つの異なる形式のエラーまたはイベント データを解析できます。

- シンプル
- XML

### シンプル形式のエラーまたはイベント データ

通常、シンプル形式のスクリプトは、以下の固定文字列で始まります。  
event:

コロンの後に続く内容の説明

- 「name=value」 ペアの一部であり、各ペアはアンパサンド (&) で区別されます。
- イベントに対するオプションのパラメータです。

下記の例は、Firefox ブラウザのプロセスを監視し、ブラウザが終了すると通知を送信するスクリプト例です。

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



## 単純な XML 形式のエラーまたはイベント データ

イベントは、XML 形式で指定することもでき、この場合、エージェント内のイベントを完全に表現できます。もっとも単純な XML 形式のイベントは、イベントを生成したリソースの名前（たとえば、「接続プール」または「Java 仮想マシン」など）を通知します。以下の例は、*Some Resource* でイベントが発生したことを通知しています。

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

タイムスタンプは、イベントが生成された時間であり、イベント所要時間はゼロです。

## パラメータと時間データを含む XML 形式のエラーまたはイベント データ

明示的なタイムスタンプと明示的な継続時間を示すイベント通知を構成できます。タイムスタンプの形式は、Java で解析可能な任意の形式を指定します。所要時間はミリ秒単位です。以下の例は、継続時間が 1 分 (60,000 ミリ秒) のイベントです。

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

## XML 形式でのエラー スナップショットの作成

エラー スナップショットは、パラメータ内でタイプをエラー スナップショットとして示す必要があります。

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

## ネストコンポーネント

以下の例は、ネストされたサブコンポーネントを持つイベントを示しています。イベントは、ゼロから無制限の数のサブコンポーネントを持つことができ、それらがそれぞれゼロから無制限の数のサブコンポーネントを持つこともできます。実際には、ネストの階層は小規模なものであるか、まったくないのが普通です。

```
<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 イベントとトランザクション追跡

[追跡ビュー] タブを選択して、EPAgent イベントをトランザクション追跡としてイベントビューアに表示することができます。

EPAgent から送信されたイベントに時間情報が含まれていると、追跡ビューが理解しやすいものになります。時間情報を含めるには、`<event>` および `<calledComponent>` タグの `startTime` および `offset` 属性を使用します。

`startTime` 属性は絶対時間です。形式は、`java.util.Date.parse()` によって解析できるすべての形式です。`<event>` エlementに `startTime` を指定する必要はありません。指定されていない場合、`System.currentTimeMillis()` または `new Date().getTime()` Java メソッドによって指定される際に、現在の時刻の値がデフォルトで設定されます。`startTime` を `<calledComponent>` Elementから省略すると、時間のデフォルトは含まれるElementの時間になるので、`startTime` 属性がどこにも指定されていない場合は、デフォルトですべて現在の時刻に設定されます。

`offset` 属性は整数値です。数値はミリ秒単位の時間と解釈され、`startTime` 属性 (`startTime` がデフォルトであれ明示的であれ) に加算されて、`<event>` または `<calledComponent>` にレポートされる実際の時刻を生成します。

### 例 1

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

このイベントのトレース ビューには、123456789 に開始された「カスタマイズ Web サーバ」および 123457089 (123456789 + 300) に開始された「Web サーバ モジュール」があります。各エレメントの所要時間の指定を合わせると、以下に示すような有用なトレース ビューを生成します。

- 「カスタマイズ Web サーバ」が 300 ミリ秒実行
- 「カスタマイズ Web サーバ」に呼び出され、「Web サーバ モジュール」が 100 ミリ秒実行
- 「Web サーバ モジュール」から戻った後に、「カスタマイズ Web サーバ」が 100 ミリ秒実行

### 例 2

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

この例は、例 1 と似ていますが「カスタマイズ Web サーバ」は現在の時刻に開始され、「Web サーバ モジュール」が 300 ミリ秒後に開始されます。この例の構成では、EPAgent スクリプトで現在時刻の取得が必要ない点に注意してください。

### 例 3

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

ここで、「カスタマイズ Web サーバ」は 123001000 (123000000 + 1000) に開始され、「Web サーバ モジュール」が 123003000 に開始されます。

所要時間を指定することで可読性と使いやすさが向上します。startTime、offset、duration が誤って指定された場合、トレース ビューが見にくくなるため、使用の際にはご注意ください。特に、<calledComponent> エレメントの startTime と offset を加えて計算した開始時刻は常に、それを含む <event> または <calledComponent> の開始時刻より後である必要があります。また、<calledComponent> の (開始時刻 + 継続時間) は常に、それを含む <event> または <calledComponent> の (開始時刻 + 継続時間) より少なくする必要があります。

## エラーまたはイベント データの XML スキーマ

フォーマル XSD スキーマは、以下の内容をサポートしています。

```
<?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. (イベントのルート エレメント。)
      This element is nearly equivalent to the calledComponent element, except that the event
      element must occur only once, at the outermost level. (このエレメントは、イベント エレ
      メントが最も外側のレベルで一度のみ発生する必要があることを除き、ほぼ calledComponent エレメ
      ントと同等です。) </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. (含んでいる
      エレメントによって呼び出されるコンポーネント。) This element is nearly equivalent to the
      event element, except that this element cannot occur at the outermost level. (この
      エレメントは、最も外側のレベルで発生しないことを除き、ほぼ イベント エレメントと同等です。)
    </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>
```

## 第 2 章: EPAgent のデフォルト プラグイン

---

このセクションでは、CA APM Environment Performance Agent で提供されるデフォルトのプラグイン、および各オプションのコマンドライン引数について説明します。

このセクションには、以下のトピックが含まれています。

[solaris/diskStats.pl](#) (P. 37)

[windows/processAvailability.pl](#)、および [solaris/processAvailability.pl](#) (P. 38)

[windows/diskStats.pl](#) (P. 39)

[apacheLogReader.pl](#)、[weblogicLogReader.pl](#)、および [websphereLogReader.pl](#) (P. 40)

[httpSvcAvailability.pl](#) (P. 41)

[kstat.pl](#) (P. 43)

[iisLogReader.pl](#) (P. 44)

### solaris/diskStats.pl

このプラグインは、システムのハードディスクに関する情報を含むメトリックを返します。

### 生成されるメトリック

*/solaris/diskStats.pl* プラグインによって作成されるメトリックの詳細については、「Solaris カーネル統計情報 (Kstat) メトリック」を参照してください。

## コマンドライン引数

このプラグインでは、コマンドライン引数としてディスク名を指定します。ディスク名が指定されていない場合、すべてのディスクに関する情報がレポートされます。

- たとえば、以下のコマンドは、すべてのローカルディスクについてのメトリックをレポートします。  
`perl diskStats.pl`
- 以下のコマンドは、ディスクの「/」および「/usr」のみについてのメトリックをレポートします。  
`perl diskStats.pl / /usr`

注: この引数を使用して結果の範囲を限定することをお勧めします。

## windows/processAvailability.pl、および solaris/processAvailability.pl

このプラグインは、引数の値で渡されたものと一致するプロセス名の数と同じ値のメトリックを返します。返されるメトリックのタイプは、「IntCounter」です。

## 生成されるメトリック

*/windows/processAvailability.pl*、および*solaris/processAvailability.pl* プラグインによって作成されるメトリックの詳細については、「プロセスの可用性メトリック」を参照してください。

## コマンドライン引数

これらのプラグインには、以下の 5 つのオプションのコマンドライン引数が必要です。

### **match**

検索に使用する Perl の正規表現

### **resource**

メトリック名のリソース部

### **subResource**

メトリック名のサブリソース部

### **metric**

メトリック名の主要部

### **distinctMatch**

これは値を持たず、各正規表現のそれぞれに一致するカウントを保持し、レポートするかどうかを決定します。このオプションは、指定されていれば、subResource 引数の値に優先します。

以下の例では、「sh」文字で始まるプロセスの数をレポートします。  
`perl processAvailability.pl -match ^sh`

## windows/diskStats.pl

このプラグインは、Windows パフォーマンス モニタによる統計情報に基づいたディスクの使用率/容量の統計をレポートします。

このプラグインがすべての統計情報を適切にレポートするためには、「diskperf -y」コマンドを発行する必要があります。その後で、新しい設定をアクティブ化するために、マシンを再起動する必要があります。

## 生成されるメトリック

`/windows/diskStats.pl` プラグインによって作成されるメトリックの詳細については、「ディスク統計情報 (Windows) メトリック」を参照してください。

## コマンドライン引数

このプラグインでは、コマンドライン引数としてディスク名を指定します。ディスク名が指定されていない場合、すべてのディスクに関する情報がレポートされます。

- たとえば、以下のコマンドは、すべてのローカルディスクについてのメトリックをレポートします。  
`perl diskStats.pl`
- 以下のコマンドは、ディスク `c:` および `d:` のみについてのメトリックをレポートします。  
`perl diskStats.pl c: d:`

注: この引数を使用して結果の範囲を限定することをお勧めします。

## apacheLogReader.pl、weblogicLogReader.pl、および websphereLogReader.pl

これらのプラグインは、<Web サーバ>`LogReader.cfg` ファイル内のメトリック構成に基づいてメトリックをレポートします。

それぞれの Web サーバのアプリケーション ログファイルパーサには、それぞれ、スリープ遅延、サブリソース名、リソース名、ログファイルディレクトリ、およびログファイル名パターンのデフォルトの値を設定した独自の環境設定ファイルがあります。

注: これらのログリーダープラグインのいずれかを使用すると、新しいエントリがログファイルに追加されるまで、EPAgent は Enterprise Manager へのレポートを開始しません。

## 生成されるメトリック

これらのプラグインによって作成されるメトリックの詳細については、以下で説明します。

- Apache ログメトリック
- WebSphere ログメトリック
- WebLogic ログメトリック



## コマンドライン引数

これらのプラグインには、以下の 4 つのオプションのコマンドライン引数が必要です。

### **logfileDir**

ログファイルが格納されるディレクトリ

### **sleepTime**

新しいメッセージをログファイルにログするかをチェックする間の  
(遅延) 秒数

### **logfile**

生成されるログファイルの名前に対応する正規表現

### **subResource**

メトリック名のサブリソース部

例：

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

## httpSvcAvailability.pl

このプラグインは、指定した URL での HTTP サービスの可用性およびパフォーマンスに基づくメトリックをレポートします。

## 生成されるメトリック

*httpSvcAvailability.pl* プラグインによって作成されるメトリックの詳細については、「HTTP/HTTPS サービス可用性メトリック」を参照してください。

## コマンドライン引数

このプラグインには、以下の 3 つのオプションのコマンドライン引数が必要です。

### url

オプションでポート番号の指定のあるカンマ区切りの URL セット。URL にはポート番号（デフォルトとは異なる場合）を含めることができ、また HTTP と HTTPS を識別するために `http:` または `https:` を含めることもできます。

### get

サーバに対して発行する、HTTP 要求のカンマ区切りリスト（スペースなし）が必要です。

Web サイトによっては HEAD 要求を受け入れないことがあり、またアプリケーションサーバによっては GET がサポートされないことがあります。

GET、HEAD のうち少なくとも 1 つを指定する必要があります。

GET と HEAD の両方が指定されると、HEAD がまず実行され、続いて GET が実行されます。

デフォルト： `-request HEAD`

### metric

返されるメトリックのカンマ区切りリスト

- `av` = 使用可能/接続可能（0= 不可、1= 可）
- `rt` = 往復応答時間（接続の場合）
- `tp` = スループット（KB。返された/往復応答の KB 数）
- `all` = すべての統計情報

デフォルト： `all`

例：

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

## Perl サポートの問題

*httpSvcAvailability.pl* プラグイン (HTTP) の場合 :

*httpSvcAvailability.pl* プラグイン (HTTP) では、基本的な http サービスの確認用に Perl ライブラリ *libwww-perl* (v5.65 以降) のインストールが必要です。 [www.activestate.com](http://www.activestate.com) では、*libwww-perl* をインストールの一部として含んだ Perl v5.6.1 を提供しています。

*httpSvcAvailability.pl* プラグイン (HTTPS) の場合 :

- Windows で https サービスを確認するには、*Crypt-SSLeay* (v0.17.1 以降) をインストールする必要があります。
- Solaris で https サービスを確認するには、*Net-SSLeay* (v1.05 以降) をインストールする必要があります。このどちらの場合でも、ActiveState Perl に含まれる PPM パッケージマネージャ ツールを使用すると、最も容易にインストールできます。

## kstat.pl

このプラグインは、Solaris の *kstat* ユーティリティによるカーネル統計情報に基づいたメトリックをレポートします。

## 生成されるメトリック

*kstat.pl* プラグインによって作成されるメトリックの詳細については、「Solaris カーネル統計情報 (Kstat) メトリック」を参照してください。

## コマンドライン引数

このプラグインには、以下の 4 つのオプションのコマンドライン引数が必要です。

### class

*kstat* クラスを照合する正規表現

### module

*kstat* モジュールを照合する正規表現

**name**

kstat 名を照合する正規表現

**statistic**

kstat 統計情報を照合する正規表現

**注:** これらの引数を使用して結果の範囲を限定することをお勧めします。 コマンドラインスイッチが指定されていない場合、プラグインは、クラス、モジュール、名前、および統計情報のすべての Solaris カーネルの情報を返します。

**例:**

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

Introscope メトリックは、ノード VM を作成し、次にリソース UNIX を、次に segmap を作成してから、「free」を表示し、次に値を表示します。

**注:** kstat によりリソース名に対して null 値がレポートされた場合、メトリックは生成されません。

## Perl サポートの問題

パフォーマンスを向上させるには、*kstat.pl* プラグインを Solaris 提供の Perl (通常、*/usr/bin/perl* にあり、Perl ライブラリは */usr/perl5* にインストールされています) で実行することをお勧めします。これは、内蔵されている Perl ライブラリが Solaris カーネルサブシステムへの直接のインタフェースをサポートしているためです。 *kstat.pl* プラグインを別の Perl インストールで実行する場合は、*kstat.pl* コマンドラインユーティリティを使用します。ただし、少し余分にオーバーヘッドを生じます。

## iisLogReader.pl

このプラグインは、*iisLogReader.cfg* ファイルの設定に基づいたメトリックをレポートします。

**注:** このログリーダープラグインを使用する場合、新しいエントリがログファイルに追加されるまで、EPAgent は Enterprise Manager へのレポートを開始しません。

## 生成されるメトリック

*iisLogReader.pl* プラグインによって作成されるメトリックの詳細については、「IIS ログリーダー メトリック」を参照してください。

## コマンドライン引数

このプラグインには、以下の 4 つのオプションのコマンドライン引数が必要です。

### **logfileDir**

ログファイルが格納されるディレクトリ

### **sleepTime**

新しいメッセージをログファイルにログするかをチェックする間の（遅延）秒数

### **logfile**

生成されるログファイルの名前に対応する正規表現

### **subResource**

メトリック名のサブリソース部

例：

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

## iisLogReader.cfg 環境設定ファイル

*iisLogReader.cfg* ファイルには、*iisLogReader.pl* ファイル内のコマンドライン引数の値が含まれています。

それぞれの Web サーバのアプリケーション ログファイルパーサには、それぞれ、スリープ遅延、サブリソース名、リソース名、ログファイルディレクトリ、およびログファイル名パターンのデフォルトの値を設定した独自の環境設定ファイルがあります。環境設定ファイルには、一致を照合し、レポートする、メトリックのリストも含まれています。各メトリックの設定では、**match**（正規表現）、**metricType**、**metricName**、および **metricValue** を指定できます。

## LOGFILEDIR

関連付けられた Web サーバのログファイルを含むディレクトリの完全パス名。

**注:** Windows のパスでもスラッシュは有効なパス区切り文字です（円記号も使用できます）。

**デフォルト**

C:/WINNT/system32/LogFiles/W3SVC1

## LOGFILE

読み取るログファイルの名前。これは、時間とともに変化する正規表現（たとえば、ログファイルの循環利用）である必要があります。この照合では、大文字と小文字が区別されません。

**注:** これは、ファイルの名前全体（パスを除く）が一致している必要があります。

**注:** IIS が W3C 拡張ログファイル形式に設定されている場合、デフォルトはログファイル名と一致するように設定されます。これは、*iisLogReader.pl* プラグインによってのみサポートされる形式です。

**デフォルト**

ex(tend)?%d+%log

## SLEEPTIME

新しいメッセージをログファイルにログするかをチェックする間の（遅延）秒数。

カウンタメトリックは、「間隔ごと」（メトリックの表を参照）と定義され、通常、SLEEPTIME は、Enterprise Manager のタイムスライス（15 秒）以上に設定する必要があります。それ以外の場合は、カウンタメトリックが不正確になります。

**デフォルト**

15

## RESOURCE

Introscope メトリック名のリソース部 Investigator Workstation のエージェント ノードの下の最初のノードに対応しています。

デフォルト

IISWebServer

## SUBRESOURCE

Introscope メトリック名のサブリソース部です。これは、上記で指定した Resource ノードの下のサブノードに対応しています。

デフォルト

<空>

## LINE\_FORMAT

IIS Web サーバがログに記録する行の正確な形式を定義します。

行の形式のサブプロパティを、LINE\_FORMAT サブプロパティの表で説明します。

デフォルト

該当なし

## LINE\_FORMAT サブプロパティ

以下の表では、プロパティ LINE\_FORMAT、およびそのすべてのサブプロパティについて説明します。

以下の LINE\_FORMAT のプロパティは、IIS Web サーバによってログ ファイルに記録されたフィールドと正確に一致する必要があります。一致しない場合、iisLogReader.pl プラグインは一部またはすべてのメトリックをレポートできなくなります。

IIS Web サーバの正確なフィールドは、IIS 管理ツール (または古いバージョンの IIS では Web ページ) 内で設定されます。[IIS4.0 および IIS5.0 で IIS Web サーバ ログのログ記録フィールドを設定](#) (P. 54) できます。

デフォルトでは、IIS Web サーバは以下のフィールドにのみログを記録します。

- 時間
- クライアント IP アドレス
- メソッド
- URI ステム
- プロトコルの状態

一致させるには、デフォルトの `iisLogReader.cfg` ファイルでもこれら 5 つの `LINE_FORMAT` サブプロパティのみを有効にします。

初期状態の `iisLogReader.pl` プラグインでは、IIS Web サーバ ログ設定のデフォルト値を前提としています。このプラグインは、`Bytes Transferred` を除くすべてのメトリックをレポートできます。`Bytes Transferred` については、IIS Web サーバによって `BytesSent` データ フィールドにログが記録され、`iisLogReader.cfg` サブプロパティ `BytesSent` のコメント化が解除 (有効化) される必要があります。IIS Web サーバ、および `iisLogReader.cfg` の `BytesSent` の有効化に失敗した場合、`Bytes Transferred` メトリックは CA Introscope® にレポートされません。

サブプロパティ行がコメント化解除されている場合、`iisLogReader.pl` プラグインは、IIS Web サーバ ログファイル (`LOGFILEDIR` および `LOGFILE` プロパティによって指定された) に該当のフィールドがあると想定します。反対に、このサブプロパティ行がコメント化されている場合、`iisLogReader.pl` プラグインは、ログファイルでこのフィールドを検索しません。このサブプロパティ行の順序が左から右の順で一致済みのときは、IIS Web サーバはデータをログに記録します (フィールドが IIS Web サーバで有効化された場合)。順序は変更しないでください。

## DELIMITER

空白 (スペース、タブ) の連続

デフォルト

¥s+



## FIELDS

該当なし

デフォルト

該当なし

## FIELDS->Date

アクティビティが発生した日付。YYYY-MM-DD。

デフォルト

¥d¥d¥d¥d-¥d¥d-¥d¥d

## FIELDS->Time

デフォルトで有効です。

アクティビティが発生した時間。グリニッジ標準時での 24 時間制の HH:MM:SS。

デフォルト

¥d¥d¥d¥d-¥d¥d-¥d¥d

## FIELDS->ClientIPAddress

デフォルトで有効です。

使用するサーバにアクセスしたクライアントの IP アドレス。ピリオドで区切られた 4 つの数字。

デフォルト

¥d+¥.¥d+¥.¥d+¥.¥d+

#### FIELDS->UserName

使用するサーバにアクセスしたユーザの名前。円記号、および英数字の任意の組み合わせ。

デフォルト

`[-¥¥¥w]+`

#### FIELDS->ServiceName

クライアントコンピュータ上で実行中だったインターネットサービス。英数字の任意の組み合わせ。

デフォルト

`¥w+`

#### FIELDS->ServerName

ログエントリが生成されたサーバの名前。ピリオド、ハイフン、英数字の任意の組み合わせ。

デフォルト

`[-.¥w]+`

#### FIELDS->ServerIPAddress

ログエントリが生成されたサーバの IP アドレス。ピリオドで区切られた 4 つの数字。

デフォルト

`¥d+¥.¥d+¥.¥d+¥.¥d+`

#### FIELDS->ServerPort

クライアントが接続するポート番号。1桁以上の数字。

デフォルト

`¥d+`

## FIELDS->Method

デフォルトで有効です。

クライアントが実行を試行したアクション（たとえば、GET コマンド）。

デフォルト

GET|HEAD|PUT|POST|DELETE

## FIELDS->URISem

デフォルトで有効です。

HTML ページ、CGI プログラム、またはスクリプトなどにアクセスしたりソース。スペース無しのシーケンス。

デフォルト

¥S+

## FIELDS->URIQuery

クライアントが実行を試行したクエリ、つまり、クライアントが一致を検索した 1 つ以上の検索文字列（ある場合）。スペース無しのシーケンス。

デフォルト

¥S+

## FIELDS->ProtocolStatus

デフォルトで有効です。

HTTP を使用したアクションのステータス。

デフォルト

¥d+

#### FIELDS->Win32Status

Windows で使用されるアクションのステータス。1桁以上の数字。

デフォルト

¥d+

#### FIELDS->BytesSent

サーバが送信したバイト数の合計。1桁以上の数字。

注: IIS では、このフィールドはデフォルトで有効ではありません。「転送バイト数」メトリックを収集するにはオンになっている必要があります。

デフォルト

¥d+

#### FIELDS->BytesReceived

サーバが受信したバイト数の合計。1桁以上の数字。

デフォルト

¥d+

#### FIELDS->TimeTaken

アクションにかかった時間の長さ（ミリ秒単位）。1桁以上の数字。

デフォルト

¥d+

#### FIELDS->ProtocolVersion

クライアントが使用するプロトコル（HTTP、FTP）のバージョン。HTTP の場合、これは、HTTP 1.0 か HTTP 1.1 のいずれかです。英数字、スラッシュ、およびピリオドのシーケンスです。

デフォルト

[¥w/.]+

## FIELDS->Host

クライアントのホストです。英数字、ピリオド、コロンのシーケンスです。

デフォルト

[¥w.:]+

## FIELDS->UserAgent

クライアント上で使用されるブラウザ。スペース無しのシーケンス。

デフォルト

¥S+

## FIELDS->Cookie

送信または受信したクッキーの内容（ある場合）。スペース無しのシーケンス。

デフォルト

¥S+

## FIELDS->Referer

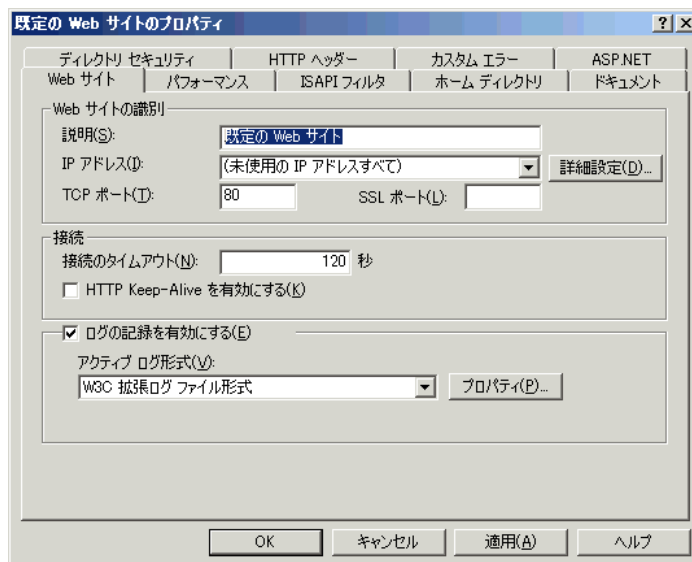
ユーザを現在のサイトに移動させたサイト。スペース無しのシーケンス。

デフォルト

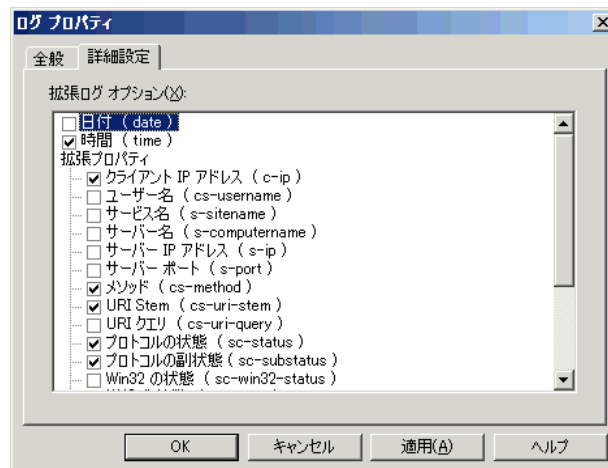
¥S+

## IIS4.0 および IIS5.0 での IIS Web サーバ ログ記録の設定

1. インターネット インフォメーション サービス マネージャを起動します。
2. 左側で、使用するマシンのツリー ノードを展開します。  
以下の 3 つのノードが表示されます。
  - FTP サイト
  - Web サイト
  - SMTP 仮想サーバ
3. [Web サイト] ノードを右クリックし、[プロパティ] を選択します。  
[プロパティ] ダイアログ ボックスが表示されます。



4. [Web サイト] タブを選択し、以下の設定を確認します。
  - [ログ記録を有効にする] チェックボックスがオン。
  - [アクティブ ログ形式] ドロップダウン リストで [W3C 拡張ログ ファイル形式] がオン。
5. [プロパティ] をクリックします。  
[拡張ログのプロパティ] ダイアログ ボックスが表示されます。



6. [全般プロパティ] タブで、ログ ファイル ディレクトリを指定します。*iisLogReader.cfg* ファイル内の LOGFILEDIR プロパティと一致している必要があります。
7. [拡張プロパティ] タブの以下のフィールドは、*iisLogReader.pl* プラグインがそのメトリックのすべてをレポートするのに必要/必須なものです。
  - 時間
  - クライアント IP アドレス
  - メソッド
  - URI ステム
  - プロトコルの状態
  - 送信バイト数 (転送バイト数を表示するために必要です)

有効になっている (チェックされている) すべてのフィールドは、**LINE\_FORMAT->FIELDS** セクションの下の *iisLogReader.cfg* ファイルで有効化 (コメント化解除) されたフィールドに一致する必要があります。*iisLogReader.cfg* ファイルのこの部分は、*iisLogReader.pl* プラグインにこのログファイルで前提となる正しい行の形式を知らせます。不一致がある場合、*iisLogReader.pl* プラグインは、メトリックをレポートできないか、または正しくないメトリックをレポートします。





# 第 3 章: EPAgent デフォルト プラグイン メトリック

---

この章では、EPAgent に付属するデフォルトのプラグインによって生成されるメトリックについて説明します。

すべてのメトリックは、システムクロックにより設定された時間を使用して計算されています。システムクロックがトランザクション処理中にリセットされた場合、そのトランザクションでレポートされた経過時間は誤っている可能性があります。

このセクションには、以下のトピックが含まれています。

[無効な文字の置き換え \(P. 57\)](#)

[IIS ログリーダーメトリック \(P. 58\)](#)

[HTTP/HTTPS サービス可用性メトリック \(P. 63\)](#)

[Solaris カーネル統計情報 \(Kstat\) メトリック \(P. 64\)](#)

[プロセスの可用性メトリック \(P. 65\)](#)

[ディスク統計情報 \(Windows\) メトリック \(P. 65\)](#)

[ディスク統計情報 \(Solaris\) メトリック \(P. 70\)](#)

[Apache ログメトリック \(P. 74\)](#)

[WebSphere ログメトリック \(P. 79\)](#)

[WebLogic ログメトリック \(P. 86\)](#)

## 無効な文字の置き換え

すべてのリソース名、サブリソース名、メトリック名は、Introscope のメトリック名として使用できない無効な文字がチェックされます。特に、パイプ文字 (|) およびコロン (:) は、予約済みの文字です。ユーザまたはシステムが作成した名前にこれらの文字が含まれる場合は、以下のよう自動的に置き換えられます。

| 元の文字 | 置き換える文字 | 説明                  |
|------|---------|---------------------|
|      | _       | パイプはアンダースコアに置き換えます。 |
| :    | ^       | コロンはキャレットに置き換えます。   |

## IIS ログリーダー メトリック

以下のセクションでは、*iisLogReader.pl* プラグインによって生成されるメトリックについて説明します。これらのメトリックのカウントは間隔あたりの数値で、最後にプラグインがログファイルをチェックしてからのものです。また、IISWebServer <サブリソース> メトリックについても説明します。

### Total Page Hits

#### CA Introscope® データ型

IntCounter

単位

1 (ヒット)

デフォルト

オン

### Total Visitors Count

#### CA Introscope® データ型

IntCounter

単位

1 (ヒット)

デフォルト

オン

## Page Hits For <URL>

### CA Introscope® データ型

IntCounter

#### 単位

1 (ヒット)

デフォルト

オン

## Bytes Transferred

### CA Introscope® データ型

IntCounter

#### 単位

1 (MB)

デフォルト

オン

## ErrorCount\_400\_BadRequest

### CA Introscope® データ型

IntCounter

#### 単位

1 (エラー)

デフォルト

オン

## ErrorCount\_401\_Unauthorized

**CA Introscope® データ型**

IntCounter

**単位**

1 (エラー)

**デフォルト**

オン

## ErrorCount\_403\_Forbidden

**CA Introscope® データ型**

IntCounter

**単位**

1 (エラー)

**デフォルト**

オン

## ErrorCount\_404\_ServerConfiguration

**CA Introscope® データ型**

IntCounter

**単位**

1 (エラー)

**デフォルト**

オン

## ErrorCount\_500\_InternalServerError

**CA Introscope® データ型**

IntCounter

**単位**

1 (エラー)

**デフォルト**

オン

## ErrorMessage\_400\_BadRequest

**CA Introscope® データ型**

StringEvent

**単位**

該当なし

**デフォルト**

オフ

## ErrorMessage\_401\_Unauthorized

**CA Introscope® データ型**

StringEvent

**単位**

該当なし

**デフォルト**

オフ

## ErrorMessage\_403\_Forbidden

**CA Introscope® データ型**

StringEvent

**単位**

該当なし

デフォルト

オフ

## ErrorMessage\_404\_PageNotFound

**CA Introscope® データ型**

StringEvent

**単位**

該当なし

デフォルト

オフ

## ErrorMessage\_500\_InternalServerError

**CA Introscope® データ型**

StringEvent

**単位**

該当なし

デフォルト

オフ

## HTTP/HTTPS サービス可用性メトリック

以下のセクションでは、*httpSvcAvailability.pl* プラグインによって生成されるメトリックについて説明します。また、Remote Machine <ホスト名または IP アドレス>|サービス|HTTP<サブリソース> メトリックについても説明します。

### Available

#### CA Introscope® データ型

IntCounter

#### 単位

0 または 1

#### デフォルト

オン

### 応答時間

#### CA Introscope® データ型

IntCounter

#### 単位

秒

#### デフォルト

オン

## Average Throughput

CA Introscope® データ型

IntCounter

単位

KB/ミリ秒

デフォルト

オン

## Solaris カーネル統計情報 (Kstat) メトリック

以下に *kstat.pl* プラグインによって生成されるメトリックを示します。

*kstat* プラグインによって生成される特定のメトリックの名前は、監視対象の Solaris システム、Solaris 命名コンセプトの両方、および、Environment Performance Agent (*IntroscopeEPAgent.properties* で設定されます) による *kstat.pl* プラグインのコマンドライン呼び出しにより変化します。

## Solaris Kernel <クラス>|<モジュール>|<名前> メトリック

以下のセクションでは、Solaris Kernel <クラス>|<モジュール>|<名前> メトリックについて説明します。

### <統計情報>

CA Introscope® データ型

IntCounter

単位

多様

デフォルト

オン



## プロセスの可用性メトリック

*processAvailability.pl* プラグインによって生成されるメトリックを以下に示します。

以下のセクションでは、**RunningProcesses<サブリソース>** メトリックについて説明します。

### Count

#### CA Introscope® データ型

IntCounter

#### 単位

1 (プロセス)

#### デフォルト

オン

## ディスク統計情報 (Windows) メトリック

*<EPA\_Home>/windows/diskStats.pl* プラグインによって生成されるメトリックを以下に示します。

以下の各セクションでは、**RunningProcesses <サブリソース>** メトリックについて説明します。

詳細:

[Disk Reads/sec](#) (P. 66)

[Disk Writes/sec](#) (P. 66)

[Disk Read Bytes/sec](#) (P. 67)

[Disk Write Bytes/sec](#) (P. 67)

[Current Disk Queue Length](#) (P. 67)

[Average Disk Read Time \(ms\)](#) (P. 68)

[Average Disk Write Time \(ms\)](#) (P. 68)

[Average Disk Bytes/Read](#) (P. 68)

[Average Disk Bytes/Write](#) (P. 69)

[Free Disk Space \(mb\)](#) (P. 69)

[Free Disk Space \(%\)](#) (P. 69)

## Disk Reads/sec

**CA Introscope® データ型**

IntCounter

単位

1 (カウント)

デフォルト

オン

## Disk Writes/sec

**CA Introscope® データ型**

IntCounter

単位

1 (カウント)

デフォルト

オン

## Disk Read Bytes/sec

**CA Introscope® データ型**

IntCounter

**単位**

1 バイト

デフォルト

オン

## Disk Write Bytes/sec

**CA Introscope® データ型**

IntCounter

**単位**

1 バイト

デフォルト

オン

## Current Disk Queue Length

**CA Introscope® データ型**

IntCounter

**単位**

1 (要求)

デフォルト

オン

## Average Disk Read Time (ms)

**CA Introscope® データ型**

IntCounter

**単位**

1 ミリ秒

デフォルト

オン

## Average Disk Write Time (ms)

**CA Introscope® データ型**

IntCounter

**単位**

1 ミリ秒

デフォルト

オン

## Average Disk Bytes/Read

**CA Introscope® データ型**

IntCounter

**単位**

1 バイト

デフォルト

オン

## Average Disk Bytes/Write

**CA Introscope® データ型**

IntCounter

**単位**

1 バイト

デフォルト

オン

## Free Disk Space (mb)

**CA Introscope® データ型**

IntCounter

**単位**

1 MB

デフォルト

オン

## Free Disk Space (%)

**CA Introscope® データ型**

IntCounter

**単位**

1 (プロセス)

デフォルト

オン

## ディスク統計情報 (Solaris) メトリック

<EPAgent\_Home>/solaris/diskStats.pl プラグインによって生成されるメトリックを以下に示します。

以下の各セクションでは、ディスク <サブリソース> メトリックについて説明します。

詳細:

[Disk Time \(%\)](#) (P. 70)

[Transaction Waiting Time \(%\)](#) (P. 71)

[Average # Transaction Active](#) (P. 71)

[Average # Transaction Waiting](#) (P. 71)

[Disk Reads/sec](#) (P. 72)

[Disk Writes/sec](#) (P. 72)

[KB Read per Second](#) (P. 72)

[KB Written per Second](#) (P. 73)

[Average Service Time in Waiting Queue \(ms\)](#) (P. 73)

[Average Service Time for Active Transactions \(ms\)](#) (P. 73)

[Free Disk Space \(mb\)](#) (P. 74)

[Free Disk Space \(%\)](#) (P. 74)

### Disk Time (%)

**CA Introscope® データ型**

IntCounter

単位

1 パーセント

デフォルト

オン

## Transaction Waiting Time (%)

**CA Introscope® データ型**

IntCounter

**単位**

1 パーセント

デフォルト

オン

## Average # Transaction Active

**CA Introscope® データ型**

IntCounter

**単位**

1 トランザクション

デフォルト

オン

## Average # Transaction Waiting

**CA Introscope® データ型**

IntCounter

**単位**

1 トランザクション

デフォルト

オン

## Disk Reads/sec

**CA Introscope® データ型**

IntCounter

**単位**

1 (カウント)

デフォルト

オン

## Disk Writes/sec

**CA Introscope® データ型**

IntCounter

**単位**

1 (カウント)

デフォルト

オン

## KB Read per Second

**CA Introscope® データ型**

IntCounter

**単位**

1 バイト

デフォルト

オン



## KB Written per Second

**CA Introscope® データ型**

IntCounter

**単位**

1 バイト

デフォルト

オン

## Average Service Time in Waiting Queue (ms)

**CA Introscope® データ型**

IntCounter

**単位**

1 ミリ秒

デフォルト

オン

## Average Service Time for Active Transactions (ms)

**CA Introscope® データ型**

IntCounter

**単位**

1 ミリ秒

デフォルト

オン

## Free Disk Space (mb)

CA Introscope® データ型

IntCounter

単位

1 MB

デフォルト

オン

## Free Disk Space (%)

CA Introscope® データ型

IntCounter

単位

1 パーセント

デフォルト

オン

## Apache ログ メトリック

*apacheLogReader.pl* プラグインによって生成されるメトリックを以下に示します。

以下の各セクションでは、ApacheWebServer <サブリソース> メトリックについて説明します。

詳細:

[PageHitCount](#) (P. 75)

[ErrorCount\\_400\\_BadRequest](#) (P. 75)

[ErrorCount\\_401\\_Unauthorized](#) (P. 76)

[ErrorCount\\_403\\_Forbidden](#) (P. 76)

[ErrorCount\\_404\\_PageNotFound](#) (P. 76)

[ErrorCount\\_500\\_ServerConfiguration](#) (P. 77)

[ErrorMessage\\_400\\_BadRequest](#) (P. 77)

[ErrorMessage\\_401\\_Unauthorized](#) (P. 77)

[ErrorMessage\\_403\\_Forbidden](#) (P. 78)

[ErrorMessage\\_404\\_PageNotFound](#) (P. 78)

[ErrorMessage\\_500\\_ServerConfiguration](#) (P. 78)

## PageHitCount

**CA Introscope® データ型**

IntCounter

単位

1 (ヒット)

デフォルト

オン

## ErrorCount\_400\_BadRequest

**CA Introscope® データ型**

IntCounter

単位

1 (エラー)

デフォルト

オン

## ErrorCount\_401\_Unauthorized

**CA Introscope® データ型**

IntCounter

**単位**

1 (エラー)

**デフォルト**

オン

## ErrorCount\_403\_Forbidden

**CA Introscope® データ型**

IntCounter

**単位**

1 (エラー)

**デフォルト**

オン

## ErrorCount\_404\_PageNotFound

**CA Introscope® データ型**

IntCounter

**単位**

1 (エラー)

**デフォルト**

オン

## ErrorCount\_500\_ServerConfiguration

**CA Introscope® データ型**

IntCounter

**単位**

1 (エラー)

**デフォルト**

オン

## ErrorMessage\_400\_BadRequest

**CA Introscope® データ型**

StringEvent

**単位**

1 (エラー)

**デフォルト**

オン

## ErrorMessage\_401\_Unauthorized

**CA Introscope® データ型**

StringEvent

**単位**

1 (エラー)

**デフォルト**

オン

## ErrorMessage\_403\_Forbidden

**CA Introscope® データ型**

StringEvent

**単位**

1 (エラー)

デフォルト

オン

## ErrorMessage\_404\_PageNotFound

**CA Introscope® データ型**

StringEvent

**単位**

1 (エラー)

デフォルト

オン

## ErrorMessage\_500\_ServerConfiguration

**CA Introscope® データ型**

StringEvent

**単位**

1 (エラー)

デフォルト

オン

## WebSphere ログ メトリック

*websphereLogReader.pl* プラグインによって生成されるメトリックを以下に示します。

以下の各セクションでは、WebSphereApplicationServer <サブリソース> メトリックについて説明します。

詳細:

[MessageCount\\_Audit](#) (P. 79)

[MessageCount\\_Warning](#) (P. 80)

[MessageCount\\_EventDebug](#) (P. 80)

[MessageCount\\_Error](#) (P. 80)

[MessageCount\\_Fatal](#) (P. 81)

[MessageText\\_Audit](#) (P. 81)

[MessageText\\_Warning](#) (P. 81)

[MessageText\\_EventDebug](#) (P. 82)

[MessageText\\_Error](#) (P. 82)

[MessageText\\_Fatal](#) (P. 82)

### MessageCount\_Audit

CA Introscope® データ型

IntCounter

単位

1 (ヒット)

デフォルト

オン

## MessageCount\_Warning

**CA Introscope® データ型**

IntCounter

**単位**

1 (ヒット)

**デフォルト**

オン

## MessageCount\_EventDebug

**CA Introscope® データ型**

IntCounter

**単位**

1 (ヒット)

**デフォルト**

オン

## MessageCount\_Error

**CA Introscope® データ型**

IntCounter

**単位**

1 (ヒット)

**デフォルト**

オン



## MessageCount\_Fatal

**CA Introscope® データ型**

IntCounter

**単位**

1 (ヒット)

**デフォルト**

オン

## MessageText\_Audit

**CA Introscope® データ型**

IntCounter

**単位**

1 (ヒット)

**デフォルト**

オン

## MessageText\_Warning

**CA Introscope® データ型**

IntCounter

**単位**

1 (ヒット)

**デフォルト**

オン

## MessageText\_EventDebug

**CA Introscope® データ型**

IntCounter

**単位**

1 (ヒット)

**デフォルト**

オン

## MessageText\_Error

**CA Introscope® データ型**

IntCounter

**単位**

1 (ヒット)

**デフォルト**

オン

## MessageText\_Fatal

**CA Introscope® データ型**

IntCounter

**単位**

1 (ヒット)

デフォルト

オン

**CA Introscope® データ型**

StringEvent

**単位**

該当なし

デフォルト

オン

**CA Introscope® データ型**

IntCounter

**単位**

1 (メッセージ)

デフォルト

オン

**CA Introscope® データ型**

IntCounter

**単位**

1 (メッセージ)

デフォルト

オン

**CA Introscope® データ型**

IntCounter

単位

1 (メッセージ)

デフォルト

オン

**CA Introscope® データ型**

IntCounter

単位

1 (メッセージ)

デフォルト

オン

**CA Introscope® データ型**

IntCounter

単位

1 (メッセージ)

デフォルト

オン

**CA Introscope® データ型**

StringEvent

単位

該当なし

デフォルト

オフ

**CA Introscope® データ型**

StringEvent

**単位**

該当なし

デフォルト

オフ

**CA Introscope® データ型**

StringEvent

**単位**

該当なし

デフォルト

オフ

**CA Introscope® データ型**

StringEvent

**単位**

該当なし

デフォルト

オフ

**CA Introscope® データ型**

StringEvent

**単位**

該当なし

デフォルト

オフ

## WebLogic ログ メトリック

以下の表に、*weblogicLogReader.p1* プラグインによって生成されるメトリックを示します。

以下の各セクションでは、WebLogicApplicationServer <サブリソース> メトリックについて説明します。

詳細:

[ServerVersion](#) (P. 86)

[MessageCount Warning](#) (P. 87)

[MessageCount Error](#) (P. 87)

[MessageCount Notice](#) (P. 87)

[MessageCount Critical](#) (P. 88)

[MessageCount Alert](#) (P. 88)

[MessageCount Emergency](#) (P. 88)

[MessageText Warning](#) (P. 89)

[MessageText Error](#) (P. 89)

[MessageText Notice](#) (P. 89)

[MessageText Critical](#) (P. 90)

[MessageText Alert](#) (P. 90)

[MessageText Emergency](#) (P. 90)

### ServerVersion

**CA Introscope® データ型**

StringEvent

単位

該当なし

デフォルト

オン

## MessageCount\_Warning

**CA Introscope® データ型**

IntCounter

**単位**

1 (メッセージ)

**デフォルト**

オン

## MessageCount\_Error

**CA Introscope® データ型**

IntCounter

**単位**

1 (メッセージ)

**デフォルト**

オン

## MessageCount\_Notice

**CA Introscope® データ型**

IntCounter

**単位**

1 (メッセージ)

**デフォルト**

オン

## MessageCount\_Critical

**CA Introscope® データ型**

IntCounter

**単位**

1 (メッセージ)

**デフォルト**

オン

## MessageCount\_Alert

**CA Introscope® データ型**

IntCounter

**単位**

1 (メッセージ)

**デフォルト**

オン

## MessageCount\_Emergency

**CA Introscope® データ型**

IntCounter

**単位**

1 (メッセージ)

**デフォルト**

オン



## MessageText\_Warning

**CA Introscope® データ型**

StringEvent

**単位**

該当なし

デフォルト

オフ

## MessageText\_Error

**CA Introscope® データ型**

StringEvent

**単位**

該当なし

デフォルト

オフ

## MessageText\_Notice

**CA Introscope® データ型**

StringEvent

**単位**

該当なし

デフォルト

オフ

## MessageText\_Critical

**CA Introscope® データ型**

StringEvent

**単位**

該当なし

デフォルト

オフ

## MessageText\_Alert

**CA Introscope® データ型**

StringEvent

**単位**

該当なし

デフォルト

オフ

## MessageText\_Emergency

**CA Introscope® データ型**

StringEvent

**単位**

該当なし

デフォルト

オフ

## 第 4 章: EAgent のプロパティ

---

この付録では、Environment Performance Agent をカスタマイズする際に使用するプロパティ ファイルおよび設定について説明します。その他の詳細情報は、プロパティ ファイル内で参照できます

(`<EAgent_Home>/epagent/IntroscopeEAgent.properties` にあります)。

注: 以下に説明するファイルを Windows マシンで開いてパスを追加する場合、各円記号にもう 1 つ円記号を付加して (すべての円記号を二重にして) エスケープ処理する必要があります。たとえば、

「`C:¥¥Introscope¥¥lib¥¥Agent.jar`」のようにします。

このセクションには、以下のトピックが含まれています。

[IntroscopeEAgent.properties ファイル](#) (P. 91)

[Agent メトリックのエイジング](#) (P. 93)

[Case-Sensitive メトリック](#) (P. 98)

[ChangeDetector の設定](#) (P. 98)

[Enterprise Manager 接続](#) (P. 103)

[EAgent Configuration](#) (P. 104)

[拡張ディレクトリ](#) (P. 105)

[ログ](#) (P. 106)

[プロセス](#) (P. 107)

[リモート設定](#) (P. 108)

[ステートフルプラグイン](#) (P. 109)

[ステートレスプラグイン](#) (P. 110)

### IntroscopeEAgent.properties ファイル

EAgent の設定は、CA Introscope® エージェント プロファイルで定義される設定と同じタイプのものを使用します。

## エージェント名およびプロセス名の設定

2つのプロパティ `introscope.agent.agentName`、および `introscope.agent.customProcessName` は EPAgent の名前とそのプロセスの名前を定義します。両方とも、完全修飾エージェント名（ドメイン | ホスト | プロセス | エージェント名）の一部として使用され、複数のエージェントセッションにまたがる単一の EPAgent の情報を追跡するために使用されます。

これらのプロパティはオプションです。エージェント名が定義されていない場合は、「UnknownAgent」という名前になります。

### Process Name

CA Introscope® は、以下の基準で EPAgent の命名を行います。この名前付けの優先度は以下に示されている順序です。

- プロパティ `introscope.agent.customProcessName` によって定義される名前。
- アプリケーションのメインクラスの名前から類推される名前。
- `introscope.agent.defaultProcessName` によって設定される名前。

詳細:

[プロセス](#) (P. 107)

### エージェント名

`introscope.agent.agentName` プロパティは、EPAgent の名前を定義します。

CA Introscope® は、以下の基準で Environment Performance Agent の命名を行います。この名前付けの優先度は以下に示されている順序です。

- プロパティ [introscope.agent.agentName](#) (P. 108) によって定義される名前。
- `introscope.agent.agentName` で一意の名前が定義されていない場合、Environment Performance Agent は「UnknownAgent」と名付けられます。

## Agent メトリックのエイジング

エージェント メトリック エイジングは、エージェントのメモリ キャッシュから定期的にデッドメトリックを削除します。デッドメトリックとは、設定された時間内に新しいデータをレポートしていないメトリックを表します。古いメトリックを削除することで、エージェントのパフォーマンスを改善し、急増メトリックの危険性を回避することができます。

**注:** 急増メトリックは、不注意から、システムの処理能力を超える量のメトリックをレポートするようにエージェントが設定されている場合に発生します。非常に多くのメトリックがレポートされると、エージェントがアプリケーションサーバの性能に影響を与える可能性があり、極端な場合では、サーバがまったく機能しなくなる可能性があります。

同一のグループに存在するメトリックは、グループ内のすべてのメトリックが削除の対象であると判断される場合にのみ削除されます。現在、*BlamePointTracer* および *MetricRecordingAdministrator* メトリックのみがグループとして削除されます。その他のメトリックは個別に削除されます。

*MetricRecordingAdministrator* には、メトリック グループを作成、取得、または削除するための以下のインターフェースがあります。

- `getAgent().IAgent_getMetricRecordingAdministrator.addMetricGroup`  
文字列のコンポーネント、収集メトリックです。コンポーネント名は、メトリック グループのメトリック リソース名です。複数のメトリックがグループであるとみなされるためには、同じメトリック ノード下にある必要があります。メトリックは、`com.wily.introscope.spec.metric.AgentMetric` データ構造体のコレクションです。このコレクションには *AgentMetric* データ構造体のみ追加することができます。
- `getAgent().IAgent_getMetricRecordingAdministrator.getMetricGroup`  
文字列のコンポーネントです。メトリック リソース名であるコンポーネント名に基づいて、メトリックのコレクションを取得することができます。
- `getAgent().IAgent_getMetricRecordingAdministrator.removeMetricGroup`  
文字列のコンポーネントです。メトリック グループは、メトリック リソース名であるコンポーネントに基づいて削除されます。

- `getAgent().IAgent_getDataAccumulatorFactory.isRemoved`

メトリックが削除されたかどうかを確認します。このインターフェースは、拡張機能にアキュムレータのインスタンスを保持する場合に使用します。メトリック エイジングによってアキュムレータが削除された場合、このインターフェースを使用して無効な参照を保持することを防ぎます。

**重要:** `MetricRecordingAdministrator` インターフェースを使用する拡張機能を作成する場合は(たとえば、ほかの **CA Technologies** 製品で使用するため)、必ずアキュムレータの独自のインスタンスを削除してください。メトリックが長い間呼び出されなかったためにエイジアウトし、その後データがそのメトリックで使用できるようになった場合、古いアキュムレータインスタンスは新しいメトリック データ ポイントを作成しません。このような状況を回避するには、アキュムレータの独自のインスタンスを削除せず、`getDataAccumulatorFactory` インターフェースを代わりに使用してください。

## エージェント メトリック エイジングの設定

エージェント メトリック エイジングは、デフォルトでオンになっています。プロパティ [introscope.agent.metricAging.turnOn](#) (P. 95) を使用して、この機能をオフにするよう選択することもできます。 `IntroscopeAgent.profile` からこのプロパティを削除すると、エージェント メトリック エイジングはデフォルトでオフになります。

エージェント メトリック エイジングは、エージェントのハートビートで実行されます。ハートビートはプロパティ [introscope.agent.metricAging.heartbeatInterval](#) (P. 96) を使用して設定します。ハートビートの頻度は低く設定するようにしてください。ハートビートの頻度が高いと、エージェントおよび **CA Introscope®** のパフォーマンスに影響を与えます。

各ハートビートの中に、メトリックの特定のセットがチェックされます。これは、プロパティ [introscope.agent.metricAging.dataChunk](#) (P. 96) を使用すると設定できます。また、高い値はパフォーマンスに影響を与えるため、この値を低くすることも重要なことです。ハートビートごとにチェックされるメトリック数のデフォルト値は 500 です。削除の対象となるメトリックがないかどうか、500 メトリックのそれぞれがチェックされます。たとえば、このプロパティを 1 回のハートビートにつき 500 メトリックずつチェックするよう設定し、エージェントメモリには合計で 10,000 のメトリックがある場合、10,000 メトリックすべてをチェックするのに時間はかかりますが、パフォーマンスへの影響は小さくなります。しかし、このプロパティを大きな数字に設定すると、10,000 メトリックすべてをチェックする時間は短くなりますが、オーバーヘッドが大きくなってしまいう可能性があります。

メトリックが一定期間新しいデータを受信していないと、そのメトリックは削除の候補になります。この期間はプロパティ [introscope.agent.metricAging.numberTimeslices](#) (P. 97) を使用して設定できます。デフォルトでは、このプロパティは 3000 に設定されています。メトリックが削除の条件と一致すると、グループ内のすべてのメトリックがメトリック削除の候補になっているかどうかのチェックが実行されます。この要件も満たした場合、メトリックは削除されます。

## introscope.agent.metricAging.turnOn

エージェントメトリックエイジングをオンまたはオフにします。

### プロパティ設定

True または False

### デフォルト

True

### 例

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

### 注

このプロパティへの変更はただちに有効となり、管理対象アプリケーションを再起動する必要はありません。

## introscope.agent.metricAging.heartbeatInterval

メトリックを削除するかどうかをチェックする間隔（秒単位）を指定します。

### デフォルト

1800

### 例

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

### 注

このプロパティの変更を有効にするには、マネージドアプリケーションを再起動する必要があります。

## introscope.agent.metricAging.dataChunk

各間隔ごとにチェックされるメトリックの数を指定します。

### デフォルト

500

### 例

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

### 注

このプロパティへの変更はただちに有効となり、管理対象アプリケーションを再起動する必要はありません。



## introscope.agent.metricAging.numberTimeslices

このプロパティは、削除の候補にする前に、新しいデータを受信しなかったかどうかをチェックする間隔を指定します。

### デフォルト

3000

### 例

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

### 注

このプロパティへの変更はただちに有効になります。管理対象アプリケーションを再起動する必要はありません。

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

指定したメトリックを削除対象から除外します。エイジングから 1 つ以上のメトリックを除外するには、メトリック名またはメトリック フィルタをリストへ追加します。

### プロパティ設定

カンマ区切りのメトリックのリスト。メトリック名には、アスタリスク (\*) をワイルドカードとして使用できます。

### デフォルト

デフォルトは「Threads」で始まるメトリック名です (*Threads\**)。

### 例

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

### 注

このプロパティへの変更はただちに有効となり、管理対象アプリケーションを再起動する必要はありません。

## Case-Sensitive メトリック

以下のプロパティは、大文字と小文字を区別するかどうかを設定します。

### introscope.epagent.metricscasesensitive

メトリック名の大文字と小文字の区別を有効にします。

#### プロパティ設定

True または False

#### デフォルト

true

#### 例

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

#### 注

- デフォルトでは、コメント化されます。
- このプロパティを **true** に設定すると、メトリックでの大文字と小文字の区別が有効になります。
- プロパティを **false** に設定すると、大文字と小文字の区別がオフになります。

## ChangeDetector の設定

以下のプロパティで、EPAgent と ChangeDetector とのやり取りを設定します。

## introscope.changeDetector.enable

ChangeDetector を有効または無効に指定します。ChangeDetector を有効にするには、プロパティを `true` に設定します。デフォルトではコメント化され、`false` に設定されます。ChangeDetector を有効にする場合、追加の ChangeDetector 関連プロパティを設定する必要があります。

### プロパティ設定

True または False

### デフォルト

False

### 例

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

### 注

このプロパティの変更を有効にするには、マネージドアプリケーションを再起動する必要があります。

## introscope.changeDetector.rootDir

ChangeDetector ファイルのルートディレクトリを指定します。ルートディレクトリは、ChangeDetector がローカル キャッシュ ファイルを作成するフォルダです。

### プロパティ設定

ChangeDetector ファイルのルートディレクトリへのフルパス（テキスト文字列）。

### デフォルト

デフォルトパスは `c://sw//AppServer//wily//change_detector` です。

### 例

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

### 注

円記号をエスケープするには、この例のように円記号を使用します。

## introscope.changeDetector.isengardStartupWaitTimeInSec

エージェントが起動してから ChangeDetector が Enterprise Manager への接続を試行するまでの待機する秒数を設定します。このプロパティは、デフォルトでコメント化されています。

### デフォルト

デフォルトは 15 秒です。

### 例

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

## introscope.changeDetector.waitTimeBetweenReconnectInSec

Enterprise Manager への接続を再試行する前に、ChangeDetector が待機する秒数を指定します。このプロパティは、デフォルトでコメント化されています。

### デフォルト

デフォルトは 10 秒です。

### 例

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

## introscope.changeDetector.enableEPA

ChangeDetector EAgent が有効な場合、EAgent プラグインは XML 形式のデータ変更のデータソースとして使用できます。

### プロパティ設定

True または False

### デフォルト

true

### 例

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

### 注

デフォルトでは、コメント化されます。

### introscope.changeDetector.agentID

ChangeDetector で EPAgent の識別に使用されるテキスト文字列を設定します。

#### 例

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

#### 注

デフォルトでは、コメント化されます。

### introscope.changeDetector.profile

ChangeDetector データソース設定ファイルへの絶対パスまたは相対パスを設定します。このプロパティは、デフォルトでコメント化されています。

#### デフォルト

デフォルトは `ChangeDetector-config.xml` です。

#### 例

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

#### 注

円記号をエスケープするには、この例のように円記号を使用します。

## introscope.changeDetector.profileDir

データソース設定ファイルを含むディレクトリへの絶対パスまたは相対パスを指定します。このプロパティが設定される場合、このディレクトリ内のデータソース設定ファイルはすべて *introscope.changeDetector.profile* プロパティによって指定された任意のファイルに加えて使用されます。このプロパティは、デフォルトでコメント化されています。

### デフォルト

デフォルトは `changeDetector_profiles` です。

### 例

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

### 注

円記号をエスケープするには円記号を使用します。

## Enterprise Manager 接続

以下のプロパティは、Enterprise Manager への接続を設定します。

## introscope.agent.enterprisemanager.connectionorder

EPAgent がプライマリの Enterprise Manager から切断された場合に、エージェントが使用するバックアップ Enterprise Manager の接続順序を設定します。コンマ区切りのリストの形式で値を指定する必要があります。

### デフォルト

DEFAULT

### 例

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

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

デフォルトでエージェントが接続する Enterprise Manager を実行するコンピュータのホスト名を指定します。

### デフォルト

localhost

### 例

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

### 注

このプロパティの変更を有効にするには、マネージドアプリケーションを再起動する必要があります。

## EPAgent Configuration

以下のプロパティは、EPAgent 自体を設定します。

## introscope.epagent.config.networkDataPort

XML データを受信するために、EPAgent がネットワーク接続でリスンするポートを指定します。

### デフォルト

8000

### 例

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

### 注

このプロパティは、デフォルトでコメント化されています。



## introscope.epagent.config.httpServerPort

EPAgent が HTTP GET コマンドをリスンするポートを指定します。

### デフォルト

8080

### 例

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

### 注

このプロパティは、デフォルトでコメント化されています。

## 拡張ディレクトリ

以下のメトリックは、拡張ディレクトリを指定します。

## introscope.agent.extensions.directory

エージェントによってロードされるすべての拡張機能がある場所を指定します。ディレクトリへの絶対パスまたは相対パスを指定できます。絶対パスを指定しない場合、指定する値が *IntroscopeAgent.profiles* ファイルの場所を起点とする相対パスに解決されます。

### デフォルト

デフォルトの場所は <EPAgent\_Home>/epagent/ext ディレクトリの ext ディレクトリです。

### 例

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

### 注

このプロパティの変更を有効にするには、マネージドアプリケーションを再起動する必要があります。

## ログ

ログは以下のプロパティで設定します。

### log4j.logger.EPAgent

EPAgent ログの詳細度、およびログ ファイルの出力場所を指定します。

デフォルト

INFO、console

例

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

注

- *INFO* を *VERBOSE#com.wily.util.feedback.Log4JSeverityLevel* で置き換え、ログの詳細レベルを上げます。
- *console* を *logfile* で置き換え、コンソールの代わりに、ログ ファイルへ出力を送信します。

### log4j.appender.logfile.File

このプロパティは、エージェントのログに記録するログ レベルと出力場所の両方を制御します。

デフォルト

IntroscopeEPA.log

例

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

注

ログファイルの場所と名前の完全パス名を使用することができます。

## プロセス

プロセスは以下のプロパティで設定します。

### introscope.agent.customProcessName

監視対象のプロセスの名前を指定します。

デフォルト

EPAgentProcess

例

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

注

アプリケーションサーバ固有のインストーラの場合、アプリケーションサーバ名に設定します。

### introscope.agent.defaultProcessName

*introscope.Agent.customProcessName* が設定されておらず、かつアプリケーションのメインクラスから名前を類推できない場合に、デフォルトでエージェントに対応付ける名前を指定します。

デフォルト

UnknownProcess

例

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

注

デフォルトはインストーラによって異なることがあります。

### introscope.agent.agentName

Enterprise Manager および Workstation 内に表示される EAgent の名前を指定します。

デフォルト

EAgent

例

```
introscope.agent.agentName=EAgent
```

注

アプリケーション固有のインストーラの場合、UnknownAgent または <appservername>Agent のように、アプリケーションサーバの名前に設定します。

## リモート設定

以下のプロパティを使用すると、リモートで EAgent を設定することができます。

### introscope.agent.remoteagentconfiguration.enabled

EAgent のリモート設定を有効または無効にします。

プロパティ設定

True または False

デフォルト

False

例

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

注

このプロパティへの変更はただちに有効となり、管理対象アプリケーションを再起動する必要はありません。

## ステートフル プラグイン

以下のプロパティは、ステートフル プラグインを構成します。

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

実行するステートフル プラグインの名前を指定します。このプロパティは EPAgent プラグインを有効にする場合に必須です。

#### プロパティ設定

プラグイン名のカンマ区切りリスト。

#### デフォルト

APACHE,WEBLOGIC,WEBSPPHERE,IIS

#### 例

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

#### 注

- このプロパティは、デフォルトでコメント化されています。
- これらの名前はユーザ定義が可能ですが、以下の *introscope.epagent.plugins.stateful.<pluginname>.command* プロパティで使用する名前と一致させます。

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

*introscope.epagent.plugins.stateful.names* プロパティで指定されたプラグインを実行するコマンドを指定します。指定されたプラグイン名ごとにプロパティの行が 1 行必要です。

#### 例

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

#### 注

このプロパティは、デフォルトでコメント化されています。

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

プラグインは、「ステートレス」または「ステートフル」のいずれかです。ステートレス プラグインは、cron ライクのスケジュールまたは遅延のいずれかに基づいて EAgent によって定期的に行われます。ステートフル プラグインは、一度 EAgent によって起動され、永久に実行することを意図されています。ステートフル プラグインが終了した場合は、再起動されます。

プラグインは、「コマンド」サブキーを使って指定される外部プロセスであるか、または EAgent 仮想マシン内で実行される java クラスであるかのいずれであってもかまいません。Java クラス プラグインは「クラス」サブキーを使って指定され、完全修飾クラス名の後に任意のオプションの引数を指定します。たとえば、*my.package.path.Class arg1 arg2 arg3* などです。

#### 例

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

#### 注

このプロパティは、デフォルトでコメント化されています。

## ステートレス プラグイン

以下のプロパティは、ステートレス プラグインを設定します。

## introscope.epagent.plugins.stateless.names

実行するステートレス プラグインの名前を指定します。

デフォルト

DISK、ORACLE、HTTP、KSTAT

例

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

注

- このプロパティは、デフォルトでコメント化されています。
- これらの名前はユーザ定義が可能です。以下の *introscope.epagent.plugins.stateless.<pluginname>.command* プロパティで使用される名前と一致させます。

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

*introscope.epagent.plugins.stateless.names* プロパティで指定されたプラグインを実行するコマンドを指定します。指定されたプラグイン名ごとにプロパティの行が 1 行必要です。

例

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

注

このプロパティは、デフォルトでコメント化されています。

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

EPAgent VM 内でステートレス プラグインとして実行されるクラスを指定します。

注

このプロパティは、デフォルトでコメント化されています。

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

EPAgent が指定のプラグインを実行する頻度（秒単位）を指定します。指定されたプラグイン名ごとにプロパティの行が 1 行必要です。

#### デフォルト

900

#### 例

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

#### 注

このプロパティは、デフォルトでコメント化されています。

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

*introscope.epagent.stateless.<pluginname>.command* プロパティで指定されたプラグインを実行するスケジュールを指定します。入力には、CronTab スケジュール構文（UNIX システムで「man 5 crontab」を参照）になります。

#### 注

- このプロパティは、デフォルトでコメント化されています。
- このプロパティは、*introscope.epagent.stateless.<pluginname>.delayInSeconds* プロパティの代わりに使用します。両方のプロパティが存在する場合、EPAgent は終了します。



## introscope.epagent.config.stalledStatelessPluginTimeoutInSeconds

ストールしたステートレス プラグインを終了する前に待機する時間の長さを指定します。

### デフォルト

60 秒

### 例

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

### 注

このプロパティは、デフォルトでコメント化されています。

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

このプロパティは、ステートレス プラグインが値のレポートを停止した場合の動作を指定するために使用されます。

#### プロパティ設定

このプロパティでは、以下の値が有効です。

- **same** - メトリックがエイジアウトするまで継続して同じ値をレポートします。
- **zero** - 値 0 でメトリックをレポートします（メトリックタイプが `stringevent` または `timestamp` の場合、アクション「same」が実行されます）。
- **stop** - メトリック値のレポートを停止します。

#### デフォルト

same

#### 例

以下の `MATCHEDPROCS` の例では、`processAvailability` プラグインを使用して、一致パラメータによって指定された正規表現に名前が一致する実行中の各プロセス（ここでは、プロセス名が「f」で始まるもの）について 1 を返します。`metricNotReportedAction` パラメータの値「zero」は、プロセスが実行されなくなったときに、メトリック値を 0 にリセットするために使用されます。

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