

# vRealize Log Insight エージェントの操作

2022 年 5 月 24 日

vRealize Log Insight 8.1

最新の技術ドキュメントは、VMware の Web サイト (<https://docs.vmware.com/jp/>)

**VMware, Inc.**  
3401 Hillview Ave.  
Palo Alto, CA 94304  
[www.vmware.com](http://www.vmware.com)

**ヴィエムウェア株式会社**  
〒108-0023 東京都港区芝浦 3-1-1  
田町ステーションタワー N 18 階  
[www.vmware.com/jp](http://www.vmware.com/jp)

Copyright © 2022 VMware, Inc. All rights reserved. 著作権および商標情報。

# 目次

vRealize Log Insight エージェントの操作	5
<b>1 vRealize Log InsightAgent の概要</b>	<b>6</b>
<b>2 ログのローテーション スキームのタイプ</b>	<b>8</b>
<b>3 vRealize Log Insight エージェントのインストールまたはアップグレード</b>	<b>9</b>
エージェントのインストール ファイルのダウンロード	10
Windows エージェントのインストール	11
インストール ウィザードによる vRealize Log InsightWindows エージェントのインストールまたは更新	11
コマンド ラインからの vRealize Log InsightWindows エージェントのインストールまたは更新	12
複数マシンへの Log Insight Windows Agent の展開	14
vRealize Log InsightLinux Agent RPM パッケージのインストールまたは更新	17
vRealize Log Insight Linux Agent DEB パッケージのインストールまたは更新	19
Debian Linux 用のエージェントのインストールのカスタマイズ	19
Log Insight Linux Agent バイナリ パッケージのインストール	22
Linux での vRealize Log Insight エージェントのインストールのコマンド ライン オプション	24
vRealize Log Insight エージェントの自動更新	25
個々のエージェントでの自動更新の無効化と有効化	25
<b>4 vRealize Log Insight エージェントの構成</b>	<b>27</b>
Log Insight Windows Agent の構成	28
Log Insight Windows Agent のデフォルト構成	28
Windows イベント チャネルからのイベントの収集	31
ログ ファイルからのイベントの収集	35
Log Insight Windows Agent へのイベントの転送	40
Log Insight Linux Agent の構成	41
vRealize Log InsightLinux Agent のデフォルト構成	41
ログ ファイルからのイベントの収集	43
vRealize Log Insight エージェントからのイベントのフィルタリング	50
vRealize Log Insight エージェントからの情報の転送	52
ターゲット vRealize Log InsightServer の設定	52
エージェントの宛先の指定	55
vRealize Log Insight エージェントの構成の一元化	59
構成結合の例	60
エージェント構成に対する共通値の使用	61
ログの解析	63

ログ パーサの構成 63

**5 vRealize Log Insight エージェントのアンインストール 91**

- Log Insight Windows Agent のアンインストール 91
- Log Insight Linux Agent RPM パッケージのアンインストール 91
- Log Insight Linux Agent DEB パッケージのアンインストール 92
- Log Insight Linux Agent bin パッケージのアンインストール 92
- Log Insight Linux Agent bin パッケージの手動アンインストール 93

**6 vRealize Log Insight エージェントのトラブルシューティング 94**

- Log Insight Windows Agent のサポート バンドルの作成 94
- Log Insight Linux Agent のサポート バンドルの作成 95
- Log Insight Agents のログの詳細レベルの定義 95
- 管理 UI に Log Insight Agents が表示されない 96
- vRealize Log Insight Agent でイベントを送信しない 97
- Log Insight Windows Agent に対する送信例外ルールの追加 98
- Windows ファイアウォールでの Log Insight Windows Agent からの送信接続の許可 99
- Log Insight Windows Agent の一括展開に失敗しました 100
- Log Insight Agents による自己署名証明書の拒否 100
- vRealize Log Insight サーバによる暗号化されていないトラフィックの接続の拒否 101

# vRealize Log Insight エージェントの操作

『vRealize Log Insight エージェントの操作』では、vRealize™ Log Insight™ の Windows および Linux エージェントのインストールと構成方法について説明します。トラブルシューティングのヒントについても説明します。

この情報は、Log Insight Agents をインストール、構成、またはトラブルシューティングするユーザーを対象としています。記載されている情報は、Windows または Linux のシステム管理者としての経験があり、仮想マシンテクノロジーおよびデータセンターの運用に詳しい方を対象としています。

vRealize Log Insight サーバを使用してエージェントに構成クラスを作成する方法については、『vRealize Log Insight の管理』を参照してください。

# vRealize Log InsightAgent の概要

# 1

vRealize Log Insight エージェントはログ ファイルからイベントを収集し、vRealize Log Insight サーバまたは任意のサードパーティの Syslog 宛先に転送します。

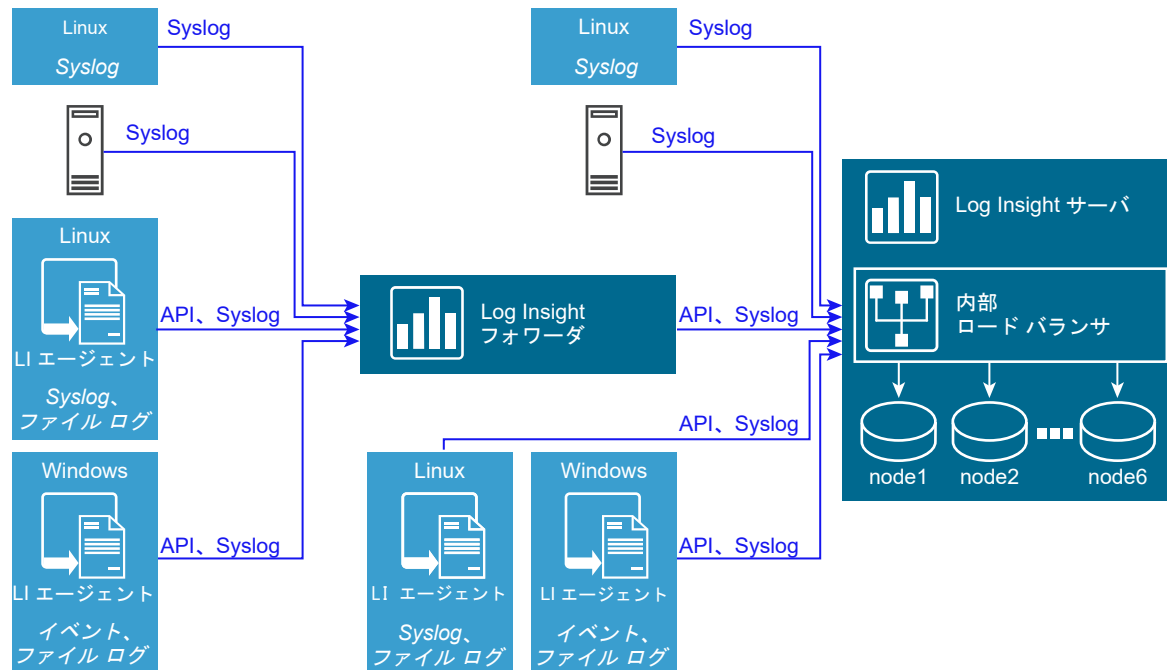
エージェントは、Syslog と vRealize Log Insight 取り込み API (cfapi プロトコル) をサポートし、Linux または Windows プラットフォームで使用できます。Web インターフェイスからエージェントを設定するには、サーバとクライアントの liagent.ini ファイルを使用するか、インストール中に設定を行います。

エージェントには次の機能が含まれます。

- 単一またはグループでの展開
- 自動アップグレード
- ログ メッセージに対して実行され、構造化データを抽出する解析。FileLog と WinLog コレクタまたはその両方のパーサを構成できます。
- 複数行のメッセージのサポート
- いくつかのログのローテーション スキームのネイティブ サポート
- クライアント側の圧縮、暗号化、およびイベントへのメタデータの追加機能を含む広範な取り込み API

vRealize Log Insight サーバは、一元化された構成管理およびエージェント グループの作成と管理をサポートします。

次の図は、エージェントのデプロイ構成の要素を示しています。



vRealize Log Insight フォワーダは、その主なジョブがリモートの宛先にイベントを転送することである vRealize Log Insight サーバの専用インスタンスです。通常、フォワーダとして使用するサーバ インスタンスは、クエリには使用されません。フォワーダは内部ロード バランサを使用し、使用しない場合は vRealize Log Insight サーバのように構造化されます。

エージェントは、エージェント自身の運用ログを記録します。Windows の場合、これらのログは `C:\ProgramData\VMware\Log Insight Agent\logs` ディレクトリにあります。Linux の場合、操作ログのパスは `/var/log/loginsight-agent/liagent_*.log` です。エージェントが再起動するか、ファイルが 10 MB になると、ログ ファイルがローテーションされます。ローテーションでは、組み合わせたファイルの合計が 50 MB という制限が保持されます。vRealize Log Insight エージェント自体は、エージェント ログを収集できません。

エージェントはリアルタイムのログの収集に使用されます。サポート バンドルを含む履歴ログ コレクションをインポートするには、vRealize Log Insight Importer を使用します。

Windows および Linux オペレーティング システム用に個別のインストール パッケージのダウンロードが提供されています。

Windows システムの場合、エージェントは Windows サービスとして実行し、インストール直後に起動します。エージェントは、アプリケーション ログ ファイルと Windows イベント チャネル、および関連する Windows システム イベントを収集するプールを監視します。収集されたイベントは、vRealize Log Insight サーバまたはサードパーティの Syslog の宛先に転送されます。

Linux システムの場合、エージェントはデーモンとして実行し、インストール直後に起動します。vRealize Log Insight Linux エージェントは、Linux マシン上のログ ファイルからイベントを収集し、vRealize Log Insight サーバまたは Syslog 宛先に転送します。Debian、Red Hat、および Linux バイナリのインストール パッケージが使用できます。

# vRealize Log Insight エージェントでサポートされるログのローテーションスキーム

## 2

vRealize Log Insight エージェントは、いくつかのログのローテーション スキームをサポートします。

ログのローテーションにより、ログ ファイルが無限に増えるのを防止します。いくつかのログのローテーション スキームがあり、それぞれ特定のユースケース セット用に設計されています。vRealize Log Insight には、次のスキームに対するネイティブ サポートが含まれています。

表 2-1. vRealize Log Insight エージェントによってサポートされるログのローテーション スキーム

ログのローテーション スキーム	説明
create-new	サイズまたは時間の制限値に達すると、新しいログ ファイルが作成されます。ロガー プロセスは、現在のログ ファイルへの書き込みを停止し、ログの出力先を新しく作成したファイルに変更します。既存のファイルが名前の変更あるいはその他の方法で処理されることはありません。
rename-recreate	サイズまたは時間の制限に達すると、logrotate などの外部ユーティリティによってログ ファイルの名前が変更されます。次にロガー プロセスはログ ファイルを以前の名前で作成します。
copy-truncate	サイズまたは時間の制限に達すると、logrotate などの外部ユーティリティによってログ ファイルがコピーされます。ログのプロセスでは、コピーされたファイルの名前が変更され、サイズが 0 になるように元のファイルが切り捨てられます。ロガー プロセスは、元のファイルへのログの書き込みを続行できます。



# vRealize Log Insight エージェントのインストールまたはアップグレード

## 3

サードパーティのログ管理システムがインストールされたマシンを含む、Windows または Linux マシンで vRealize Log Insight エージェントをインストールまたはアップグレードします。

エージェントはイベントを収集して vRealize Log Insight サーバに転送します。インストール中に、サーバ、ポート、プロトコルを設定するパラメータを指定するか、デフォルトの設定をそのまま使用します。

エージェントは、インストールと同じ方法でアップグレードするか、自動更新することができます。自動更新では、vRealize Log Insight の新しいバージョンを展開する際に、アップグレードをエージェントに適用します。詳細については、[vRealize Log Insight エージェントの自動更新](#)を参照してください。Linux の bin パッケージはアップグレードできません。

## ハードウェア サポート

vRealize Log Insight エージェントをインストールして実行する場合、x86 および x86\_64 アーキテクチャ、MMX、SSE、SSE2 および SSE3 命令セットをホスト/マシンがサポートするために必要な最小パラメータをハードウェアがサポートする必要があります。

## プラットフォーム サポート

オペレーティング システム	プロセッサ アーキテクチャ
Windows 7、Windows 8、Windows 8.1、Windows 10	x86_64、x86_32
Windows Server 2008、Windows Server 2008 R2、	x86_64、x86_32
Windows Server 2012、Windows Server 2012 R2、Windows Server 2016、および Windows Server 2019	x86_64
RHEL 5、RHEL 6、RHEL 7、および RHEL 8	x86_64、x86_32
SuSE Enterprise Linux (SLES) 11 SP3、SLES 12 SP1	x86_64
Ubuntu 14.04 LTS、Ubuntu 16.04 LTS、および Ubuntu 18.04	x86_64
VMware Photon、バージョン 1 リビジョン 2、バージョン 2、およびバージョン 3	x86_64

## Linux に関する注意

root 権限を持たないユーザーに Log Insight Linux エージェントのデフォルト インストールを実行して使用する場合、デフォルトの設定でデータ収集に問題が発生する場合があります。エージェントは、チャンネルへのサブスクリプションが失敗し、収集のファイルに読み取り権限がない場合でも警告を記録しません。「Inaccessible log file ... will try later」というメッセージが繰り返しログに追加されます。問題を引き起こしたり、ユーザー権限を変更したデフォルトの構成はコメントアウトできます。

rpm または DEB パッケージを使用して Linux エージェントをインストールする場合、liagentd という名前の init.d スクリプトがパッケージ インストールの一部としてインストールされます。bin パッケージでもスクリプトが追加されますが、登録はされません。スクリプトは手動で登録できます。

インストールが成功したことを確認するには、(/sbin/) service liagentd status コマンドを実行します。

この章には、次のトピックが含まれています。

- [エージェントのインストール ファイルのダウンロード](#)
- [Windows エージェントのインストール](#)
- [vRealize Log InsightLinux Agent RPM パッケージのインストールまたは更新](#)
- [vRealize Log Insight Linux Agent DEB パッケージのインストールまたは更新](#)
- [Debian Linux 用のエージェントのインストールのカスタマイズ](#)
- [Log Insight Linux Agent バイナリ パッケージのインストール](#)
- [Linux での vRealize Log Insight エージェントのインストールのコマンド ライン オプション](#)
- [vRealize Log Insight エージェントの自動更新](#)

## エージェントのインストール ファイルのダウンロード

vRealize Log Insight エージェントを設定する最初の手順は、お使いのプラットフォームのエージェント インストール パッケージをダウンロードすることです。

vRealize Log Insight サーバのエージェント ページからダウンロードしたすべてのパッケージには、パッケージ名の末尾にターゲットのホスト名が追加されます。MSI、RPM、および DEB の各エージェントの初期インストール中に、server.hostname が適用されます。構成ファイル内にホスト名がある場合や、ホスト名パラメータを使用してパッケージを実行している場合には、埋め込まれたサーバのホスト名は無視されます。

### 手順

- 1 vRealize Log Insight Web ユーザー インターフェイスの [管理] タブに移動します。
- 2 管理セクションで、[エージェント] をクリックします。
- 3 画面の一番下までスクロールして、[Log Insight エージェントのダウンロード] をクリックします。

- 4 ポップアップ メニューからインストール パッケージを選択し、[保存] をクリックしてダウンロードします。

オプション	説明
Windows MSI	Windows プラットフォーム用インストール パッケージ (32 ビット/64 ビット)
Linux RPM	Linux Red Hat, openSUSE (32 ビット/64 ビット) または VMware Photon Platform 用インストール パッケージ
Linux DEB	Debian Linux プラットフォーム用インストール パッケージ (32 ビット/64 ビット)
Linux BIN	Linux 用自己インストール パッケージ (32 ビット/64 ビット) パッケージ管理システムは必要ありません。

#### 次のステップ

ダウンロードされたファイルを使用して、vRealize Log Insight エージェントを展開します。

## Windows エージェントのインストール

インストール ウィザードまたはコマンド ラインからエージェントを 1 台のマシンにインストールするか、またはスクリプトを使用して複数のインスタンスに展開できます。

### Windows エージェントのアップグレード

インストールに使用した方法のいずれかを使用してアップグレード ファイルを適用することで、Windows エージェントをアップグレードできます。また、自動更新機能を使用して、エージェントをバックグラウンドでアップグレードすることもできます。

## インストール ウィザードによる vRealize Log InsightWindows エージェントのインストールまたは更新

インストール ウィザードを使用して、単一のマシン上で Windows エージェントをインストールまたはアップグレードすることができます。

#### 前提条件

- vRealize Log InsightWindows Agent の .msi ファイルのコピーがあることを確認します。[エージェントのインストール ファイルのダウンロード](#)を参照してください。
- Windows マシン上でインストールを実行し、サービスを開始する権限を持っていることを確認します。

#### 手順

- 1 vRealize Log InsightWindows エージェントをインストールする Windows マシンにログインします。
- 2 vRealize Log InsightWindows エージェントの .msi ファイルがあるディレクトリに変更します。
- 3 vRealize Log InsightWindows エージェントの .msi ファイルをダブルクリックし、使用許諾契約条件を承諾し、[次へ] をクリックします。

- 4 vRealize Log InsightServer の IP アドレスまたはホスト名を入力し、[インストール] をクリックします。

ウィザードが vRealize Log InsightWindows エージェントを LocalSystem サービス アカウントの下にある自動 Windows サービスとしてインストールまたは更新します。

- 5 [終了] をクリックします。

#### 次のステップ

liagent.ini ファイルを編集して vRealize Log Insight Windows エージェントを構成します。「[Log Insight Windows Agent の構成](#)」を参照してください。

## コマンド ラインからの vRealize Log InsightWindows エージェントのインストールまたは更新

コマンド ラインから Windows エージェントをインストールまたは更新できます。

デフォルトの設定を使用するか、またはサービス アカウントを指定できます。また、コマンド ラインのパラメータを使用して、サーバ、ポート、プロトコルの情報を指定できます。MSI コマンド ライン オプションについては、Microsoft Developer Network (MSDN) ライブラリの Web サイトを参照し、MSI コマンド ライン オプションを検索してください。

#### 前提条件

- vRealize Log InsightWindows Agent の .msi ファイルのコピーがあることを確認します。[エージェントのインストール ファイルのダウンロード](#)を参照してください。
- Windows マシン上でインストールを実行し、サービスを開始する権限を持っていることを確認します。
- サイレント インストール オプション /quiet または/qn を使用する場合は、管理者としてインストールを実行することを確認します。管理者でないユーザーがサイレント インストールを実行する場合、インストールでは管理者権限が要求されずに失敗します。ログ オプションとパラメータ /l\*xv\* *file\_name* は診断のために使用します。

#### 手順

- 1 vRealize Log InsightWindows Agent をインストールまたは更新する Windows マシンにログインします。
- 2 [コマンド プロンプト] ウィンドウを開きます。
- 3 vRealize Log InsightWindows エージェントの .msi ファイルがあるディレクトリに変更します。

- 4 次の形式のコマンドを実行して、デフォルト値を使用してインストールまたは更新します。 *version-build\_number* をバージョンとビルド番号で置き換えてください。

/quiet オプションを指定すると、コマンドがサイレントに実行されます。/lxv オプションを指定すると、現在のディレクトリにログ ファイルが作成されます。

```
Drive:\path-to-msi_file>VMware-Log-Insight-Agent-version-build_number.msi /
quiet /lxv* li_install.log
```

- 5 (オプション) vRealize Log Insight Windows Agent サービスを実行するユーザー サービス アカウントを指定します。

```
Drive:\path-to-msi_file>VMware-Log-Insight-Agent-*.msi SERVICEACCOUNT=domain\user
SERVICEPASSWORD=user_password
```

**注：** SERVICEACCOUNT パラメータで指定したアカウントには サービスとしてログオン 権限が与えられ、%ProgramData%\VMware\Log Insight Agent ディレクトリへの完全な書き込みアクセスが許可されます。指定したアカウントが存在しない場合には作成されます。ユーザー名は 20 文字以内にする必要があります。SERVICEACCOUNT パラメータを指定しない場合、vRealize Log Insight Windows Agent サービスは LocalSystem サービス アカウントでインストールまたは更新されます。

- 6 (オプション) 必要に応じて、次のコマンド ライン オプションに値を指定できます。

オプション	説明
<b>SERVERHOST=hostname</b>	vRealize Log Insight 仮想アプライアンスの IP アドレスまたはホスト名。
<b>SERVERPROTO=protocol</b>	エージェントが、イベントを vRealize Log Insight サーバに送信するために使用するプロトコル。設定可能な値は cfapi および syslog です。 デフォルトは cfapi です。
<b>SERVERPORT=portnumber</b>	エージェントが、イベントを vRealize Log Insight またはサードパーティのサーバに送信するために使用する通信ポート。デフォルトでは、エージェントは SSL とプロトコルに設定されたオプションに基づいて適切なポートを使用します。以下のリストに指定されたデフォルトのポート値を参照してください。ポート オプションは、これらのデフォルトと異なる場合にのみ指定する必要があります。 <ul style="list-style-type: none"> <li>■ SSL が有効になっている cfapi: 9543</li> <li>■ SSL が無効になっている cfapi: 9000</li> <li>■ SSL が有効になっている syslog: 6514</li> <li>■ SSL が無効になっている syslog: 514</li> </ul>
<b>SERVICEACCOUNT=account-name</b>	Log Insight Windows Agent サービスが実行されるユーザー サービス アカウント。  <b>注：</b> インストーラが正しく実行されるには、SERVICEACCOUNT で指定されたアカウントが、サービスとしてログイン 権限を持ち、%ProgramData%\VMware\Log Insight Agent ディレクトリに書き込みアクセス権を持っている必要があります。 SERVICEACCOUNT パラメータを指定しない場合、vRealize Log Insight Windows エージェントサービスは LocalSystem サービス アカウントでインストールされます。
<b>SERVICEPASSWORD=password</b>	ユーザー サービス アカウントのパスワード。

オプション	説明
<b>AUTOUPDATE={yes no}</b>	エージェントの自動更新を有効または無効にします。自動更新を完全に有効にするには、vRealize Log Insight サーバから自動更新を有効にする必要があります。デフォルトは yes です。
<b>LIAGENT_SSL={yes no}</b>	安全な接続を有効にします。SSL が有効になっている場合、エージェントは TLS 1.2 プロトコルを使用してサーバと通信します。デフォルトは yes です。

## 結果

このコマンドは、vRealize Log Insight Windows Agent を Windows サービスとしてインストールまたは更新します。Windows マシンを起動すると vRealize Log Insight Windows Agent サービスが開始されます。

## 次のステップ

設定したコマンド ライン パラメータが `liagent.ini` ファイルに正しく適用されていることを確認します。  
[「Log Insight Windows Agent の構成」](#) を参照してください。

## 複数マシンへの Log Insight Windows Agent の展開

Windows ドメイン内のターゲット コンピュータに対し、Log Insight Windows Agent の一括展開を実行できます。

## 手順

### 1 複数の vRealize Log Insight Windows エージェントを展開するための変換ファイルの作成

複数のエージェントの展開の一環として、展開の構成パラメータを指定する変換ファイルを作成する必要があります。エージェントをインストールするときに、.mst 変換ファイルが.msi ファイルに適用されます。パラメータには、エージェントの宛先サーバ、および Log Insight エージェント サービスをインストールおよび開始するための通信プロトコル、ポート、およびユーザー アカウントが含まれています。

### 2 vRealize Log Insight Windows エージェントの複数インスタンスの展開

Windows ドメインにあるターゲット コンピュータ上に vRealize Log Insight Windows エージェントのインスタンスを複数展開することができます。

## 複数の vRealize Log Insight Windows エージェントを展開するための変換ファイルの作成

複数のエージェントの展開の一環として、展開の構成パラメータを指定する変換ファイルを作成する必要があります。エージェントをインストールするときに、.mst 変換ファイルが.msi ファイルに適用されます。パラメータには、エージェントの宛先サーバ、および Log Insight エージェント サービスをインストールおよび開始するための通信プロトコル、ポート、およびユーザー アカウントが含まれています。

パラメータには、エージェントの宛先サーバ、および Log Insight エージェント サービスをインストールおよび開始するための通信プロトコル、ポート、およびユーザー アカウントが含まれています。

## 前提条件

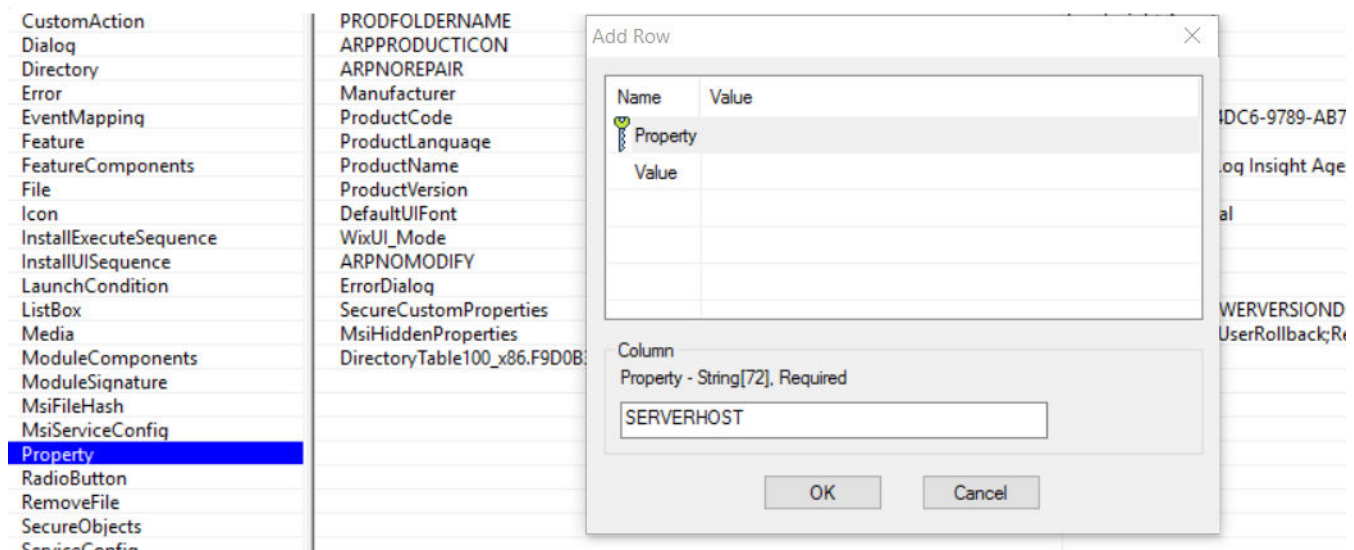
- vRealize Log Insight Windows の .msi ファイルのコピーがあることを確認します。[エージェントのインストール ファイルのダウンロード](#)を参照してください。

- Orca データベース エディタをダウンロードしてインストールします。<http://support.microsoft.com/kb/255905> を参照してください。

#### 手順

- 1 Orca エディタで vRealize Log Insight Windows エージェントの .msi ファイルを開き、[変換] - [新しい変換] を選択します。
- 2 [プロパティ] テーブルを編集し、カスタム インストールまたはアップグレードに必要なパラメータおよび値を追加します。

図 3-1. プロパティ テーブル



- a [プロパティ] をクリックします。
- b [テーブル] ドロップダウン メニューから [行の追加] を選択します。
- c [行の追加] ダイアログ ボックスにプロパティの名前と値を入力します。

パラメータが次の表に示されます。

パラメータ	説明
SERVERHOST	vRealize Log Insight 仮想アプライアンスの IP アドレスまたはホスト名。 デフォルトは <b>loginsight</b> です。
SERVERPROTO	エージェントが、イベントを vRealize Log Insight サーバに送信するために使用するプロトコル。設定可能な値は cfapi および syslog です。 デフォルトは cfapi です。

パラメータ	説明
SERVERPORT	<p>エージェントが、イベントを vRealize Log Insight またはサードパーティのサーバに送信するために使用する通信ポート。デフォルトでは、エージェントは SSL とプロトコルに設定されたオプションに基づいて適切なポートを使用します。以下のリストに指定されたデフォルトのポート値を参照してください。ポート オプションは、これらのデフォルトと異なる場合にのみ指定する必要があります。</p> <ul style="list-style-type: none"> <li>■ SSL が有効になっている cfapi: 9543</li> <li>■ SSL が無効になっている cfapi: 9000</li> <li>■ SSL が有効になっている syslog: 6514</li> <li>■ SSL が無効になっている syslog: 514</li> </ul>
SERVICEACCOUNT	<p>Log Insight Windows Agent サービスが実行されるユーザー サービス アカウント。</p> <p><b>注：</b> インストーラが正しく実行されるには、SERVICEACCOUNT で指定されたアカウントが、サービスとしてログイン 権限を持ち、%ProgramData%\VMware\Log Insight Agent ディレクトリに書き込みアクセス権を持っている必要があります。</p> <p>SERVICEACCOUNT パラメータを指定しない場合、vRealize Log Insight Windows エージェントサービスは LocalSystem サービス アカウントでインストールされます。</p>
SERVICEPASSWORD	ユーザー サービス アカウントのパスワード。
AUTOUPDATE	<p>エージェントの自動更新を有効または無効にします。自動更新を完全に有効にするには、vRealize Log Insight サーバから自動更新を有効にする必要があります。デフォルトは yes です。</p>
LIAGENT_SSL	<p>安全な接続を有効にします。SSL が有効になっている場合、エージェントは TLS 1.2 プロトコルを使用してサーバと通信します。デフォルトは yes です。</p>

3 [変換] - [変換の生成] を選択し、.mst 変換ファイルを保存します。

#### 次のステップ

.msi および .mst ファイルを使用して vRealize Log Insight Windows エージェントを展開します。

### vRealize Log Insight Windows エージェントの複数インスタンスの展開

Windows ドメインにあるターゲット コンピュータ上に vRealize Log Insight Windows エージェントのインスタンスを複数展開することができます。

クライアント マシンを 2 度再起動する必要がある理由については、<http://support.microsoft.com/kb/305293> を参照してください。

#### 前提条件

- ドメイン コントローラ上での管理者アカウントまたは管理権限のあるアカウントがあることを確認します。
- vRealize Log Insight Windows Agent の .msi ファイルのコピーがあることを確認します。[エージェントのインストール ファイルのダウンロード](#)を参照してください。
- <http://support.microsoft.com/kb/887405> および <http://support.microsoft.com/kb/816102> で説明されている手順を理解しておく必要があります。

#### 手順

1 ドメイン コントローラに管理者または管理権限のあるユーザーとしてログインします。



- 2 配布ポイントを作成し、vRealize Log Insight Windows エージェントの .msi ファイルを配布ポイントにコピーします。
- 3 グループ ポリシー管理コンソールを開き、グループ ポリシー オブジェクトを作成して vRealize Log Insight Windows エージェントの .msi ファイルを展開します。
- 4 ソフトウェアを展開するためのグループ ポリシー オブジェクトを編集し、パッケージを割り当てます。
- 5 (オプション) 展開の前に .mst ファイルを生成した場合は、[GPO プロパティ] ウィンドウの [編集] タブで .mst 構成ファイルを選択します。さらに、詳細メソッドを使用してグループ ポリシー オブジェクトを編集し、.msi パッケージを展開します。
- 6 (オプション) vRealize Log Insight Windows エージェントをアップグレードします。
  - a アップグレードの .msi ファイルを配布ポイントにコピーします。
  - b グループ ポリシー オブジェクトの [プロパティ] ウィンドウにある ☐ [アップグレード] タブをクリックします。
  - c [このパッケージによってアップグレードされるパッケージ] セクションにある、最初にインストールされたバージョンの .msi ファイルを追加します。
- 7 vRealize Log Insight Windows エージェントを、ドメイン ユーザーを含む特定のセキュリティ グループに展開します。
- 8 ドメイン コントローラのすべてのグループ ポリシー管理コンソールとグループ ポリシー管理エディタ ウィンドウを閉じて、クライアント マシンを再起動します。  
  
高速ログインの最適化を有効にしている場合は、クライアント マシンを 2 回再起動します。
- 9 vRealize Log Insight Windows エージェントがクライアント マシンにローカル サービスとしてインストールされていることを確認します。  
  
.mst ファイルを使用して vRealize Log Insight Windows エージェントの複数のインスタンスを展開するように SERVICEACCOUNT および SERVICEPASSWORD パラメータを構成している場合は、指定したユーザー アカウントでクライアント マシンに vRealize Log Insight Windows エージェントがインストールされていることを確認します。

#### 次のステップ

vRealize Log Insight Windows エージェントの複数のインスタンスが失敗する場合は、[Log Insight Windows Agent の一括展開に失敗しました](#)を参照してください。

## vRealize Log InsightLinux Agent RPM パッケージのインストールまたは更新

root または root 以外のユーザーとして vRealize Log InsightLinux エージェントをインストールまたは更新し、インストール中に設定パラメータを設定できます。インストール後に、インストールしたバージョンを確認できます。

## 前提条件

- インストールのデフォルトとその変更方法については、[Linux での vRealize Log Insight エージェントのインストールのコマンド ライン オプション](#) を参照してください。
- root としてログインするか、または sudo を使用してコンソール コマンドを実行します。
- vRealize Log InsightLinux エージェントが機能するためには、syslog およびネットワーク サービスにアクセスする必要があります。vRealize Log InsightLinux エージェントは、実行レベル 3 または 5 でインストールおよび実行してください。他の実行レベルで vRealize Log InsightLinux エージェントを実行する場合は、それに応じて適切にシステムを構成します。

## 手順

- 1 コンソールからエージェントをインストールまたはアップグレードできます。

- デフォルトの設定で vRealize Log InsightLinux エージェントをインストールするには、コンソールを開いて次のコマンドを実行します。

```
rpm -i VMware-Log-Insight-Agent-<version-and-build-number>.rpm
```

- 現在の設定を変更せずにエージェントをアップグレードするには、コンソールを開いて次のコマンドを実行します。

```
rpm -Uhv VMware-Log-Insight-Agent-<version-and-build-number>.rpm
```

- 2 (オプション) 更新中に、インストールのデフォルトの設定値または現在の設定値をオーバーライドできます。この操作を行うには、インストールまたはアップグレード コマンドにオプションを指定します。

```
sudo <OPTION=value> rpm -i <version-and-build-number>.rpm
```

- 3 (オプション) 次のコマンドを実行して、インストールされたバージョンを確認します。

```
rpm -qa | grep Log-Insight-Agent
```

## 例：Linux エージェントのインストールと更新の例

- 次のコマンドは、Linux RPM ディストリビューション用の vRealize Log Insight エージェントをインストールします。別のサーバにエージェントをインストールし、デフォルト以外のポート番号を割り当て、vRealize Log Insight エージェントのユーザーを作成します。

```
sudo SERVERHOST=myagentserver SERVERPORT=1234 LIAGENTUSER=liagent rpm -i VMware-Log-Insight-Agent-44.1234.rpm
```

- 次のコマンドは、指定された rpm ファイルでエージェントを更新します。エージェントの現在の設定は変更されません。

```
rpm -Uhv VMware-Log-Insight-Agent-44.1234.rpm
```

## vRealize Log Insight Linux Agent DEB パッケージのインストールまたは更新

vRealize Log Insight Linux Agent DEB (Debian) パッケージは、コマンドラインまたは debconf データベースからインストールまたは更新できます。インストール後に、インストールしたバージョンを確認できます。

### 前提条件

- [Linux での vRealize Log Insight エージェントのインストールのコマンドラインオプション](#) でインストールのデフォルト値とその変更方法を確認して、変更します。
- root としてログインするか、または sudo を使用してコンソール コマンドを実行します。
- vRealize Log InsightLinux エージェントが Syslog サービスとネットワーク サービスにアクセスできて機能することを確認します。デフォルトでは、vRealize Log InsightLinux エージェントは実行レベル 2、3、4、5 で実行し、実行レベル 0、1、6 で停止します。
- 詳細および例については、[Debian Linux 用のエージェントのインストールのカスタマイズ](#)を参照してください。

### 手順

- 1 vRealize Log Insight Linux エージェントをインストールまたは更新するには、コンソールを開き、dpkg -i *package\_name* コマンドを実行します。

*package\_name* は、プリフィックス **vmware-log-insight-agent-** とダウンロード版のバージョンのビルド番号で構成されます。

次のコマンドの形式は、デフォルト値を使用してパッケージをインストールします。

```
dpkg -i vmware-log-insight-agent-VERSION-BUILD_NUMBER_all.deb
```

- 2 (オプション) 次のコマンドを実行して、インストールされたバージョンを確認します。

```
dpkg -l | grep -i vmware-log-insight-agent
```

### 例

- コマンドラインから接続プロトコルを設定します。

デフォルトの接続プロトコルをオーバーライドするには、次の例に示すように SERVERPROTO 変数を使用します。

```
sudo SERVERPROTO=syslog dpkg -i vmware-log-insight-agent-<version-and-build-number>_all.deb
```

## Debian Linux 用のエージェントのインストールのカスタマイズ

インストールの現在の構成値をオーバーライドするコマンド オプションを使用するか、debconf データベースを構成することで、インストールをカスタマイズできます。

## コマンドラインからのカスタマイズ

コマンドラインからインストールを構成するには、次の形式のコマンドを使用します。

```
sudo <オプション=値> dpkg -i vmware-log-insight-agent-<version-and-build-number>_all.deb
```

オプションの全リストは、[Linux での vRealize Log Insight エージェントのインストールのコマンドライン オプション](#) を参照してください。

次の例は、コマンドラインから実行される典型的な構成を示します。

- ターゲットの vRealize Log Insight サーバを指定します。
- インストール中にターゲットを設定するには、sudo コマンド (hostname は vRealize Log Insight サーバの IP アドレスまたはホスト名に置き換え) を実行します。次に例を示します。

```
sudo SERVERHOST=hostname dpkg -iv mware-log-insight-agent-<version-and-build-number>_all.deb
```

インストール中に --force-confold フラグを有効にしないと、新しいバージョンに更新するときに必ず、システムから liagent.ini 構成ファイルを保持または置き換えるように指示されます。次のシステム メッセージが表示されます。

```
Configuration file `/var/lib/loginsight-agent/liagent.ini'
==> Modified (by you or by a script) since installation.
==> Package distributor has shipped an updated version.
What would you like to do about it ? Your options are:
  Y or I : install the package maintainer's version
  N or O : keep your currently-installed version
  D      : show the differences between the versions
  Z      : start a shell to examine the situation
The default action is to keep your current version.
*** liagent.ini (Y/I/N/O/D/Z) [default=N] ?
```

既存の構成を保持するには、[default=N] を使用します。コマンドラインから受け取った追加のパラメータが引き続き適用されます。

- 接続プロトコルを構成します。

デフォルトの接続プロトコルをオーバーライドするには、次の例に示すように SERVERPROTO 変数を使用します。

```
sudo SERVERPROTO=syslog dpkg -i vmware-log-insight-agent-<version-and-build-number>_all.deb
```

- 接続ポートを構成します。

デフォルトの接続ポートをオーバーライドするには、次の例に示すように、インストーラに SERVERPORT 変数の値を指定します。

```
sudo SERVERPORT=1234 dpkg -i vmware-log-insight-agent-<version-and-build-number>_all.deb
```

- 非 root ユーザーとしてエージェントを実行します。

vRealize Log Insight Linux エージェントを non root ユーザーとして実行するには、`sudo` コマンドを実行します。

```
sudo LIAGENTUSER=liagent dpkg -i vmware-log-insight-agent-<version-build-number>_all.deb
```

指定したユーザーが存在しない場合、vRealize Log InsightLinux エージェントはインストール中にそのユーザー アカウントを作成します。作成されたアカウントは、アンインストール後も削除されません。Linux エージェントを `LIAGENTUSER=non_root_user` パラメータを使用してインストールし、`LIAGENTUSER=non_root_user2` パラメータを使用してアップグレードを試みた場合、不一致が生じます。`non_root_user2` ユーザーに `non_root_user` ユーザーの権限がないために警告が表示されます。

## debconf データベース用の DEB パッケージのカスタマイズ オプション

エージェント DEB パッケージは、debconf データベースを使用して設定することもできます。次の表に、サポートされている debconf オプションとそれに対応する vRealize Log Insight エージェント DEB インストーラ オプションを示します。

コマンドライン オプション	debconf オプション	説明
<code>SERVERHOST=hostname</code>	<code>vmware-log-insight-agent/serverhost</code>	vRealize Log Insight 仮想アプライアンスの IP アドレスまたはホスト名。 デフォルトは <b>loginsight</b> です。
<code>SERVERPROTO={cfapi syslog}</code>	<code>vmware-log-insight-agent/serverproto</code>	エージェントが、イベントを vRealize Log Insight サーバに送信するために使用するプロトコル。設定可能な値は <code>cfapi</code> および <code>syslog</code> です。 デフォルトは <code>cfapi</code> です。
<code>SERVERPORT=portnumber</code>	<code>vmware-log-insight-agent/serverport</code>	エージェントが、イベントを vRealize Log Insight またはサードパーティのサーバに送信するために使用する通信ポート。デフォルトでは、エージェントは SSL とプロトコルに設定されたオプションに基づいて適切なポートを使用します。以下のリストに指定されたデフォルトのポート値を参照してください。ポート オプションは、これらのデフォルトと異なる場合にのみ指定する必要があります。 <ul style="list-style-type: none"> <li>■ SSL が有効になっている <code>cfapi</code>: 9543</li> <li>■ SSL が無効になっている <code>cfapi</code>: 9000</li> <li>■ SSL が有効になっている <code>syslog</code>: 6514</li> <li>■ SSL が無効になっている <code>syslog</code>: 514</li> </ul>
<code>LIAGENT_INITSYSTEM={init systemd}</code>	<code>log-insight-agent/init_system</code>	インストール時に、エージェントがインストール先の init システムの種類を自動的に検出します。この動作をオーバーライドするには、このオプションでシステムの種類を指定します。サポートされている init システムは、 <code>init</code> と <code>systemd</code> の 2 種類です。
<code>LIAGENT_AUTOUPDATE={yes no}</code>	<code>vmware-log-insight-agent/auto_update</code>	エージェントの自動更新を有効または無効にします。自動更新を完全に有効にするには、vRealize Log Insight サーバから自動更新を有効にする必要があります。デフォルトは <code>yes</code> です。自動更新は Linux の <code>bin</code> パッケージではサポートされません。

コマンドライン オプション	debconf オプション	説明
LI_AGENT_RUNSERVICES	vmware-log-insight-agent/init_system	デフォルトでは、インストール後すぐに liagentd (エージェント) と liupdaterd (updater) サービスが開始します。開始されないようにするには、LIAGENT_RUNSERVICES debconf パラメータを <b>no</b> に設定します。デフォルトは <b>yes</b> です。許容できる値は <b>yes</b> と <b>no</b> のみです。1 または [0] はサポートされません。
LIAGENT_SSL	vmware-log-insight-agent/ssl	C
LIAGENTUSER= <i>user-account-name</i>	vmware-log-insight-agent/liagentuser	<p>エージェントを実行するアカウントを指定します。ユーザーが存在しない場合、インストーラが一般ユーザーとしてユーザーを作成します。指定したユーザー アカウントが存在しない場合、vRealize Log InsightLinux エージェントはインストール中にそのユーザー アカウントを作成します。作成されたアカウントは、アンインストール後も削除されません。</p> <p>デフォルトでは、エージェントがインストールされ、root ユーザーとして実行されます。</p> <p>エージェントを LIAGENTUSER=<i>non_root_user</i> パラメータを使用してインストールし、LIAGENTUSER=<i>non_root_user2</i> を使用してアップグレードを試みた場合、不一致が生じます。<i>non_root_user2</i> ユーザがユーザ <i>non_root_user</i> の権限を持っていないために警告が表示されます。</p> <p>作成したユーザーはアンインストール時に削除されません。手動で削除できます。このパラメータはエージェント サービス専用です。updater サービスは常に root ユーザーとして実行されます。</p>

## Log Insight Linux Agent バイナリ パッケージのインストール

バイナリ パッケージのインストールでは、.bin ファイルを実行可能ファイルに変更し、エージェントをインストールします。

.bin パッケージのアップグレードは、正式にはサポートされていません。既存の Log Insight Linux Agent をインストールするのに .bin パッケージを使用していた場合には、/var/lib/loginsight-agent ディレクトリにある liagent.ini ファイルのバックアップ コピーを作成してローカルの構成を保持します。バックアップ コピーをとったら、手動で Log Insight Linux Agent をアンインストールします。[Log Insight Linux Agent bin パッケージの手動アンインストール](#) を参照してください。

Linux エージェントのインストールに .bin パッケージを使用する場合は、liagentd という名前の init.d スクリプトがパッケージ インストールの一部としてインストールされますが、パッケージはスクリプトを登録しません。スクリプトは手動で登録できます。

(/sbin/)service liagentd status コマンドを使用して、インストールが正常に実行されたことを確認できます。

### 前提条件

- Log Insight Linux Agent .bin パッケージをダウンロードしてターゲット Linux マシンにコピーします。

- Log Insight Linux Agent が Syslog サービスとネットワーク サービスにアクセスできることを確認します。
- デフォルトの構成値と、インストール時にその値を変更する方法についての説明をお読みください。[Linux での vRealize Log Insight エージェントのインストールのコマンド ライン オプション](#) を参照してください。

## 手順

- 1 コンソールを開き、chmod コマンドを実行して .bin ファイルを実行可能ファイルに変更します。

*filename-version* は該当するバージョンに置き換えます。

```
chmod +x filename-version.bin
```

- 2 コマンド プロンプトで ./filename-version.bin コマンドを実行してエージェントをインストールします。

*filename-version* は該当するバージョンに置き換えます。

```
./filename-version.bin
```

- 3 (オプション) ターゲットの vRealize Log Insight サーバをインストール中に設定するには、sudo SERVERHOST=hostname ./filename-version.bin コマンドを実行します。

*hostname* は vRealize Log Insight サーバの IP アドレスまたはホスト名に置き換えます。

```
sudo SERVERHOST=hostname ./filename-version.bin
```

- 4 (オプション) デフォルトの接続プロトコルをオーバーライドするには、次の例に示すように SERVERPROTO 変数を使用します。

```
sudo SERVERPROTO=syslog ./filename-version.htm
```

- 5 (オプション) デフォルトの接続ポートをオーバーライドするには、次の例に示すように、インストーラーに SERVERPORT 変数の値を指定します。

```
sudo SERVERPORT=1234 ./filename-version.htm
```

- 6 (オプション) Log Insight Linux Agent を non root ユーザーとして実行するには、sudo コマンドを実行します。

```
sudo LIAGENTUSER=liagent ./filename-version.bin
```

指定したユーザーが存在しない場合、Log Insight Linux Agent はインストール中にそのユーザー アカウントを作成します。作成されたアカウントは、アンインストール後も削除されません。Log Insight Linux Agent を LIAGENTUSER=non\_root\_user パラメータを使用してインストールし、LIAGENTUSER=non\_root\_user2 パラメータを使用してアップグレードを試みた場合、不一致が生じ、non\_root\_user2 ユーザーに non\_root\_user ユーザーの権限がないために警告が表示されます。

# Linux での vRealize Log Insight エージェントのインストールのコマンドラインオプション

vRealize Log Insight エージェントをコマンドラインからインストールする場合、インストール中に展開を構成するオプションを含めることができます。これらのオプションは、`liagent.ini` ファイルの設定に対応しています。

インストール時に次のオプションを使用して、Linux システムで動作する vRealize Log Insight エージェントを構成できます。

オプション	説明
<code>SERVERHOST=<i>hostname</i></code>	vRealize Log Insight 仮想アプライアンスの IP アドレスまたはホスト名。 デフォルトは <b>loginsight</b> です。
<code>SERVERPROTO={<b>cfapi</b> <b>syslog</b>}</code>	エージェントが、イベントを vRealize Log Insight サーバに送信するために使用するプロトコル。設定可能な値は <code>cfapi</code> および <code>syslog</code> です。 デフォルトは <code>cfapi</code> です。
<code>SERVERPORT=<i>portnumber</i></code>	エージェントが、イベントを vRealize Log Insight またはサードパーティのサーバに送信するために使用する通信ポート。デフォルトでは、エージェントは SSL とプロトコルに設定されたオプションに基づいて適切なポートを使用します。以下のリストに指定されたデフォルトのポート値を参照してください。ポート オプションは、これらのデフォルトと異なる場合にのみ指定する必要があります。 <ul style="list-style-type: none"> <li>■ SSL が有効になっている <code>cfapi</code>: 9543</li> <li>■ SSL が無効になっている <code>cfapi</code>: 9000</li> <li>■ SSL が有効になっている <code>syslog</code>: 6514</li> <li>■ SSL が無効になっている <code>syslog</code>: 514</li> </ul>
<code>LIAGENT_INITSYSTEM={<b>init</b> <b>systemd</b>}</code>	インストール時に、エージェントがインストール先の <code>init</code> システムの種類を自動的に検出します。この動作をオーバーライドするには、このオプションでシステムの種類を指定します。サポートされている <code>init</code> システムは、 <code>init</code> と <code>systemd</code> の 2 種類です。
<code>LIAGENT_AUTOUPDATE={<b>yes</b> <b>no</b>}</code>	エージェントの自動更新を有効または無効にします。自動更新を完全に有効にするには、vRealize Log Insight サーバから自動更新を有効にする必要があります。デフォルトは <code>yes</code> です。 自動更新は Linux の <code>bin</code> パッケージではサポートされません。



オプション	説明
<code>LIAGENT_SSL={yes no}</code>	安全な接続を有効にします。SSL が有効になっている場合、エージェントは TLS 1.2 プロトコルを使用してサーバと通信します。デフォルトは yes です。
<code>LIAGENTUSER=user-account-name</code>	<p>エージェントを実行するアカウントを指定します。ユーザーが存在しない場合、インストーラが一般ユーザーとしてユーザーを作成します。指定したユーザー アカウントが存在しない場合、vRealize Log InsightLinux エージェントはインストール中にそのユーザー アカウントを作成します。作成されたアカウントは、アンインストール後も削除されません。</p> <p>デフォルトでは、エージェントがインストールされ、root ユーザーとして実行されます。</p> <p><code>LIAGENTUSER=non_root_user</code> パラメータを使用してインストールし、<code>LIAGENTUSER=non_root_user2</code> を使用してアップグレードを試みた場合、不一致が生じます。<code>non_root_user2</code> ユーザーがユーザー <code>non_root_user</code> の権限を持っていないために警告が表示されます。作成したユーザーはアンインストール時に削除されません。手動で削除できます。このパラメータはエージェント サービス専用です。updater サービスは常に root ユーザーとして実行されます。</p>

## vRealize Log Insight エージェントの自動更新

vRealize Log Insight エージェントの自動更新機能では、アクティブ状態のエージェントが、エージェント インストール パッケージに基づいて、vRealize Log Insight サーバから更新を自動的にチェック、ダウンロード、適用します。

自動更新は、すべてのエージェントのサーバから、または個々のエージェント インスタンスのクライアントから有効にすることができます。エージェントはアクティブ状態で、バージョン 4.3 以降である必要があります。

自動更新は Linux の bin パッケージではサポートされません。

## 個々のエージェントでの自動更新の無効化と有効化

クライアント側の設定ファイルを編集することで、個々のエージェントの自動更新を有効にしたり無効にしたりすることができます。

デフォルトでは、自動更新はエージェントのクライアント側で有効になっています。

### 前提条件

エージェントのバージョンは、4.3 以降でなければなりません。

### 手順

- 1 エディタでローカルの `liagent.ini` ファイルを開きます。

## 2 [update] セクションに移動します。

次の例のようになっています。

```
[update]
; Do not change this parameter
package_type=msi
; Enable automatic update of the agent. If enabled:
; the agent will silently check for updates from the server and
; if available will automatically download and apply the update.
; auto_update=yes
```

## 3 自動更新を無効にするには、auto\_update=yes をコメント解除し、auto\_update=no に変更します。

---

**注：** エージェントの自動更新はデフォルトで有効になっています。このため、auto\_update のデフォルト値はコメントされている場合でも「yes」です。

---

## 4 liagent.ini ファイルを保存して閉じます。

# vRealize Log Insight エージェントの構成

# 4

エージェントの展開後、選択した vRealize Log Insight サーバにイベントを送信したり、通信プロトコルを指定したり、その他のパラメータを設定したりするようにエージェントを構成できます。

必要に応じてこれらの手順を使用し、エージェントを要件に合わせて構成してください。

## ■ Log Insight Windows Agent の構成

インストールした Log Insight Windows Agent を構成できます。liagent.ini ファイルを編集して、vRealize Log Insight にイベントを送信するように Log Insight Windows Agent を構成し、通信プロトコルおよびポートを設定し、Windows イベント チャンネルを追加し、フラット ファイル ログの収集を構成します。ファイルは %ProgramData%\VMware\Log Insight Agent ディレクトリにあります。

## ■ Log Insight Linux Agent の構成

インストールした Log Insight Linux Agent を構成できます。

## ■ vRealize Log Insight エージェントからのイベントのフィルタリング

ローカルの liagent.ini ファイルの [server|<dest\_id>] セクションにあるフィルタ オプションを使用して、エージェントが宛先に送信する情報を提供できます。

## ■ vRealize Log Insight エージェントからの情報の転送

エージェントによって収集されたイベントは最大 3 つの宛先に転送できます。宛先には、vRealize Log Insight サーバ、フォワーダ、またはサードパーティのログ管理ソリューションを含めることができます。

## ■ vRealize Log Insight エージェントの構成の一元化

複数の vRealize Log Insight エージェントを構成することができます。

## ■ エージェント構成に対する共通値の使用

Windows および Linux エージェントで、エージェント構成ファイルのデフォルト値を各エージェント構成セクションに適用される共通パラメータ値でオーバーライドできます。

## ■ ログの解析

エージェント側のパーサは、vRealize Log Insight サーバへの配信前に加工前のログから構造化データを抽出します。vRealize Log Insight は、ログパーサを使用して分析したログから情報を抽出し、サーバにその分析結果を表示することができます。ログパーサは Windows および Linux vRealize Log Insight Agent のいずれに対しても構成することができます。

## Log Insight Windows Agent の構成

インストールした Log Insight Windows Agent を構成できます。liagent.ini ファイルを編集して、vRealize Log Insight にイベントを送信するように Log Insight Windows Agent を構成し、通信プロトコルおよびポートを設定し、Windows イベント チャンネルを追加し、フラット ファイル ログの収集を構成します。ファイルは %ProgramData%\VMware\Log Insight Agent ディレクトリにあります。

## Log Insight Windows Agent のデフォルト構成

インストール後、liagent.ini ファイルには、Log Insight Windows Agent 用の事前構成されたデフォルト設定が格納されます。

### Log Insight Windows Agent の liagent.ini のデフォルト構成

名前および値に ASCII 以外の文字を使用する場合は、構成を UTF-8 で保存します。

一元的な構成を使用している場合は、最終構成として、このファイルがサーバの設定と組み合わせられて liagent-effective.ini ファイルが生成されます。

この設定は、サーバのエージェント ページから構成するとより効率的な場合があります。

```
; Client-side configuration of VMware Log Insight Agent.
; See liagent-effective.ini for the actual configuration used by VMware Log Insight Agent.

[server]
; Log Insight server hostname or ip address
; If omitted the default value is LOGINSIGHT
;hostname=LOGINSIGHT

;Enables or disables centralized configuration from the vRealize Log Insight server.
;When enabled, agent configuration changes made to the liagent.ini file on the server
;are joined with the settings in this file. to this agent. Accepted values are yes or no and
0 or 1.
;The default is yes.
;
;central_config=yes
;

; Set protocol to use:
; cfapi - Log Insight REST API
; syslog - Syslog protocol
; If omitted the default value is cfapi
;
;proto=cfapi

; Log Insight server port to connect to. If omitted the default value is:
; for syslog: 514
; for cfapi without ssl: 9000
; for cfapi with ssl: 9543
;port=9000

;ssl - enable/disable SSL. Applies to cfapi protocol only.
; Possible values are yes or no. If omitted the default value is no.
```

```

;ssl=no

; Time in minutes to force reconnection to the server
; If omitted the default value is 30
;reconnect=30

[storage]
;max_disk_buffer - max disk usage limit (data + logs) in MB:
; 100 - 2000 MB, default 200
;max_disk_buffer=200

[logging]
;debug_level - the level of debug messages to enable:
; 0 - no debug messages
; 1 - trace essential debug messages
; 2 - verbose debug messages (will have negative impact on performance)
;debug_level=0
;
;The interval in minutes to print statistics
;stats_period=15

[update]
; Do not change this parameter
package_type=msi

; Enable automatic update of the agent. If enabled:
; the agent will silently check for updates from the server and
; if available will automatically download and apply the update.
;auto_update=yes

[winlog|Application]
channel=Application
raw_syslog=no

[winlog|Security]
channel=Security

[winlog|System]
channel=System

```

パラメータ	デフォルト値	説明
hostname	LOGINSIGHT	vRealize Log Insight 仮想アプライアンスの IP アドレスまたはホスト名。 デフォルトは <b>loginsight</b> です。
central_config	yes	このエージェントについて、一元化された構成を有効または無効にします。一元化された構成が無効になっている場合、エージェントは vRealize Log Insight サーバが提供する構成を無視します。許容値は yes、no、1、または 0 です。デフォルト値は yes です。

パラメータ	デフォルト値	説明
proto	cfapi	<p>エージェントが、イベントを vRealize Log Insight サーバに送信するために使用するプロトコル。設定可能な値は cfapi および syslog です。</p> <p>デフォルトは cfapi です。</p>
port	9543、9000、6514、および 514	<p>エージェントが、イベントを vRealize Log Insight またはサードパーティのサーバに送信するために使用する通信ポート。デフォルトでは、エージェントは SSL とプロトコルに設定されたオプションに基づいて適切なポートを使用します。以下のリストに指定されたデフォルトのポート値を参照してください。ポート オプションは、これらのデフォルトと異なる場合にのみ指定する必要があります。</p> <ul style="list-style-type: none"> <li>■ SSL が有効になっている cfapi: 9543</li> <li>■ SSL が無効になっている cfapi: 9000</li> <li>■ SSL が有効になっている syslog: 6514</li> <li>■ SSL が無効になっている syslog: 514</li> </ul>
ssl	yes	<p>SSL を有効または無効にします。デフォルト値は yes です。</p> <p>ssl を yes に設定すると、ポートの値を設定していない場合、自動的に 9543 ポートが割り当てられます。</p>
max_disk_buffer	200	<p>Log Insight Windows Agent がイベントとそのログをバッファに格納するために使用する MB 単位での最大ディスク容量です。</p> <p>指定された max_disk_buffer に達すると、エージェントは新たに受信したイベントのドロップを開始します。</p>
debug_level	0	<p>ログの詳細レベルを定義します。 <a href="#">Log Insight Agents のログの詳細レベルの定義</a>を参照してください。</p>

パラメータ	デフォルト値	説明
channel	Application、Security、System	アプリケーション、セキュリティ、およびシステムの Windows イベント ログのチャンネルは、デフォルトではコメント設定されています。Log Insight Windows Agent は、これらのチャンネルからログを収集しません。 <a href="#">Windows イベント チャンネルからのイベントの収集</a> を参照してください。
raw_syslog	no	Syslog プロトコルを使用するエージェントの場合、未加工の Syslog イベントの収集と送信をエージェントに許可します。デフォルトは no で、収集されたイベントはユーザーが指定した Syslog 属性で変換されます。Syslog 変換を行わずにイベントを収集するには、このオプションを有効にします。  許容値は、yes または no、あるいは 0 または 1 です。

## Windows イベント チャンネルからのイベントの収集

Windows イベント チャンネルを Log Insight Windows Agent 構成に追加できます。Log Insight Windows Agent はイベントを収集し、vRealize Log Insight サーバに送信します。

フィールド名には制限があります。次の名前は予約されているため、フィールド名としては使用できません。

- event\_type
- hostname
- source
- text

### 前提条件

vRealize Log InsightWindows エージェントをインストールした Windows マシンにログインし、サービス マネージャを起動して vRealize Log Insight エージェント サービスがインストールされていることを確認します。

### 手順

- 1 vRealize Log InsightWindows エージェントのプログラム データ ディレクトリに移動します。  
%ProgramData%\VMware\Log Insight Agent
- 2 任意のテキスト エディタで liagent.ini を開きます。

### 3 次のパラメータを追加し、使用環境の値を設定します。

パラメータ	説明
<code>[winlog section_name]</code>	構成セクションの一意な名前。
<code>channel</code>	イベント ビューア組み込み Windows アプリケーションに表示されるイベント チャネルのフル ネーム。正しいチャネル名をコピーするには、イベント ビューアでチャネルを右クリックし、[プロパティ] を選択して [フル ネーム] フィールドの内容をコピーします。
<code>enabled</code>	構成セクションを有効または無効にするオプションのパラメータ。有効値は yes または no です（大文字と小文字は区別されません）。デフォルト値は yes です。
<code>tags</code>	収集したイベントのフィールドにカスタム タグを追加するためのオプションのパラメータ。JSON 表記を使用してタグを定義します。タグ名には文字、数字、アンダースコアを含めることができます。タグ名の先頭は文字またはアンダースコアにする必要があります、タグ名の長さは 64 文字以内にする必要があります。タグ名は、大文字と小文字が区別されません。たとえば、tags={"tag_name1" : "tag value 1", "Tag_Name1" : "tag value 2" } を使用する場合、Tag_Name1 は重複として無視されます。event_type および timestamp をタグ名として使用することはできません。同一の宣言内での重複は無視されます。 ターゲットが Syslog サーバの場合、タグで APP-NAME フィールドをオーバーライドできます。例：tags={"appname":"VROPS"}
<code>whitelist, blacklist</code>	ログ イベントを明示的に含める、または除外するオプションのパラメータ。 <b>注：</b> blacklist オプションはフィールドに対してのみ機能し、テキストをブロックするためには使用できません。
<code>exclude_fields</code>	(オプション) 収集から個別のフィールドを除外するパラメータ。複数の値をセミコロンで区切って指定することができます。例：exclude_fields=EventId; ProviderName

```
[winlog|section_name]
channel=event_channel_name
enabled=yes_or_no
tags={"tag_name1" : "Tag value 1", "tag_name2" : "tag value 2" }
```

### 4 liagent.ini ファイルを保存して閉じます。

#### 例：構成

次の [winlog| 構成の例を参照してください。

```
[winlog|Events_Firewall ]
channel=Microsoft-Windows-Windows Firewall With Advanced Security/Firewall
enabled=no
```

```
[winlog|custom]
channel=Custom
tags={"ChannelDescription": "Events testing channel"}
```

## Windows イベント チャネルのフィルタの設定

Windows イベント チャネルのフィルタを設定して、ログ イベントを明示的に含めるか、除外します。



フィルタ式を評価するには、whitelist および blacklist パラメータを使用します。フィルタ式は、イベントフィールドおよび演算子で構成されるブール式です。

**注：** blacklist オプションはフィールドに対してのみ機能し、テキストをブロックするためには使用できません。

- whitelist パラメータは、フィルタ式が非ゼロと評価されたログ イベントのみを収集します。このパラメータを省略する場合、値は暗黙の値 1 です。
- blacklist パラメータは、フィルタ式が非ゼロと評価されたログ イベントを除外します。デフォルト値は 0 です。

Windows イベントおよび演算子の完全なリストについては、[イベントのフィールドおよび演算子](#) を参照してください。

#### 前提条件

vRealize Log Insight Windows エージェントをインストールした Windows マシンにログインし、サービス マネージャを起動して vRealize Log Insight エージェント サービスがインストールされていることを確認します。

#### 手順

- 1 vRealize Log Insight Windows エージェントのプログラム データ ディレクトリに移動します。

```
%ProgramData%\VMware\Log Insight Agent
```

- 2 任意のテキスト エディタで liagent.ini を開きます。

- 3 [winlog|] セクションに whitelist または blacklist パラメータを追加します。

例：

```
[winlog|unique_section_name]
channel = event_channel_name
blacklist = filter_expression
```

- 4 Windows イベント フィールドおよび演算子を使用してフィルタ式を作成します。

例：

```
whitelist = level > WINLOG_LEVEL_SUCCESS and level < WINLOG_LEVEL_INFO
```

- 5 liagent.ini ファイルを保存して閉じます。

#### 例：フィルタ構成

たとえば次のように、エラー イベントのみを収集するようにエージェントを構成できます。

```
[winlog|Security-Error]
channel = Security
whitelist = Level == WINLOG_LEVEL_CRITICAL or Level == WINLOG_LEVEL_ERROR
```

たとえば次のように、アプリケーション チャネルから VMware ネットワーク イベントのみを収集するようにエージェントを構成できます。

```
[winlog|VMwareNetwork]
channel = Application
whitelist = ProviderName == "VMnetAdapter" or ProviderName == "VMnetBridge" or ProviderName == "VMnetDHCP"
```

たとえば次のように、セキュリティ チャネルから特定のイベント以外のすべてのイベントを収集するようにエージェントを構成できます。

```
[winlog|Security-Verbose]
channel = Security
blacklist = EventID == 4688 or EventID == 5447
```

## イベントのフィールドおよび演算子

Windows イベントのフィールドおよび演算子を使用して、フィルタ式を作成します。

### フィルタ式の演算子

演算子	説明
==, !=	等しいおよび等しくない。数値フィールドと文字列フィールドのいずれとも使用できます。
>=, >, <, <=	以上、より大きい、未満、以下。数値フィールドのみと使用します。
&,  , ^, ~	ビット AND、OR、XOR および補数演算子。数値フィールドのみと使用します。
and, or	論理 AND および OR。単純な式を組み合わせることで複雑な式を作成するために使用します。
not	論理 NOT 単項演算子。式の値を逆にするために使用します。
()	評価の順序を変更するには、論理式で括弧を使用します。

### Windows イベント フィールド

フィルタ式では次の Windows イベント フィールドを使用できます。

フィールド名	フィールド タイプ
Hostname	文字列
Text	文字列
ProviderName	文字列
EventSourceName	文字列
EventID	数値
EventRecordID	数値
Channel	文字列
UserID	文字列

フィールド名	フィールド タイプ
Level	数値 次の定義済み定数を使用できます。 <ul style="list-style-type: none"> <li>■ WINLOG_LEVEL_SUCCESS = 0</li> <li>■ WINLOG_LEVEL_CRITICAL = 1</li> <li>■ WINLOG_LEVEL_ERROR = 2</li> <li>■ WINLOG_LEVEL_WARNING = 3</li> <li>■ WINLOG_LEVEL_INFO = 4</li> <li>■ WINLOG_LEVEL_VERBOSE = 5</li> </ul>
タスク	数値
OpCode	数値
Keywords	数値 次の事前に定義されたビット マスクを使用できます。 <ul style="list-style-type: none"> <li>■ WINLOG_KEYWORD_RESPONSETIME = 0x0001000000000000;</li> <li>■ WINLOG_KEYWORD_WDICONTEXT = 0x0002000000000000;</li> <li>■ WINLOG_KEYWORD_WDIDIAGNOSTIC = 0x0004000000000000;</li> <li>■ WINLOG_KEYWORD_SQM = 0x0008000000000000;</li> <li>■ WINLOG_KEYWORD_AUDITFAILURE = 0x0010000000000000;</li> <li>■ WINLOG_KEYWORD_AUDITSUCCESS = 0x0020000000000000;</li> <li>■ WINLOG_KEYWORD_CORRELATIONHINT = 0x0040000000000000;</li> <li>■ WINLOG_KEYWORD_CLASSIC = 0x0080000000000000;</li> </ul>

## 例

すべての重大、エラー、警告のイベントを収集します。

```
[winlog|app]
channel = Application
whitelist = level > WINLOG_LEVEL_SUCCESS and level < WINLOG_LEVEL_INFO
```

セキュリティ チャネルから監査エラー イベントのみを収集します。

```
[winlog|security]
channel = Security
whitelist = Keywords & WINLOG_KEYWORD_AUDITFAILURE
```

## ログ ファイルからのイベントの収集

1 つまたは複数のログ ファイルからイベントを収集するように、vRealize Log Insight Windows エージェントを構成できます。

フィールド名には制限があります。次の名前は予約されているため、フィールド名としては使用できません。

- event\_type
- hostname
- source
- text

エージェント情報には最大 3 つの宛先を設定し、送信前に情報をフィルタできます。[vRealize Log Insight エージェントからの情報の転送](#)を参照してください。

---

**注：**

- 数千を超えるような多数のファイルを監視する場合、エージェントによるリソースの使用率が高くなり、ホストマシン全体のパフォーマンスに影響を与えます。これを防ぐには、パターンと glob を使用して必要なファイルのみを監視するようにエージェントを構成するか、古いログ ファイルをアーカイブします。多数のファイルを監視する必要がある場合は、CPU や RAM などのホスト パラメータの値を増やします。
  - エージェントがディレクトリを暗号化したユーザーによって実行されている場合のみ、エージェントは暗号化されたディレクトリから収集できます。
  - エージェントは静的ディレクトリ構造のみをサポートします。ディレクトリが名前変更または追加されている場合は、これらのディレクトリの監視を開始するためにエージェントを再起動する必要があります（構成の対象にディレクトリが含まれている場合）。
- 

**前提条件**

vRealize Log InsightWindows エージェントをインストールした Windows マシンにログインし、サービス マネージャを起動して vRealize Log Insight エージェント サービスがインストールされていることを確認します。

**手順**

- 1 vRealize Log InsightWindows エージェントのプログラム データ ディレクトリに移動します。  
`%ProgramData%\VMware\Log Insight Agent`
- 2 任意のテキスト エディタで `liagent.ini` を開きます。

- 3 ファイルで [server|<dest\_id>] セクションを探します。構成パラメータを追加し、使用環境の値を設定します。

```
[filelog|section_name]
directory=path_to_log_directory
include=glob_pattern
...
```

パラメータ	説明
[filelog section_name]	構成セクションの一意な名前。
directory=full-path-to-log-file	<p>ログ ファイル ディレクトリへの完全パス。Glob パターンがサポートされます。構成例：</p> <ul style="list-style-type: none"> <li>■ D:\Logs\new_test_logs ディレクトリのすべてのサブディレクトリから収集するには、directory=D:\Logs\new_test_logs\* を使用します。</li> <li>■ サブディレクトリ自身にサブディレクトリがある場合は、次の構成を使用してすべてのサブディレクトリを監視します。directory=D:\Logs\new_test_logs\*\*</li> </ul> <p><b>注：</b> ファイルとディレクトリ数を制限し、リソースの消費を抑える場合、最初または 2 番目のレベルのディレクトリ（たとえば directory=c:/tmp/* または directory=c:\Logs\*）のディレクトリ glob を定義することはできません。ディレクトリパスは 2 つ以上のレベルにする必要があります。</p> <p>まだ存在しないディレクトリのパスを定義できます。実際にディレクトリとファイルが作成されると、エージェントはそのディレクトリ内のログ ファイルを収集します。</p> <p>1 つまたは複数の異なる構成セクションで同一のディレクトリを定義し、同一ファイルから繰り返しログを収集することができます。この処理により、同一イベントのソースに異なるタグやフィルタを適用することができます。</p> <p><b>注：</b> これらのセクションに同一の構成を使用すると、重複したイベントがサーバー側で確認されます。</p>
include=file_name; ...	<p>（オプション）データを収集するファイル名またはファイル マスク（glob パターン）。複数の値をセミコロンで区切って指定することができます。デフォルトの値は *で、これはすべてのファイルが含まれることを意味します。パラメータは大文字と小文字が区別されます。</p> <p>ファイル マスク（グロブ パターン）を使用して、同じ命名規則に従う、単一のファイル名で指定されるファイルをグループ化することができます。たとえば、vRealize Ops Analytics.log および vRealize Ops Collector.log のようにファイル名にスベースが含まれる場合は、vRealize?Ops?Analytics*.log または vRealize*.log のように指定できます。ファイル マスクを使用すると、Linux および Windows ホストのエージェント構成で、使用可能なファイル名を指定できます。</p> <p>デフォルトでは、.zip ファイルおよび .gz ファイルが収集から除外されています。</p> <p><b>重要：</b> ローテーションされるログ ファイルを収集している場合、include および exclude パラメータを使用してプライマリ ファイルとローテーションされるファイルの両方に一致する glob パターンを指定します。glob パターンがプライマリ ログ ファイルにのみ一致する場合、vRealize Log Insight エージェントはローテーション中にイベントを収集できない場合があります。vRealize Log Insight エージェントは自動的にローテーション ファイルの正しい順序を判断し、vRealize Log Insight サーバに正しい順序でイベントを送信します。たとえば、プライマリ ログ ファイルの名前が myapp.log で、ローテーション ログが myapp.log.1 および myapp.log.2 と続く場合、次の include パターンを使用できます。</p> <pre>include= myapp.log;myapp.log.*</pre>

パラメータ	説明
<b>exclude=regular_expression</b>	(オプション) 収集から除外するファイル名またはファイル マスク (glob パターン)。複数の値をセミコロンで区切って指定することができます。デフォルトの値は空で、これは除外するファイルがないことを意味します。
<b>event_marker=regular_expression</b>	<p>(オプション) ログ ファイル内のイベントの開始を示す正規表現。省略した場合、デフォルトは改行です。入力する表現には、Perl 正規表現構文を使用する必要があります。</p> <p><b>注：</b> 引用符 (" ") のような文字は、正規表現ではラッパーとみなされることはありません。これらはパターンの一部とみなされます。</p> <p>vRealize Log Insight エージェントはリアルタイム収集用に最適化されているため、内部遅延によって書き込まれた部分的なログ メッセージが複数のイベントに分割されることがあります。ログ ファイルの追加が 200 ミリ秒以上停止していて、その間に新しい event_marker が観察されなかった場合は、部分的なイベントが完了、解析済み、および配信済みとして処理されます。このタイミング ロジックは構成することができず、event_marker 設定よりも優先します。ログ ファイル アペンダによってすべてのイベントがフラッシュされます。</p>
<b>enabled=yes no</b>	(オプション) 構成セクションを有効または無効にするパラメータ。設定可能な値は yes または no です。デフォルト値は yes です。
<b>charset=char-encoding-type</b>	<p>(オプション) エージェントが監視するログ ファイルの文字エンコーディング。値は次のとおりです。</p> <ul style="list-style-type: none"> <li>■ UTF-8</li> <li>■ UTF-16LE</li> <li>■ UTF-16BE</li> </ul> <p>デフォルト値は UTF-8 です。</p>
<b>tags={"tag-name": "tag-value", ...}</b>	<p>(オプション) 収集したイベントのフィールドにカスタム タグを追加するためのパラメータ。JSON 表記を使用してタグを定義します。タグ名には文字、数字、アンダースコアを含めることができます。タグ名の先頭は文字またはアンダースコアにする必要があります、タグ名の長さは 64 文字以内にする必要があります。タグ名は、大文字と小文字が区別されません。たとえば、tags={"tag_name1": "tag value 1", "Tag_Name1": "tag value 2"} を使用する場合、Tag_Name1 は重複として無視されます。event_type および timestamp をタグ名として使用することはできません。同一の宣言内での重複は無視されます。</p> <p>ターゲットが Syslog サーバの場合、タグで APP-NAME フィールドをオーバーライドできます。例: tags={"appname": "VROPS"}</p>
<b>exclude_fields</b>	<p>(オプション) 収集から個別のフィールドを除外するパラメータ。複数の値をセミコロン区切りリストまたはコンマ区切りリストとして指定することができます。次に例を示します。</p> <ul style="list-style-type: none"> <li>■ exclude_fields=hostname; filepath</li> <li>■ exclude_fields=type; size</li> <li>■ exclude_fields=type, size</li> </ul>
<b>raw_syslog=Yes No</b>	Syslog プロトコルを使用するエージェントの場合、このオプションを使用すると、未加工の Syslog イベントの収集と送信をエージェントに許可します。デフォルトは No です。収集されたイベントは、指定した Syslog 属性を使用して変換されます。Syslog 変換を行わずにイベントを収集するには、このオプションを有効にします。

## 例：構成

```
[filelog|vCenterMain]
directory=C:\ProgramData\VMware\VMware VirtualCenter\Logs
include=vpxd-*.log
```

```
exclude=vpxd-alert-*.log;vpxd-profiler-*.log
event_marker=^\\d{4}-\\d{2}-\\d{2}[A-Z]\\d{2}:\\d{2}:\\d{2}\\.\\d{3}
```

```
[filelog|ApacheAccessLogs]
enabled=yes
directory=C:\\Program Files (x86)\\Apache Software Foundation\\Apache2.2\\logs
include=*.log
exclude=*_old.log
tags={"Provider" : "Apache"}
```

```
[filelog|MSSQL]
directory=C:\\Program Files\\Microsoft SQL Server\\MSSQL10.MSSQLSERVER\\MSSQL\\Log
charset=UTF-16LE
event_marker=^[^\\s]
```

## Windows ログ ファイル チャンネル フィルタリングの設定

Windows ログ ファイルのフィルタを設定して、ログ イベントを明示的に含めるか、除外します。

フィルタ式を評価するには、whitelist および blacklist パラメータを使用します。フィルタ式は、イベント フィールドおよび演算子で構成されるブール式です。

---

**注：** blacklist オプションはフィールドに対してのみ機能し、テキストをブロックするためには使用できません。

---

- whitelist パラメータは、フィルタ式が非ゼロと評価されたログ イベントのみを収集します。このパラメータを省略する場合、値は暗黙の値 1 です。
- blacklist パラメータは、フィルタ式が非ゼロと評価されたログ イベントを除外します。デフォルト値は 0 です。

Windows イベントおよび演算子の完全なリストについては、[イベントのフィールドおよび演算子](#) を参照してください。

### 前提条件

vRealize Log InsightWindows エージェントをインストールした Windows マシンにログインし、サービス マネージャを起動して vRealize Log Insight エージェント サービスがインストールされていることを確認します。

### 手順

- 1 vRealize Log InsightWindows エージェントのプログラム データ ディレクトリに移動します。  
`%ProgramData%\VMware\Log Insight Agent`
- 2 任意のテキスト エディタで liagent.ini を開きます。

- 3 [filelog|] セクションに whitelist または blacklist パラメータを追加します。

例：

```
[filelog|apache]
directory = path_to_log_directory
include = glob_pattern
blacklist = filter_expression
```

- 4 Windows イベント フィールドおよび演算子を使用してフィルタ式を作成します。

例：

```
whitelist = myServer
```

- 5 liagent.ini ファイルを保存して閉じます。

例： フィルタ構成

server\_name が以下の設定のような Apache ログのみを収集するようにエージェントを構成できます。

```
[filelog|apache]
directory=C:\Program Files\Apache Software Foundation\Apache2.4\logs
include=error.log
parser=clf
whitelist = server_name == "sample.com"
blacklist = remote_host == "127.0.0.1"
```

## Log Insight Windows Agent へのイベントの転送

Windows マシンから Log Insight Windows Agent が動作しているマシンにイベントを転送できます。

Windows イベントの転送を使用して、複数の Windows マシンから Log Insight Windows Agent がインストールされているマシンにイベントを転送できます。次に、すべての転送イベントを収集し、vRealize Log Insight サーバにそれらを送信するように Log Insight Windows Agent を構成できます。

Windows イベントの転送について理解してください。 <http://technet.microsoft.com/en-us/library/cc748890.aspx> および [http://msdn.microsoft.com/en-us/library/windows/desktop/bb870973\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/bb870973(v=vs.85).aspx) を参照してください。

### 前提条件

[Windows イベント チャンネルからのイベントの収集](#) を参照してください。

### 手順

- 1 Log Insight Windows Agent 構成に新しいセクションを追加して、転送イベントを受信する Windows イベント チャンネルからイベントを収集します。  
デフォルトのチャンネル名は ForwardedEvents です。
- 2 Windows イベントの転送を設定します。



## 次のステップ

vRealize Log Insight Web ユーザー インターフェイスに移動し、転送イベントが到着していることを確認します。

## Log Insight Linux Agent の構成

インストールした Log Insight Linux Agent を構成できます。

[一元化されたエージェントの構成](#)を使用して、vRealize Log Insight サーバにイベントを送信するようにエージェントを設定し、通信プロトコルおよびポートを指定し、フラット ファイル ログの収集を構成できます。

liagent.ini ファイルの場所については、[Log Insight Agents のログの詳細レベルの定義](#)を参照してください。

## vRealize Log InsightLinux Agent のデフォルト構成

インストール後、liagent.ini ファイルには、Log Insight Windows Agent 用の事前構成されたデフォルト設定が格納されます。

## vRealize Log InsightLinux Agent の liagent.ini のデフォルト構成

名前および値に ASCII 以外の文字を使用する場合は、構成を UTF-8 で保存します。

一元的な構成を使用している場合は、最終構成として、このファイルがサーバの設定と組み合わせられて liagent-effective.ini ファイルが生成されます。

この設定は、サーバのエージェント ページから構成するとより効率的な場合があります。

```
[server]
; Hostname or IP address of your Log Insight server / cluster load balancer. Default:
;hostname=LOGINSIGHT

;Enables or disables centralized configuration from the vRealize Log Insight server.
;When enabled, agent configuration changes made to the liagent.ini file on the server
;are joined with the settings in this file. to this agent. Accepted values are yes or no and
0 or 1.
;The default is yes.
;
;central_config=yes
;
;
; Protocol can be cfapi (Log Insight REST API), syslog. Default:
;proto=cfapi

; Log Insight server port to connect to. Default ports for protocols (all TCP):
; syslog: 514; syslog with ssl: 6514; cfapi: 9000; cfapi with ssl: 9543. Default:
;port=9543

; SSL usage. Default:
;ssl=yes
; Example of configuration with trusted CA:
;ssl=yes
```

```

;ssl_ca_path=/etc/pki/tls/certs/ca.pem

; Time in minutes to force reconnection to the server.
; This option mitigates imbalances caused by long-lived TCP connections. Default:
;reconnect=30

[logging]
; Logging verbosity: 0 (no debug messages), 1 (essentials), 2 (verbose with more impact on
performance).
; This option should always be 0 under normal operating conditions. Default:
;debug_level=0

[storage]
; Max local storage usage limit (data + logs) in MBs. Valid range: 100-2000 MB.
;max_disk_buffer=200

; Uncomment the appropriate section to collect system logs
; The recommended way is to enable the Linux content pack from LI server
;[filelog|syslog]
;directory=/var/log
;include=messages;messages.?.syslog;syslog.

```

パラメータ	デフォルト値	説明
hostname	LOGINSIGHT	vRealize Log Insight 仮想アプライアンスの IP アドレスまたはホスト名。 デフォルトは <b>loginsight</b> です。
central_config	yes	このエージェントについて、一元化された構成を有効または無効にします。一元化された構成が無効になっている場合、エージェントは vRealize Log Insight サーバが提供する構成を無視します。許容値は yes、no、1、または 0 です。デフォルト値は yes です。
proto	cfapi	エージェントが、イベントを vRealize Log Insight サーバに送信するために使用するプロトコル。設定可能な値は cfapi および syslog です。 デフォルトは cfapi です。
port	9543、9000、6514、および 514	エージェントが、イベントを vRealize Log Insight サーバに送信するために使用する通信ポート。デフォルトの値は、SSL が有効にされた cfapi の場合は 9543、SSL が無効にされた cfapi の場合は 9000、SSL が有効にされた Syslog の場合は 6514、SSL が無効にされた syslog の場合は 514 です。
ssl	yes	SSL を有効または無効にします。デフォルト値は yes です。 ssl を yes に設定すると、ポートの値を設定していない場合、自動的に 9543 ポートが割り当てられます。

パラメータ	デフォルト値	説明
max_disk_buffer	200	Log Insight Windows Agent がイベントとそのログをバッファに格納するために使用する MB 単位での最大ディスク容量です。  指定された max_disk_buffer に達すると、エージェントは新たに受信したイベントのドロップを開始します。
debug_level	0	ログの詳細レベルを定義します。 <a href="#">Log Insight Agents のログの詳細レベルの定義</a> を参照してください。

## ログ ファイルからのイベントの収集

1 つまたは複数のログ ファイルからイベントを収集するように、vRealize Log Insight Linux エージェントを構成できます。

デフォルトでは、vRealize Log Insight Linux エージェントはアプリケーションまたはエディタによって作成された隠しファイルを収集します。隠しファイルの名前はピリオドで始まります。vRealize Log Insight Linux エージェントが隠しファイルを収集しないようにするには、除外パラメータ **exclude=.\*** を追加することができます。

フィールド名には制限があります。次の名前は予約されているため、フィールド名としては使用できません。

- event\_type
- hostname
- source
- text

エージェント情報には最大 3 つの宛先を指定し、送信前に情報をフィルタできます。 [vRealize Log Insight エージェントからの情報の転送](#)を参照してください。

**注：** 数千を超えるような多数のファイルを監視する場合、vRealize Log Insight Agent によるリソースの使用率が高くなり、ホスト マシン全体のパフォーマンスに影響を与えます。これを防ぐには、パターンと glob を使用して必要なファイルのみを監視するようにエージェントを構成するか、古いログ ファイルをアーカイブします。多数のファイルを監視する必要がある場合は、CPU や RAM などのホスト パラメータの値を増やします。

### 前提条件

- root としてログインするか、または sudo を使用してコンソール コマンドを実行します。
- vRealize Log Insight Linux エージェントがインストールされ、実行されていることを確認します。vRealize Log Insight Linux エージェントをインストールした Linux マシンにログインし、コンソールを開き、pgrep liagent を実行します。

### 手順

- 1 任意のテキスト エディタで /var/lib/loginsight-agent/liagent.ini ファイルを開きます。

- 2 ファイルで [server|<dest\_id>] セクションを探します。構成パラメータを追加し、使用環境の値を設定します。

```
[filelog|section_name]
directory=path_to_log_directory
include=glob_pattern
...
```

パラメータ	説明
[filelog section_name]	構成セクションの一意な名前。
directory=full-path-to-log-file	<p>ログ ファイル ディレクトリへの完全パス。Glob パターンがサポートされます。構成例：</p> <ul style="list-style-type: none"> <li>■ D:\Logs\new_test_logs ディレクトリのすべてのサブディレクトリから収集するには、directory=D:\Logs\new_test_logs\* を使用します。</li> <li>■ サブディレクトリ自身にサブディレクトリがある場合は、次の構成を使用してすべてのサブディレクトリを監視します。directory=D:\Logs\new_test_logs\*\*</li> </ul> <p><b>注：</b> ファイルとディレクトリ数を制限し、リソースの消費を抑える場合、最初または 2 番目のレベルのディレクトリ（たとえば directory=c:/tmp/* または directory=c:\Logs\*）のディレクトリ glob を定義することはできません。ディレクトリパスは 2 つ以上のレベルにする必要があります。</p> <p>まだ存在しないディレクトリのパスを定義できます。実際にディレクトリとファイルが作成されると、エージェントはそのディレクトリ内のログ ファイルを収集します。</p> <p>1 つまたは複数の異なる構成セクションで同一のディレクトリを定義し、同一ファイルから繰り返しログを収集することができます。この処理により、同一イベントのソースに異なるタグやフィルタを適用することができます。</p> <p><b>注：</b> これらのセクションに同一の構成を使用すると、重複したイベントがサーバー側で確認されます。</p>
include=file_name; ...	<p>（オプション）データを収集するファイル名またはファイル マスク（glob パターン）。複数の値をセミコロンで区切って指定することができます。デフォルトの値は *で、これはすべてのファイルが含まれることを意味します。パラメータは大文字と小文字が区別されます。</p> <p>ファイル マスク（グロブ パターン）を使用して、同じ命名規則に従う、単一のファイル名で指定されるファイルをグループ化することができます。たとえば、vRealize Ops Analytics.log および vRealize Ops Collector.log のようにファイル名にスペースが含まれる場合は、vRealize?Ops?Analytics*.log または vRealize*.log のように指定できます。ファイル マスクを使用すると、Linux および Windows ホストのエージェント構成で、使用可能なファイル名を指定できます。</p> <p>デフォルトでは、.zip ファイルおよび .gz ファイルが収集から除外されています。</p> <p><b>重要：</b> ローテーションされるログ ファイルを収集している場合、include および exclude パラメータを使用してプライマリ ファイルとローテーションされるファイルの両方に一致する glob パターンを指定します。glob パターンがプライマリ ログ ファイルにのみ一致する場合、vRealize Log Insight エージェントはローテーション中にイベントを収集できない場合があります。vRealize Log Insight エージェントは自動的にローテーション ファイルの正しい順序を判断し、vRealize Log Insight サーバに正しい順序でイベントを送信します。たとえば、プライマリ ログ ファイルの名前が myapp.log で、ローテーション ログが myapp.log.1 および myapp.log.2 と続く場合、次の include パターンを使用できます。</p> <pre>include= myapp.log;myapp.log.*</pre>

パラメータ	説明
<b>exclude=regular_expression</b>	(オプション) 収集から除外するファイル名またはファイル マスク (glob パターン)。複数の値をセミコロンで区切って指定することができます。デフォルトの値は空で、これは除外するファイルがないことを意味します。
<b>event_marker=regular_expression</b>	<p>(オプション) ログ ファイル内のイベントの開始を示す正規表現。省略した場合、デフォルトは改行です。入力する表現には、Perl 正規表現構文を使用する必要があります。</p> <p><b>注：</b> 引用符 (" ") のような文字は、正規表現ではラッパーとみなされることはありません。これらはパターンの一部とみなされます。</p> <p>vRealize Log Insight エージェントはリアルタイム収集用に最適化されているため、内部遅延によって書き込まれた部分的なログ メッセージが複数のイベントに分割されることがあります。ログ ファイルの追加が 200 ミリ秒以上停止していて、その間に新しい event_marker が観察されなかった場合は、部分的なイベントが完了、解析済み、および配信済みとして処理されます。このタイミング ロジックは構成することができず、event_marker 設定よりも優先します。ログ ファイル アペンダによってすべてのイベントがフラッシュされます。</p>
<b>enabled=yes no</b>	(オプション) 構成セクションを有効または無効にするパラメータ。設定可能な値は yes または no です。デフォルト値は yes です。
<b>charset=char-encoding-type</b>	<p>(オプション) エージェントが監視するログ ファイルの文字エンコーディング。値は次のとおりです。</p> <ul style="list-style-type: none"> <li>■ UTF-8</li> <li>■ UTF-16LE</li> <li>■ UTF-16BE</li> </ul> <p>デフォルト値は UTF-8 です。</p>
<b>tags={"tag-name": "tag-value", ...}</b>	<p>(オプション) 収集したイベントのフィールドにカスタム タグを追加するためのパラメータ。JSON 表記を使用してタグを定義します。タグ名には文字、数字、アンダースコアを含めることができます。タグ名の先頭は文字またはアンダースコアにする必要があります、タグ名の長さは 64 文字以内にする必要があります。タグ名は、大文字と小文字が区別されません。たとえば、tags={"tag_name1": "tag value 1", "Tag_Name1": "tag value 2"} を使用する場合、Tag_Name1 は重複として無視されます。event_type および timestamp をタグ名として使用することはできません。同一の宣言内での重複は無視されます。</p> <p>ターゲットが Syslog サーバの場合、タグで APP-NAME フィールドをオーバーライドできます。例: tags={"appname": "VROPS"}</p>
<b>exclude_fields</b>	<p>(オプション) 収集から個別のフィールドを除外するパラメータ。複数の値をセミコロン区切りリストまたはコンマ区切りリストとして指定することができます。次に例を示します。</p> <ul style="list-style-type: none"> <li>■ exclude_fields=hostname; filepath</li> <li>■ exclude_fields=type; size</li> <li>■ exclude_fields=type, size</li> </ul>
<b>raw_syslog=Yes No</b>	Syslog プロトコルを使用するエージェントの場合、このオプションを使用すると、未加工の Syslog イベントの収集と送信をエージェントに許可します。デフォルトは No です。収集されたイベントは、指定した Syslog 属性を使用して変換されます。Syslog 変換を行わずにイベントを収集するには、このオプションを有効にします。

### 3 liagent.ini ファイルを保存して閉じます。

#### 例：構成

```
[filelog|messages]
directory=/var/log
```

```
include=messages;messages.?  
  
[filelog|syslog]  
directory=/var/log  
include=syslog;syslog.?  
  
[filelog|Apache]  
directory=/var/log/apache2  
include=*
```

## イベントのフィルタリング

vRealize Log Insight Linux エージェントで収集されたすべてのイベントをフィールド値に基づいてフィルタリングして、どのログ イベントを選択またはドロップするかを指定できます。whitelist および blacklist コレクタ オプションを使用して、フィルタを定義できます。

---

**ヒント:** デフォルトでは、vRealize Log Insight Linux エージェントはプログラムまたはエディタによって作成された隠しファイルを収集します。隠しファイルの名前はピリオドで始まります。vRealize Log Insight Linux エージェントが隠しファイルを収集しないようにするには、**exclude=.\*** パラメータを追加します。

---

各イベントについて、コレクタは whitelist および blacklist フィルタ式を評価します。whitelist 式が true に評価され、blacklist 式が false に評価されるか、または評価できない場合、イベントはキューに移動してさらに処理されます。それ以外の場合は、コレクタによってイベントがドロップされます。whitelist 式のデフォルト値は true で、blacklist 式のデフォルト値は false です。

---

**ヒント:** Filelog コレクタでは、フィルタリングのフィールドが少なくなります。フィルタリングのフィールドを取得するには、ログを解析します。詳細については、[ログの解析](#) を参照してください。

---

whitelist または blacklist フィルタは、単一の論理値または整数値に評価される一連の変数、リテラル、および演算子です。イベント フィールドは変数として、二重引用符で囲んだ文字列および数字はリテラルとして使用します。フィルタ式内で使用できる演算子の詳細については、[イベントのフィールドおよび演算子](#) を参照してください。

#### 注：

- 数値を文字列と比較する場合、または比較に数値文字列が含まれている場合は、各文字列が数値に変換され、比較は数値で実行されます。例：
  - 式 `whitelist = 123.0 == "000123"` は true に評価されます。
  - 式 `whitelist = "00987" == "987.00"` は true に評価されます。
  - 式 `whitelist = response_size >= "12.12"` で、`response_size` フィールドに数値が指定されている場合、式は数値として評価されます。応答のサイズが 12.12 より大きい場合、式は true、それ以外の場合は false になります。
  - 式 `whitelist = "09123" < "234"` では、両方の文字列リテラルが数値に変換され、式が false に評価されます。
- 文字列のいずれかのオペランドを数値に変換できない場合は、両方のオペランドが文字列に変換されます。単純な大文字/小文字の区別による辞書比較が実行されます。例：
  - 式 `whitelist = "1234a" == "1234A"` は文字列比較で、false に評価されます。
  - 式 `whitelist = 4 < "four"` は 4 から "4" に変換され、true に評価されます。
  - 式 `whitelist = response_size > "thousand"` では、`response_size` フィールドの値が文字列値に変換されます。これにより、式が false に評価されます。
- フィルタ式が整数値として評価されると、0 の場合は false として、そうでない場合は true として扱われます。たとえば、`some_integer` フィールドに最下位ビットが設定されている場合、式 `whitelist = some_integer & 1` は true に、それ以外の場合は false に評価されます。

イベント フィールドおよび演算子の完全なリストについては、[ログ ファイルからのイベントの収集](#) を参照してください。

この例では、ファイル `/var/log/httpd/access` から Apache アクセス ログを収集します。ファイルのサンプル ログには次のものがあります。

- `127.0.0.1 - frank [10/Oct/2016:13:55:36 +0400] "GET /apache_pb.gif HTTP/1.0" 200 2326`
- `198.51.100.56 - john [10/Oct/2016:14:15:31 +0400] "GET /some.gif HTTP/1.0" 200 8270`
- `198.51.100.12 - smith [10/Oct/2016:14:15:31 +0400] "GET /another.gif HTTP/1.0" 303 348`
- `198.51.100.32 - test [10/Oct/2016:15:22:55 +0400] "GET /experimental_page.gif HTTP/1.0" 400 46374`

- 127.0.0.1 - test [10/Oct/2016:15:22:57 +0400] "GET /experimental\_page2.gif HTTP/1.0" 301 100

#### 前提条件

- root としてログインするか、または sudo を使用してコンソール コマンドを実行します。
- vRealize Log InsightLinux エージェントがインストールされた Linux マシンにログインし、コンソールを開き、pgrep liagent を実行して、vRealize Log Insight Linux エージェントがインストールされて実行中であることを確認します。

#### 手順

- 1 次のスニペットに示すように、ログのパースャーを定義します。

```
[parser|apache-access]
base_parser=clf
format=%h %l %u %t \"%r\" %s %b
```

定義したパースャーは、ファイル /var/log/httpd/access から収集されたすべてのイベントについて、remote\_host、 remote\_log\_name、 remote\_auth\_user、 timestamp、 request、 status\_code、 および response\_size フィールドを抽出します。これらのフィールドを使用して、イベントをフィルタリングできます。

- 2 任意のテキスト エディタで /var/lib/loginsight-agent/liagent.ini ファイルを開きます。
- 3 次のスニペットに示すように、ログを収集して解析するためのファイル内の Filelog セクションを定義します。

```
[filelog|apache-access]
directory = /var/log/httpd/
include = access
parser = apache-access
```

- 4 要件に応じてイベントをフィルタリングします。

- HTTP ステータスが 200 のログを収集するには、次のスニペットに示すように Filelog セクションに whitelist を定義できます。

```
[filelog|apache-access]
directory = /var/log/httpd/
include = access
parser = apache-access
whitelist = status_code == 200
```

whitelist 式は、サンプル ログの最初と 2 番目のイベントについてのみ true に評価され、コレクタはこれらのイベントを選択します。

status\_code フィールドがログに存在しないか、解析されていないためにイベント内に存在しない場合は、whitelist 式を評価できません。これは false と評価され、コレクタによってイベントがドロップされます。



- 関心のないイベントをドロップするには、`blacklist` オプションを使用します。たとえば、ローカルトラフィックに関心がない場合は、次のスニペットに示すようにローカル IP アドレスをブロックすることができます。

```
[filelog|apache-access]
directory = /var/log/httpd/
include = access
parser = apache-access
blacklist = remote_host == "127.0.0.1"
```

コレクタは、サンプル ログから 2 番目、3 番目、4 番目のイベントを選択します。

- 複数の述語に基づいてイベントをフィルタリングするには、`or` および `and` 演算子を使用できます。たとえば、次のスニペットに示すように、ローカル IP アドレスから生成されたイベント、またはテスト ユーザーによって生成されたイベントを不要なホストから削除できます。

```
[filelog|apache-access]
directory = /var/log/httpd/
include = access
parser = apache-access
blacklist = remote_host == "127.0.0.1" or remote_auth_user == "test"
```

`or` 演算子を使用すると、`blacklist` 式が `true` に評価され、不要なイベントがスキップされます。式は、`remote_host` フィールド値が「127.0.0.1」または `remote_auth_user` フィールド値が「test」の場合にイベントをドロップするようにコレクタに指示します。

コレクタは、サンプル ログから 2 番目と 3 番目のイベントを選択します。

- テスト ユーザーによってローカル IP アドレスから生成されたイベントをドロップするには、次のスニペットに示すように、`blacklist` 式で `and` を使用できます。

```
[filelog|apache-access]
directory = /var/log/httpd/
include = access
parser = apache-access
blacklist = remote_host == "127.0.0.1" and remote_auth_user == "test"
```

コレクタは、サンプル ログから 5 番目のイベントをドロップします。

- `whitelist` と `blacklist` フィルタを一緒に使用できます。たとえば、応答サイズが 1024 バイトを超えるイベントを必要とするが、ローカル ホストから発生するイベントを必要としない場合は、次のスニペットを使用できます。

```
[filelog|apache-access]
directory = /var/log/httpd/
include = access
parser = apache-access
whitelist = response_size > 1024
blacklist = remote_host == "127.0.0.1" or remote_host == "localhost"
```

コレクタは、サンプル ログから 2 番目のイベントを選択します。

## journalld からのイベントの収集

vRealize Log Insight 4.6 以降では、エージェントが journalld システム サービスのログから、systemd を実行している Linux ディストリビューションのログ データを読み取ることができるようになりました。journalld は、systemd ベースの Linux プラットフォームでのログ用のデフォルト サービスとなりました。journalld の構成セクションでは、次のオプションをサポートしています。

### journal\_files

監視するジャーナル ファイル。次の値がサポートされています。

値	説明
all	使用可能なすべてのジャーナル ファイルを開いて監視します。
local	ローカル マシンで生成されたジャーナル ファイルのみを監視し、読み取ります。
runtime	永続的なストレージ内のファイルを除いた、揮発性ジャーナル ファイルのみを監視し、読み取ります。
system	システム サービスとカーネル ジャーナル ファイルのみを監視し、読み取ります。
user	現在のユーザーのジャーナル ファイルのみを監視し、読み取ります。

### fetch\_fields

ジャーナル ログ エントリからのメッセージとともに取得されるフィールド。このオプションの値は、カンマ区切りのフィールド名の大文字と小文字を区別しないリストです。次の値がサポートされています。

値	説明
pri_severity,pri_facility,syslog_identifier	このオプションのデフォルト値です。
*	すべてのフィールドを取得します。
all	フィールドを取得しません。

## vRealize Log Insight エージェントからのイベントのフィルタリング

ローカルの liagent.ini ファイルの [server|<dest\_id>] セクションにあるフィルタ オプションを使用し、エージェントが宛先に送信する情報を提供できます。

オプションは、次の形式です。

```
filter = {collector_type; collector_filter; event_filter}
```

フィルタの タイプ	説明
collector_ type	コレクタ タイプを定義するコンマ区切りのリスト。サポートされる値は、filelog または winlog です。値が提供されていない場合は、すべてのコレクタ タイプが使用されます。
collector_ filter	コレクタ セクションの名前を正規表現形式で指定します。たとえば、 <i>Vcops_.*</i> は、「vcops_」で始まるすべてのコレクタ セクションを意味します。
event_ filter	イベント フィールドのフィルタは、コレクタ セクションの許可リストまたは拒否リストと同じ構文を使用します。エージェントは、式が True またはゼロ以外の値に評価されるイベントのみを送信します。空の event_filter は常に True と評価されます。イベントで event_filter を使用するには、フィールド抽出のために適切なコレクタ セクション内でパーサーが定義されている必要があります。収集されたイベントにフィールドがないために式を評価できない場合、そのイベントはドロップされます。

複数のフィルタ式を指定する場合は、次の例に示すようにコンマで区切ります。

```
filter=
{winlog;Micr.*;},{filelog;apache-access;level=="error"}
```

メッセージが、宛先ターゲットのフィルタ基準セットを複数満たしている場合、メッセージは 1 回だけ送信されます。

表 4-1. 構文の例

フィルタ	意味
filter= {winlog;Microsoft.*;}	イベント名が Microsoft で始まる場合のみ winlog コレクタからイベントを送信します。
filter= {winlog;Microsoft.*; eventid == 1023}	イベント名が Microsoft で始まり、イベント ID が 1023 に等しい場合のみ、winlog コレクタからイベントを送信します。
filter= {;.};	デフォルトのフィルタ値です。すべてのソースからすべてのイベントを送信します。
filter= {winlog;.*;}	winlog セクションからすべてのイベントを送信します。
filter= {filelog;syslog;facility<5}	facility が 5 未満の場合、[filelog syslog] セクションからイベントを送信します。[filelog syslog] セクションには facility フィールドを抽出するパーサーが必要です。それ以外の場合、すべてのイベントはスキップされます。
filter= {;;}	一致するイベントがありません。イベントの転送を無効にするには、この構文を使用します。

次の例では、前の例の 2 番目の宛先の構成にフィルタを追加します。

```
; The second destination receives just syslog events through the plain syslog protocol.
[server|syslog-audit]
hostname=third_party_audit_management.eng.vmware.com
proto=syslog
ssl=no
filter= {filelog; syslog; }
```

次の例では、より複雑なフィルタ式を使用します。

```
; This destination receives vRealize Operations Manager events if they have the level field
equal
```

```
;to "error" or "warning" and they are collected by sections whose name begins with "vrops-"

[server|licf-prod1]
hostname=vrops-errors.licf.vmware.com
filter= {; vrops-.*; level == "error" || level == "warning"}
```

複数のフィルタ式を指定する場合は、次の例に示すようにコンマで区切ります。

```
filter= e.
{winlog;Micr.*;},{filelog;apache-access;level=="error"}
```

## vRealize Log Insight エージェントからの情報の転送

エージェントによって収集されたイベントは最大 3 つの宛先に転送できます。宛先には、vRealize Log Insight サーバ、フォワーダ、またはサードパーティのログ管理ソリューションを含めることができます。

たとえば、監査ログまたはシステム ログをセキュリティ チームのサーバに、アプリケーション ログを開発運用チームのサーバに、メトリック ログを IT 管理システムに送信することができます。フィルタを使用して、どの情報を宛先に送信するかを指定します。1 つの vRealize Log Insight エージェントからの情報を最大 3 つの宛先に転送できます。

エージェントの構成は、ローカルの `liagent.ini` ファイルの `[server|<dest_id>]` セクションを介して実行されます。vRealize Log Insight サーバまたはフォワーダには `cfapi` プロトコルを使用し、その他のターゲットまたは宛先には `Syslog` を使用します。

エージェントに複数の宛先を指定する場合、最初の宛先はデフォルトの `loginsight` の場所を使用します。その他の宛先の位置情報を指定する必要があります。

次の例は、`liagent.ini` ファイルの 2 つの宛先を指定する部分を示しています。デフォルトのサーバ名 `loginsight` はデフォルトで最初の宛先に暗黙的に適用されるので、指定されていません。2 番目の `[server|<dest_id>]` セクションは、宛先を指定します。

```
; The first (default) destination receives all collected events.
[server]
ssl=yes

; The second destination receives just syslog events through the plain syslog protocol.
[server|syslog-audit]
hostname=third_party_audit_management.eng.vmware.com
proto=syslog
ssl=no
```

エージェントのフィルタを作成する方法の詳細については、[vRealize Log Insight エージェントからのイベントのフィルタリング](#)を参照してください。

## ターゲット vRealize Log InsightServer の設定

Windows で実行されている vRealize Log Insight エージェントのターゲット vRealize Log Insight サーバを設定または変更できます。最大 3 つの宛先にイベントを送信し、宛先ごとに出力をフィルタできます。

デフォルトの宛先は、liagent.ini ファイルの [server] セクションで構成できます。デフォルトの宛先は常に存在し、ホスト名はデフォルトで loginsight に設定されます。ターゲットの宛先を追加するには、各ターゲットに [server|<dest\_id>] セクションを作成します。追加された接続ごとに、宛先 ID として一意のホスト名を指定する必要があります。追加の宛先に、デフォルトの [server] セクションと同じオプションを使用できます。追加の宛先を自動アップグレード用に設定したり、エージェントの設定に使用したりしないでください。追加の宛先は 2 つ指定できます。

デフォルトでは、エージェントは収集されたイベントをすべての宛先に送信します。file オプションでイベントごとに異なる宛先に送信するために、イベントをフィルタすることができます。詳細については、[vRealize Log Insight エージェントからのイベントのフィルタリング](#)を参照してください。

#### 前提条件

- vRealize Log Insight Windows エージェントをインストールした Windows マシンにログインし、サービスマネージャを起動して vRealize Log Insight エージェント サービスがインストールされていることを確認します。
- 統合ロード バランサが有効な vRealize Log Insight クラスタがある場合は、[統合ロードバランサを有効にする](#)のカスタム SSL 証明書の固有の要件を参照してください。

#### 手順

- 1 vRealize Log Insight Windows エージェントのプログラム データ ディレクトリに移動します。

```
%ProgramData%\VMware\Log Insight Agent
```

- 2 任意のテキスト エディタで liagent.ini を開きます。

- 3 次のパラメータを変更し、使用環境に合わせて値を設定します。

パラメータ	説明
proto	<p>エージェントが、イベントを vRealize Log Insight サーバに送信するために使用するプロトコル。設定可能な値は cfapi および syslog です。</p> <p>デフォルトは cfapi です。</p>
hostname	<p>vRealize Log Insight 仮想アプライアンスの IP アドレスまたはホスト名。</p> <p>IPv4 または IPv6 アドレスを指定することができます。IPv6 アドレスを指定する場合は、角括弧を使用してもしなくてもかまいません。例：</p> <pre>hostname = 2001:cdba::3257:9652 or hostname = [2001:cdba::3257:9652]</pre> <p>ホストが IPv4 と IPv6 の両方のスタックをサポートし、ドメイン名がホスト名として指定されている場合、エージェントは名前リゾルバによって返される IP アドレスに応じて IP スタックを使用します。リゾルバが IPv4 と IPv6 アドレスの両方を返す場合、エージェントは指定された順番で両方のアドレスに接続しようとします。</p>
max_disk_buffer	<p>Log Insight Windows エージェントがこの特定のサーバ用に収集されたイベントをバッファするために使用できる最大ディスク容量 (MB)。このオプションは、このサーバの [storage].max_disk_buffer の値を上書きします。</p> <p>デフォルト値は 150 MB で、バッファ サイズを 50 ～ 8000 MB に設定できます。</p>

パラメータ	説明
<b>port</b>	<p>エージェントが、イベントを vRealize Log Insight またはサードパーティのサーバに送信するために使用する通信ポート。デフォルトでは、エージェントは SSL とプロトコルに設定されたオプションに基づいて適切なポートを使用します。以下のリストに指定されたデフォルトのポート値を参照してください。ポート オプションは、これらのデフォルトと異なる場合にのみ指定する必要があります。</p> <ul style="list-style-type: none"> <li>■ SSL が有効になっている cfapi: 9543</li> <li>■ SSL が無効になっている cfapi: 9000</li> <li>■ SSL が有効になっている syslog: 6514</li> <li>■ SSL が無効になっている syslog: 514</li> </ul>
<b>ssl</b>	<p>SSL を有効または無効にします。デフォルト値は yes です。</p> <p>ssl が [はい] に設定されている場合、特に指定しない限り、ポートは 9543 に設定されます。</p>
<b>reconnect</b>	<p>サーバへ強制的に再接続する時間 (分)。デフォルト値は 30 です。</p>
<b>filter</b>	<p>エージェントによって宛先に送信される情報を指定します。このオプションでは、次の 3 つの引数が使われます。</p> <pre>{collector_type; collector_filter; event_filter}</pre>

```
[server]
hostname=LOGINSIGHT
; Hostname or IP address of your Log Insight server / cluster load balancer. Default:
;hostname=LOGINSIGHT

; Protocol can be cfapi (Log Insight REST API), syslog. Default:
;proto=cfapi

; Log Insight server port to connect to. Default ports for protocols (all TCP):
; syslog: 514; syslog with ssl: 6514; cfapi: 9000; cfapi with ssl: 9543. Default:
;port=9543

; SSL usage. Default:
;ssl=yes
```

4 liagent.ini ファイルを保存して閉じます。

## 例

次の構成例は、信頼された認証局を使用するターゲット vRealize Log InsightServer を設定します。

```
[server]
proto=cfapi
hostname=LOGINSIGHT
port=9543
ssl=yes;
ssl_ca_path=/etc/pki/tls/certs/ca.pem
```

次の例では、宛先ごとのフィルタ メッセージを含む複数の宛先の設定を示します。

```
; The first (default) destination receives all collected events.
```

```
[server]
hostname=prod1.licf.vmware.com

; The second destination receives just syslog events through the plain syslog protocol.
[server|syslog-audit]
hostname=third_party_audit_management.eng.vmware.com
proto=syslog
ssl=no
filter={filelog; syslog; }

; The third destination receives vRealize Operations Manager events if they have the level
field equal to "error" or "warning"
; and they are collected by sections whose name begins with "vrops-"

[server|licf-prod1]
hostname=vrops-errors.licf.vmware.com
filter={; vrops-.*; level == "error" || level == "warning"}

; Collecting syslog messages.
[filelog|syslog]
directory=/var/log
include=messages

; various vROPs logs. Note that all section names begin with a "vrops-" prefix, which is used
in third destination filter.
[filelog|vrops-ANALYTICS-analytics]
directory=/data/vcops/log
include=analytics*.log*
exclude=analytics*-gc.log*
parser=auto

[filelog|vrops-COLLECTOR-collector]
directory=/data/vcops/log
include=collector.log*
event_marker=^\\d{4}-\\d{2}-\\d{2} [\\s]\\d{2}:\\d{2}:\\d{2}\\.\\d{3}
parser=auto

[filelog|vrops-COLLECTOR-collector_wrapper]
directory=/data/vcops/log
include=collector-wrapper.log*
event_marker=^\\d{4}-\\d{2}-\\d{2} [\\s]\\d{2}:\\d{2}:\\d{2}\\.\\d{3}
parser=auto
```

## 次のステップ

vRealize Log Insight エージェント の追加の SSL オプションを構成できます。[サーバと Log Insight Agents 間の SSL 接続の構成](#)を参照してください。

## エージェントの宛先の指定

vRealize Log Insight Linux エージェントがイベントを送信する宛先は 3 つまで指定できます。

複数の宛先の接続を定義する場合には、`li-agent.ini` ファイルの `[server|<dest_id>]` セクションを使用します。`<dest_id>` は、設定の接続 ID ごとに固有の値になります。追加の宛先に、デフォルトの `[server]` セクションと同じオプションを使用できます。ただし、追加の宛先を自動アップグレード用に設定したり、エージェントの設定に使用したりしないでください。追加の宛先は 2 つ指定できます。

定義した最初のターゲットは、デフォルト サーバの値 `loginsight` を使用できます。追加のターゲットを定義する場合は、後続のターゲットの `[server]` セクションにホスト名を指定する必要があります。フィルタを使用しないと、エージェントは収集されたイベントをすべての宛先に送信します。これはデフォルトです。ただし、イベントをフィルタリングすると、イベントごとに異なる宛先に送信することができます。

#### 前提条件

- `root` としてログインするか、または `sudo` を使用してコンソール コマンドを実行します。
- vRealize Log InsightLinux エージェントがインストールされた Linux マシンにログインし、コンソールを開き、`pgrep liagent` を実行して、vRealize Log Insight Linux エージェントがインストールされて実行中であることを確認します。
- 統合ロード バランサが有効な vRealize Log Insight クラスタがある場合は、[統合ロードバランサを有効にする](#)のカスタム SSL 証明書の固有の要件を参照してください。

#### 手順

- 1 任意のテキスト エディタで `/var/lib/loginsight-agent/liagent.ini` ファイルを開きます。
- 2 次のパラメータを変更し、使用環境に合わせて値を設定します。

パラメータ	説明
<b>proto</b>	<p>エージェントが、イベントを vRealize Log Insight サーバに送信するために使用するプロトコル。設定可能な値は <code>cfapi</code> および <code>syslog</code> です。</p> <p>デフォルトは <code>cfapi</code> です。</p>
<b>hostname</b>	<p>vRealize Log Insight 仮想アプライアンスの IP アドレスまたはホスト名。</p> <p>IPv4 または IPv6 アドレスを指定することができます。IPv6 アドレスを指定する場合は、角括弧を使用してもしなくてもかまいません。例：</p> <pre>hostname = 2001:cdba::3257:9652 or hostname = [2001:cdba::3257:9652]</pre> <p>ホストが IPv4 と IPv6 の両方のスタックをサポートし、ドメイン名がホスト名として指定されている場合、エージェントは名前リゾルバによって返される IP アドレスに応じて IP スタックを使用します。リゾルバが IPv4 と IPv6 アドレスの両方を返す場合、エージェントは指定された順番で両方のアドレスに接続しようとします。</p>
<b>max_disk_buffer</b>	<p>Log Insight Linux エージェントがこの特定のサーバ用に収集されたイベントをバッファするための最大ディスク容量 (MB)。このオプションは、このサーバの <code>[storage].max_disk_buffer</code> の値を上書きします。</p> <p>デフォルト値は 150 MB で、バッファ サイズを 50 ～ 8000 MB に設定できます。</p>



パラメータ	説明
<b>port</b>	<p>エージェントが、イベントを vRealize Log Insight またはサードパーティのサーバに送信するために使用する通信ポート。デフォルトでは、エージェントは SSL とプロトコルに設定されたオプションに基づいて適切なポートを使用します。以下のリストに指定されたデフォルトのポート値を参照してください。ポート オプションは、これらのデフォルトと異なる場合にのみ指定する必要があります。</p> <ul style="list-style-type: none"> <li>■ SSL が有効になっている cfapi: 9543</li> <li>■ SSL が無効になっている cfapi: 9000</li> <li>■ SSL が有効になっている syslog: 6514</li> <li>■ SSL が無効になっている syslog: 514</li> </ul>
<b>ssl</b>	<p>SSL を有効または無効にします。デフォルト値は yes です。</p> <p>ssl を yes に設定すると、ポートの値を設定していない場合、自動的に 9543 ポートが割り当てられます。</p>
<b>reconnect</b>	サーバへの再接続を強制するための時間を分数で指定します。デフォルト値は 30 です。

```
[server]
hostname=LOGINSIGHT
; Hostname or IP address of your Log Insight server / cluster load balancer. Default:
;hostname=LOGINSIGHT

; Protocol can be cfapi (Log Insight REST API), syslog. Default:
;proto=cfapi

; Log Insight server port to connect to. Default ports for protocols (all TCP):
; syslog: 514; syslog with ssl: 6514; cfapi: 9000; cfapi with ssl: 9543. Default:
;port=9543

; SSL usage. Default:
;ssl=yes
```

### 3 liagent.ini ファイルを保存して閉じます。

#### 例

次の構成例は、信頼された認証局を使用するターゲット vRealize Log InsightServer を設定します。

```
[server]
proto=cfapi
hostname=LOGINSIGHT
port=9543
ssl=yes;
ssl_ca_path=/etc/pki/tls/certs/ca.pem
```

次の例では、複数の宛先を設定しています。

- 最初の宛先（デフォルト）は、収集されたすべてのイベントを受信します。

```
[server]
hostname=prod1.licf.vmware.com
```

- 2 つ目の宛先は、ブレーンの Syslog プロトコル経由で Syslog イベントのみを受信します。

```
[server|syslog-audit]
hostname=third_party_audit_management.eng.vmware.com
proto=syslog
ssl=no
filter= {filelog; syslog; }
```

- 3 つ目の宛先は、レベル フィールドが「error」または「warning」で、「vrops」で始まるセクション名で収集された vRealize Operations Manager イベントを受信します。

```
[server|licf-prod1]
hostname=vrops-errors.licf.vmware.com
filter= {; vrops-.*; level == "error" || level == "warning"}

;Collecting syslog messages.
[filelog|syslog]
directory=/var/log
include=messages

;various vRops logs. Note that all section names begin with "vrops-" prefix, which is used in
third destination filter.
[filelog|vrops-ANALYTICS-analytics]
directory=/data/vcops/log
include=analytics*.log*
exclude=analytics*-gc.log*
parser=auto
[filelog|vrops-COLLECTOR-collector]
directory=/data/vcops/log
include=collector.log*
event_marker=^\d
{4}-\d{2}-\d{2}[\s]\d{2}:\d{2}:\d{2}\,\d{3}
parser=auto

[filelog|vrops-COLLECTOR-collector_wrapper]
directory=/data/vcops/log
include=collector-wrapper.log*
event_marker=^\d{4}
-\d
{2}-\d{2}
[\s]\d
{2}:\d{2}
:\d
{2}
\.\d
{3}
parser=auto
```

#### 次のステップ

vRealize Log InsightLinux エージェント の追加の SSL オプションを構成できます。サーバと Log Insight Agents 間の SSL 接続の構成を参照してください。

## vRealize Log Insight エージェントの構成の一元化

複数の vRealize Log Insight エージェントを構成することができます。

各 vRealize Log Insight エージェントには、ローカル構成とサーバ側の構成があります。ローカル構成は、vRealize Log Insight エージェントがインストールされている仮想または物理マシンの `liagent.ini` ファイル内に格納されています。サーバ側の構成は、Web ユーザー インターフェイスの [管理] - [エージェント] からアクセスおよび編集できます。各 vRealize Log Insight エージェントはセクションとキーから構成されています。キーには構成可能な値があります。

vRealize Log Insight エージェントは、vRealize Log Insight サーバを定期的にポーリングし、サーバ側の構成を受け取ります。サーバ側構成およびローカル構成はマージされ、その結果が有効な構成になります。各 vRealize Log Insight エージェントは、この有効な構成を運用上の構成として使用します。構成のセクションはセクションを基準としてマージされ、キーはキーを基準としてマージされます。サーバ側構成の値は、ローカル構成の値よりも優先されます。マージ ルールは次のとおりです。

- セクションがローカル構成内のみにある場合、またはサーバ側構成内のみにある場合、このセクションとそのすべての内容が有効な構成に含められます。
- セクションがローカル構成内とサーバ側構成内の両方にある場合、このセクション内のキーは、次のルールに従ってマージされます。
  - キーがローカル構成内のみ、またはサーバ側構成内のみにある場合、このキーとその値が、有効な構成のこのセクションに含められます。
  - キーがローカル構成内とサーバ側構成内の両方にある場合、このキーは有効な構成のこのセクションに含められ、サーバ側構成内の値が使用されます。

vRealize Log Insight の管理者ユーザーは、一元化された構成をすべての vRealize Log Insight エージェントに適用できます。たとえば、[[管理]] ページに移動して [[管理]] セクションの [エージェント] をクリックします。構成設定を [エージェント構成] ボックス内に入力し、[すべてのエージェントの構成を保存] をクリックします。この構成は、次のポーリングサイクル中に構成可能なすべてのアクティブ エージェントに適用されます。

また、管理者 vRealize Log Insight ユーザーは、OS、エージェント バージョン、ホスト名、IP アドレス範囲などの特定のフィルタをエージェント グループ内で使用したり、特定の vRealize Log Insight エージェントに構成を適用したりできます。エージェント グループの詳細については、「[エージェント グループの操作](#)」を参照してください。

---

### 注：

- 一元化された構成は、cfapi プロトコルを使用する vRealize Log Insight エージェントのみに適用できます。
  - vRealize Log Insight エージェントは、次のいずれかのシナリオでは構成できません。
    - 現在の vRealize Log Insight サーバはプライマリ ターゲットではありません。複数の転送先の構成方法については、[エージェントの宛先の指定](#) を参照してください。
    - パラメータ `central_config = no` は、エージェントの構成で使用されます。Windows のデフォルトのエージェント構成の詳細については、[Log Insight Windows Agent のデフォルト構成](#) を参照してください。
-

## 構成結合の例

Log Insight Windows Agent のローカル構成とサーバ側の構成を結合する例。

### ローカル構成

次の Log Insight Windows Agent ローカル構成を使用できます。

```
[server]
proto=cfapi
hostname=HOST
port=9000

[winlog|Application]
channel=Application

[winlog|Security]
channel=Security

[winlog|System]
channel=System

[filelog|ApacheAccessLogs]
enabled=yes
directory=C:\Program Files (x86)\Apache Software Foundation\Apache2.2\logs
include=*.log
exclude=*_old.log
event_marker=^(\\d{1,3}\\.){3}\\d{1,3} - -
```

### サーバ側の構成

Web ユーザー インターフェイスの [管理] - [エージェント] ページでは、すべてのエージェントに一元構成を適用できます。たとえば、収集チャンネルの除外や追加、デフォルトの再接続設定の変更を行うことができます。

```
[server]
reconnect=20

[winlog|Security]
channel=Security
enabled=no

[winlog|Microsoft-Windows-DeviceSetupManagerOperational]
channel=Microsoft-Windows-DeviceSetupManager/Operational
```

### 有効な構成

有効な構成はローカルおよびサーバ側の構成を結合した結果の構成です。Log Insight Windows Agent は次のように構成されます。

- vRealize Log Insight サーバを 20 分ごとに再接続する
- アプリケーションおよびシステム イベント チャンネルの収集を続行する
- セキュリティ イベント チャンネルの収集を停止する

- Microsoft-Windows-DeviceSetupManager/Operational イベント チャンネルの収集を開始する
- ApacheAccessLogs の収集を続行する

```
[server]
proto=cfapi
hostname=HOST
port=9000
reconnect=20

[winlog|Application]
channel=Application

[winlog|Security]
channel=Security
enabled=no

[winlog|System]
channel=System

[winlog|Microsoft-Windows-DeviceSetupManagerOperational]
channel=Microsoft-Windows-DeviceSetupManager/Operational

[filelog|ApacheAccessLogs]
enabled=yes
directory=C:\Program Files (x86)\Apache Software Foundation\Apache2.2\logs
include=*.log
exclude=*_old.log
event_marker=^(\d{1,3}\.){3}\d{1,3} - -
```

## エージェント構成に対する共通値の使用

Windows および Linux エージェントで、エージェント構成ファイルのデフォルト値を各エージェント構成セクションに適用される共通パラメータ値でオーバーライドできます。

### 共通オプション

liagent.ini 構成ファイルの [common|global] セクションで指定されたオプションはすべてのセクションに伝達され、[common|filelog] セクションで指定されたオプションは filelog セクション全体にのみ伝達され、[common|winlog] オプションは winlog セクション全体にのみ伝達されます。

tags, include, exclude, event\_marker, charset, exclude\_fields および parser パラメータを以下の例のように共通セクションで定義できます。次の例は Windows エージェントのものです。

```
[common|global]

tags = {"log_source_vm":"win-2008r2-64"}
exclude_fields = test_tag;some_other_tag
parser = auto

[common|filelog]
tags = {"collector_type":"filelog"}
```

```
exclude = *.trc

[filelog|channel_1]
directory = C:\app\log
include = *.log

...
```

この例では、以下の動作が指定されます。

- filelog セクションからのすべてのログには、対応する値を持つ log\_source\_vm tags と collector\_type tags の両方があります。
- test\_tag tags と some\_other\_tag tags は、送信されるすべてのログから除外されます。
- auto パーサは収集されたすべてのログに適用されます。
- すべての filelog コレクタはデフォルトで \*.trc ファイルを監視から除外します。

[common|global] のオプションもすべての winlog セクションに適用されます。

## 基準のマージとオーバーライド

オプションが複数のセクションで定義されると、その値はマージまたはオーバーライドされ、マージ/オーバーライド時には範囲が小さいセクションが優先されます。つまり、[common|global] からの値が [common|filelog]] からの値とマージまたはオーバーライドされ、次に [filelog|sample\_section] からの値と結合またはオーバーライドされます。

マージおよびオーバーライドの動作は以下のルールに従います。

- その値が値 (tags、include、exclude および exclude\_fields) のリストを表すオプションは、優先順位が高いセクションからのオプションの値とマージされます。tags の場合、前述されているように、優先順位が高いセクションからの tags の値が優先順位が低いセクションからの同じ tag の値をオーバーライドします。
- 単一の値 (event\_marker、charset および parser) を持つことができるオプションの値は、優先順位が高いセクションからのオプションの値によってオーバーライドされます。

つまり、[filelog|sample\_section] からの charset=UTF-8 の値が [common|global] からの charset=UTF-16LE のグローバル値をオーバーライドします。

たとえば、[common|filelog] に tags={"app":"global-test"}、[filelog|flg\_test\_section] に tags={"app":"local-test","section":"flg\_test\_section"} がある場合、[filelog|flg\_test\_section] セクションからの "app" タグの値が [common|filelog] からの値をオーバーライドします。この filelog セクションで収集されたすべてのログには "local-test" 値を持つ追加の "app" タグと、"flg\_test\_section" 値を持つ "section" タグが含まれます。winlog セクションの場合、優先順位のチェーンは最も優先度が高い [winlog|...] セクションと、最も優先度が低い [common|global] セクションで同じです。

無効な値が共通セクションで指定されると、それらの値は通常スキップされ、前の対応する filelog/winlog セクションからの値とマージされません。tags または exclude\_fields オプションの値が無効な場合、エージェントは可能な限り多くの有効データを抽出し、無効データがあった場合は残りのファイルをスキップします。すべての異常がエージェント ログ ファイルに報告されます。予期しない動作が発生した場合はログ ファイルを確認し、エージェントで報告されたすべてのエラーを修正します。

エージェントは filelog セクションまたは winlog セクションのオプションに対し無効な値を検出した場合、そのセクションからのオプション値と共通セクションからのオプション値をマージせず、そのセクションを有効化しません。すべてのエラーがエージェント ログ ファイルに報告されます。予期しない動作が発生した場合はログファイルを確認し、エージェントで報告されたすべてのエラーを修正します。

## ログの解析

エージェント側のパーサは、vRealize Log Insight サーバへの配信前に加工前のログから構造化データを抽出します。vRealize Log Insight は、ログ パーサを使用して分析したログから情報を抽出し、サーバにその分析結果を表示することができます。ログ パーサは Windows および Linux vRealize Log Insight Agent のいずれに対しても構成することができます。

Syslog プロトコルを使用する場合、パーサによって抽出されたフィールドは、RFC5424 に従って STRUCTURED-DATA の一部になります。

## ログ パーサの構成

パーサは、FileLog と WinLog コレクタの両方に構成できます。

### 前提条件

vRealize Log Insight Linux Agent の場合：

- root としてログインするか、または sudo を使用してコンソール コマンドを実行します。
- Log Insight Linux Agent がインストールされた Linux マシンにログインし、コンソールを開き、pgrep liagent を実行して Log Insight Linux Agent がインストールされて実行中であることを確認します。

vRealize Log Insight Windows Agent の場合：

- Log Insight Windows Agent をインストールした Windows マシンにログインし、サービス マネージャを起動して vRealize Log Insight Agent サービスがインストールされていることを確認します。

### 手順

- 1 liagent.ini ファイルを含むフォルダに移動します。

オペレーティング システム	パス
[Linux]	/var/lib/loginsight-agent/
[Windows]	%ProgramData%\VMware\Log Insight Agent

- 2 任意のテキスト エディタで liagent.ini ファイルを開きます。
- 3 特定のパーサを構成するには、パーサ セクションを定義します。[parser|myparser]

ここで、myparser はログ ソースから参照可能なパーサの任意の名前です。パーサ セクションはすべての組み込み（または他のいずれの定義された）パーサを参照していて、必要に応じてそのパーサの必須オプションおよび、必須でないオプションを構成している必要があります。

たとえば、`base_parser=csv` は、`myparser` パーサが組み込みのパーサ `csv` から派生していることを示しています。入力ログはセミコロンで区切られた 2 つのフィールドで構成されていることを想定しています。

```
[parser|myparser]

base_parser=csv

fields=field_name1,field_name2

delimiter=";"
```

- 4 `myparser` を定義した後、ログ ソース `winlog` または `filelog` から参照します。

```
[filelog|some_csv_logs]

directory=D:\Logs

include=*.txt;*.txt.*

parser=myparser
```

D:\Logs ディレクトリなどの `some_csv_logs` ソースから収集されたログは `myparser` によって解析され、抽出されたイベントがそれぞれ `field_name1` および `field_name2` としてサーバに表示されます。

**注：** D:\Logs ディレクトリ内の静的なログはエージェントによって vRealize Log Insight に読み込まれません。ただし、D:\Logs で作成した新しいファイルは vRealize Log Insight で利用可能です。

- 5 `liagent.ini` ファイルを保存して閉じます。

## パーサの共通オプション

名前のついたフィールドを生成するすべてのパーサに共通オプションを構成することができます。

### フィールド名の予約語

フィールド名には制限があります。次の名前は予約されているため、フィールド名としては使用できません。

- `event_type`
- `hostname`
- `source`
- `text`

### パーサの共通オプション

次の表のオプションは、サポートされているすべてのパーサで使用できます。



オプション	説明
base_parser	このカスタム パーサが拡張するベース パーサの名前。組み込みのパーサ名か、別のカスタム パーサ名にできます。この構成キーは必須です。
field_decoder	<p>JSON 文字列として指定されたネストされたパーサ。キーはネストされたパーサが適用されるフィールドの名前で、値はそのフィールドに使用されるパーサの名前です。ネストされたパーサは、ベース パーサによってデコードされた、それぞれの対応するフィールドに適用されます。タイムスタンプなどフィールドの値が複雑な場合、フィールド デコーダが便利です。 <b>field_decoder</b> オプションは引数としてより複雑な JSON オブジェクトもサポートし、ネストされたパーサが適用される前にチェックされる特定のフィールド値の条件を使用できます。</p> <p><b>注：</b> 使用方法と条件付き構成の詳細については、以下の「field_decoder オプションの条件付き構成」を参照してください。</p>
field_rename	<p>抽出したフィールド名を変更します。キーがフィールドの元の名前、値がフィールドの新しい名前の JSON 文字列を使用します。</p> <p>field_decoder オプションは常に、field_rename の前に適用されます。オプションの INI ファイル内での順序は重要ではありません。分かりやすくするため、field_decoder を先に指定します。</p>
next_parser	<p>次に実行するパーサの名前。1 回の入力で複数のパーサの連続実行を可能にします。</p> <p><b>注：</b> パーサは、next_parser キーワードで定義されたすべての連続したパーサを処理し、前のパーサによって抽出済みのフィールド値を置き換えることができます。</p>
exclude_fields	サーバへの配信前にイベントから削除するフィールド名のリスト（セミコロン区切り）。フィールド名は、解析中に除外するフィールドがフィルタ条件で使用できないようにするために、イベント フィルタリングの実行前に削除されます。
debug	<p>特定のパーサのデバッグを有効にする Yes または No オプション。デバッグが有効になると、パーサは受信した入力、実行した操作、生成した結果の詳細をログに記録します。このオプションはセクションごとに、つまり特定のセクションで定義されたパーサのみに適用されます。パーサに対する debug のデフォルト値は debug=no です。</p>

### field\_decoder オプションの条件付き構成

特定のフィールド値に関連する共通フォーマットが同じで差異が大きい場合、たとえば **info** および **error** の重大度を含むログの場合、条件付きのネストされたパーサを使用して、すでに解析済みログの対応するフィールドへの不要なパーサの適用を減らすことができます。

たとえば、次のログを使用するとします。

```
2019-03-29T11:00:54.858Z host-FQDN Hostd: error hostd[2099230] [Originator@6876 sub=Default
opID=1983bdbe-cl-800f user=admin.user] AdapterServer caught exception: SSLExceptionE(SSL
Exception: error:140000DB:SSL routines:SSL routines:short read: The connection was closed by
the remote end during handshake.)
```

```
2019-03-29T11:00:55.477Z host-FQDN Hostd: info hostd[6D620B70] ['commonhost' opID=5759adcc-
cf] [transportConnector] -- FINISH task-internal-5726666 -- -- Completed connection restart --
```

次の構成を使用してログを解析できます。

```
[parser|clf_parser]
base_parser=clf
format=%t {%generator_host}i %i: {%log_severity}i %i[ {%thread_id}i] %M
field_decoder={"log_message" : {"log_severity" : {"error" : "error_parser", "info" :
"info_parser"}}}
exclude_fields=log_message

[parser|info_parser]
base_parser=clf
format=[ {%common_info}i] [ {%process}i] %M
field_rename={"log_message" : "info_log_content"}

[parser|error_parser]
base_parser=clfformat=[ {%common_info}i] {%exception_handler}i %i: {%exception_type}i: %i: %
{error_id}i: %i: %i: %i: %M
field_rename={"log_message" : "exception_content"}
```

この構成では、次の結果が生成されます。

```
timestamp=2019-03-29T11:00:54.858000 generator_host="host-FQDN" log_severity="error"
thread_id="2099230" common_info=Originator@6876 sub=Default opID=1983bdbe-cl-800f
user=admin.user exception_handler="AdapterServer" exception_type="SSLExceptionE(SSL
Exception" error_id="140000DB" exception_content="The connection was closed by the remote end
during handshake.)"
```

さらに、**info** ログに対して以下のフィールドが解析されます。

```
timestamp=2019-03-29T11:00:55.477000 generator_host="host-FQDN" log_severity="info"
thread_id="6D620B70" log_message="['commonhost' opID=5759adcc-cf] [transportConnector] --
FINISH task-internal-5726666 -- -- Completed connection restart --" common_info="'commonhost'
opID=5759adcc-cf" process="transportConnector" info_log_content="-- FINISH task-
internal-5726666 -- -- Completed connection restart --"
```

## カンマ区切り値ログパーサ

FileLog および WinLog の両方のコレクタに対応するようにカンマ区切り値 (CSV) パーサを構成できます。

csv パーサで使用可能なオプションは、fields および delimiter です。

## カンマ区切り値パーサのオプション

CSV パーサの構造については以下の情報を参照してください。

オプション	説明
fields	<p>fields オプションでは、ログに存在するフィールドの名前を指定します。リストされたフィールド名の合計数は、ログ内のカンマ区切りフィールドの合計数と一致している必要があります。</p> <p>fields オプションは CSV パーサで必須です。指定されていない場合、解析が実行されません。フィールド値を囲む二重引用符はオプションで、フィールドの内容に依存します。</p> <p>フィールド名はカンマで区切る必要があります。次に例を示します。</p> <pre>fields = field_name1, field_name2, field_name3, field_name4</pre> <p>この定義では名前 field_name1、field_name2、field_name3、および field_name4 は抽出したフィールドに連続して割り当てられることを前提にしています。</p> <p>一部のフィールドが CSV パーサによって省略されてしまう場合は、その名前はリストから省略できます。次に例を示します。</p> <pre>fields = field_name1, , field_name3, field_name4</pre> <p>この場合、パーサはイベントから第 1、第 3、第 4 フィールドのみを抽出し、名前 field_name1、field_name3、および field_name4 をそれぞれ割り当てます。</p> <p>ログにあるフィールドの完全なリストがフィールド オプションで指定されていない場合、パーサは空のリストを返します。たとえば、ログ ファイルに field1、field2、field3、field4、および field5 が含まれていて、fields=field1,field2,field3 のみが指定されていた場合、パーサは空のフィールド リストを返します。</p> <p>CSV パーサに fields=* を使用することはできません。これは、パーサが空のリストを返すためです。特定のフィールドを記述済みとして省略する必要がないかぎり、フィールドの完全なリストを指定する必要があります。</p>
delimiter	<p>delimiter オプションはパーサで使用する区切り文字を指定します。デフォルトでは、csv パーサは区切り文字としてカンマを使用しますが、この区切り文字をセミコロン、スペース、またはその他の特殊文字に変更することができます。定義した区切り文字は二重引用符で囲む必要があります。</p> <p>たとえば、delimiter="," や delimiter=";" のようにします。</p> <p>csv パーサでは、引用符で囲まれた任意の文字のセットが区切り文字としてサポートされます (" " または "asd" など)。ログに使用するフィールド値の区切り文字は、区切り文字のパラメータで定義されたパターンに完全に一致する必要があります。完全に一致しない場合、パーサは失敗します。</p> <p>スペースやタブなどの特殊文字は、それらの文字の前にエスケープ文字を指定すると、csv パーサの区切り文字として定義することができます (\", \s, \t)。たとえば、delimiter="\s" や delimiter=" " のようにします。</p> <p>delimiter オプションは任意です。</p>

## CSV ログ パーサの構成

winlog または filelog ソースのいずれかから収集したログを解析するには、以下の構成を使用します。

```
[filelog|some_csv_logs]
directory=D:\Logs
include=*.txt;*.txt.*
parser=myparser

[parser|myparser]
base_parser = csv
fields = timestamp,field_name1, field_name2, field_name3
delimiter = ","
field_decoder={"timestamp": "tsp_parser"}
```

```
[parser|tsp_parser]
; timestamp is a built-in parser
base_parser=timestamp
; "format" is an option of timestamp parser
format=%Y-%m-%d %H:%M:%S
```

この構成を使用すると、some\_csv\_logs ソースから収集されたログ（directory=D:\Logs ディレクトリなど）は myparser によって解析されます。収集されたログにセミコロンで区切られた値が 3 つ含まれていると、解析されたイベントはそれぞれ field\_name1、field\_name2、および field\_name3 という名前を受け取ります。

次の CSV ログを解析するには:

```
"United States","USA","North America","High income: OECD","Fiscal year end: September 30;
reporting period for national accounts data: CY."
```

CSV パーサ構成を定義します。

```
[parser|csv_log_parser]
base_parser=csv
fields=country_name, country_code, region, income_group, special_notes
```

CSV パーサは以下を返します。

```
country_name=United States
country_code=USA
region=North America
income_group=High income: OECD
special_notes=Fiscal year end: September 30; reporting period for national accounts data: CY.
```

## 共通ログフォーマット (Apache) ログパーサ

FileLog および WinLog の両方のコレクタに共通ログフォーマット (CLF) Apache パーサを構成できます。

### 共通ログフォーマット (Apache) パーサ

デフォルトの CLF パーサでは、以下のフィールドの順序と名前が定義されています。

```
host ident authuser datetime request statuscode bytes
```

パーサ名: clf

CLF パーサ固有のオプションは、format です。

### format オプション

format オプションは、Apache ログを生成するのに使用するフォーマットを指定します。このオプションは必須ではありません。

フォーマットが指定されていない場合、以下のデフォルトの共通ログフォーマットが使用されます。

```
%h %l %u %t \"%r\" %s %b
```

CLF パーサ フォーマットの文字列は正規表現の式を受け付けません。たとえば、式 `\s+` ではなくスペースを指定します。

他のログ フォーマットを解析するには、そのフォーマットをエージェントの構成で指定します。解析されたフィールドはサーバ側に以下の名前が表示されます。

**注：** 変数が必要な場合に、構成で `{VARNAME}` が提供されていないと、フィールドは無視されます。

フィールド	値
'%a':	"remote_ip"
'%A':	"local_ip"
'%B', '%b':	"response_size"
'%C':	フォーマットで指定された変数の名前に依存します
'%c':	フォーマットで指定された変数の名前に依存します
'%D':	"request_time_mcs"
'%E':	"error_status"
'%e':	フォーマットで指定された変数の名前に依存します
'%F', '%f':	"file_name"
'%h':	"remote_host"
'%H':	"request_protocol"
'%i':	フォーマットで指定された変数の名前に依存します
'%k':	"keepalive_request_count"
'%l':	"remote_log_name"
'%L'	"request_log_id"
'%M':	"log_message" (この指定子に到達すると、パーサは入力ログの解析を停止します)
'%m':	"request_method"
'%n':	フォーマットで指定された変数の名前に依存します
'%o':	フォーマットで指定された変数の名前に依存します
'%p':	"server_port". 追加のフォーマットが次の指定子と使用できます: <code>%{format}p</code> 。サポートされているフォーマットは "canonical"、"local"、または "remote" です。"canonical" フォーマットが使用されると、フィールド名は "server_port" のままです。"local" フォーマットが使用されるとフィールド名は "local_server_port" となり、"remote" フォーマットが使用されるとフィールド名は "remote_server_port" となります。
'%P':	"process_id". 追加フォーマットが次の指定子と使用できます: <code>%{format}P</code> 。サポートされているフォーマットは "pid"、"tid"、および "hextid" です。"pid" がフォーマットとして使用されると、フィールド名は "process_id" となります。"tid" および "hextid" フォーマットは "thread_id" という名前のフィールドを生成します

フィールド	値
'%q':	"query_string"
'%r':	"request"
'%R':	"response_handler"
'%s':	要求の最終ステータスを生成する"status_code" もサポートされています。これは "status_code" としてサーバ上に表示されます。
'%t':	<p>"timestamp" は取り込みでイベント タイムスタンプとして機能し、タイムスタンプ パーサを確認します。タイムスタンプ自動検出をオーバーライドするには、日付と時刻のフォーマットを波括弧で指定できます：%{Y-%m-%d %H:%M:%S}t。詳細については、「<a href="#">タイムスタンプ パーサ</a>」を参照してください。</p> <p>CLF パーサのタイムスタンプ フォーマットは "begin:" または "end:" プリフィックスで開始できます。フォーマットが begin: (デフォルト) で始まる場合、時刻は要求処理の始めに取得されます。end: で始まる場合、要求処理の終了に近い、ログ エントリが書き込まれた時刻になります。たとえば、%h %l %u [%{begin:%d/%b/%Y %T}t.%{msec_frac}t] \"%r\" %&gt;s %b のようなフォーマットが CLF パーサでサポートされています。</p> <p>次のフォーマット トークンも CLF パーサのタイムスタンプ フォーマット指定子でサポートされています。</p> <p><b>sec</b></p> <p>Epoch からの秒数。これは、タイムスタンプ パーサの %s 指定子に相当します。</p> <p><b>msec</b></p> <p>Epoch からのミリ秒数。</p> <p><b>usec</b></p> <p>Epoch からのマイクロ秒数。</p> <p><b>msec_frac</b></p> <p>ミリ秒の割合 (タイムスタンプ パーサの %f 指定子に相当します)</p> <p><b>usec</b></p> <p>マイクロ秒の割合 (タイムスタンプ パーサの %f 指定子に相当します)</p> <p>タイムスタンプがフォーマット トークンで示されているログを解析するには、次のフォーマットを構成で使用できます。</p> <pre>format=%h %l %u %{sec}t \"%r\" %s %b format=%h %l %u %{msec}t \"%r\" %s %b format=%h %l %u %{usec}t \"%r\" %s %b</pre> <p>これらのトークンは同じフォーマット文字列で相互に組み合わせることも、タイムスタンプ パーサ フォーマットと組み合わせることもできません。代わりに複数の %{format}t トークンが使用可能です。たとえば、ミリ秒を含むタイムスタンプを使用するには、タイムスタンプ パーサの %f 指定子を使用する以外に、次のタイムスタンプの組み合わせを使用できます： %{d/%b/%Y %T}t.%{msec_frac}t 。</p>
'%T':	"request_time_sec"
'%u':	"remote_auth_user"
'%U':	"requested_url"
'%v':	"server_name"

フィールド	値
'%V':	"self_referential_server_name"
'%X':	"connection_status" は、フォーマットで指定された変数の名前に依存します
'%x':	フォーマットで指定された変数の名前に依存します
'%I':	"received_bytes"
'%O':	"sent_bytes"
'%S':	"transferred_size"

たとえば、CLF パーサを使用して winlog または filelog ソースのいずれかから収集したログを解析する場合には、以下の構成を指定します。

```
[filelog|clflogs]
directory=D:\Logs
include=*.txt
parser=myclf

[parser|myclf]
debug=yes ;Note: use this option only while debugging and set it to 'no' when used in
production.
base_parser=clf
format=%h %l %u %b %t \"%r\" %s
```

この構成を使用すると、clflogs ソースから収集されたログ（directory=D:\Logs ディレクトリなど）は myclf によって解析されます。myclf パーサは、構成に記述されているフォーマットを使用して生成されたログのみを解析します。

パーサに対する debug のデフォルト値は debug=no です。

### CLF を使用して生成されたログの解析

CLF を使用して生成されたログを解析するには、対応するフォーマットを構成で定義する必要があります。次に例を示します。

```
format=%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User_Agent}i\"
```

指定子 %{Referer}i および %{User\_Agent}i を使用する、空でないフィールドは、vRealize Log Insight サーバにそれぞれ referer および user\_agent という名前で表示されます。

### タイムスタンプ パーサと CLF パーサとの統合

Apache ログはカスタム時刻フォーマットで解析できます。

カスタム時刻フォーマットのあるアクセス ログは以下のとおりです。

```
format = %h %l %u %{a, %d %b %Y %H:%M:%S}t \"%r\" %>s %b
```

カスタム時刻フォーマットが指定されていない場合、CLF パーサは自動タイムスタンプ パーサを実行して、自動的に時刻フォーマットを減算することを試みます。指定されている場合、カスタム時刻フォーマットが使用されます。

エラー ログでサポートされているカスタム時刻フォーマットでサポートされているものは以下のとおりです。

カスタム時刻フォーマット	説明	構成フォーマット
%{u}t	マイクロ秒を含む現在時刻	format=[%{u}t] [%l] [pid %P] [client %a] %M
%{cu}t	マイクロ秒を含むコンパクトな ISO 8601 フォーマットの現在時刻	format=[%{cu}t] [%l] [pid %P] [client %a] %M

サポートされているタイムスタンプ指定子の完全なリストについては、[タイムスタンプ パーサ](#)を参照してください。

#### 例：Windows 用の Apache デフォルト アクセス ログの構成

#### 例：Windows 用の Apache デフォルト エラー ログの構成

この例では、Windows 用の Apache v2.4 アクセス ログ構成のフォーマット方法を示します。

```
;ACCESS LOG
;127.0.0.1 - - [13/May/2015:14:44:05 +0400] "GET /xampp/navi.php HTTP/1.1" 200 4023 "http://localhost/xampp/" "Mozilla/5.0 (Windows NT 6.1; WOW64; rv:37.0) Gecko/20100101 Firefox/37.0"
;format=%h %l %u %{d/%b/%Y:%H:%M:%S %z}t \"%r\" %>s %b \"%{Referer}i\" \"%{User_agent}i\"

; Section to collect Apache ACCESS logs
[filelog|clflogs-access]
    directory=C:\xampp\apache\logs
    include=acc*
    parser=clfparsers_apache_access
    enabled=yes

;Parser to parse Apache ACCESS logs
[parser|clfparsers_apache_access]
    debug=yes
    base_parser=clf
    format=%h %l %u %{d/%b/%Y:%H:%M:%S %z}t \"%r\" %>s %b \"%{Referer}i\" \"%{User_agent}i\"
```

次の手順を実行して、アクセス ログ フォーマットを定義します。

#### 1 アクセス ログ フォーマット (httpd.conf) に対応するように Apache を構成します。

```
LogFormat "%h %l %u %{d-%b-%Y:%H:%M:%S}t \"%r\" %a %A %e %k %l %L %m %n %T %v %V %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" combined
```

#### 2 CLF パーサ構成を定義します。

```
;ACCESS LOG
;127.0.0.1 unknown - 21-May-2015:13:59:35 "GET /xampp/navi.php HTTP/1.1" 127.0.0.1 127.0.0.1 - 0 unknown - GET - 1 localhost localhost 200 4023 "http://localhost/xampp/" "-"
[filelog|clflogs-access]
    directory=C:\xampp\apache\logs
    include=acc*,_myAcc*
    parser=clfparsers_apache_access
    enabled=yes
```



```
; Parser to parse Apache ACCESS logs
[parser|clfparsers_apache_access]
  debug=yes
  base_parser=clf
  format=%h %l %u {%d-%b-%Y:%H:%M:%S}t \"%r\" %a %A %e %k %l %L %m %n %T %v %V %>s %b \"%
{Referer}i\" \"%{User-Agent}i\"
```

CLF パーサは以下を返します。

```
remote_host=127.0.0.1
timestamp=2015-05-13T10:44:05
request=GET /xampp/navi.php HTTP/1.1
status_code=200
response_size=4023
referer=http://localhost/xampp/
user_agent=Mozilla/5.0 (Windows NT 6.1; WOW64; rv:37.0) Gecko/20100101 Firefox/37.0
```

この例では、Windows 用の Apache v2.4 エラー ログの構成のフォーマット方法を示します。

```
;ERROR LOG
;[Wed May 13 14:37:17.042371 2015] [mpm_winnt:notice] [pid 4488:tid 272] AH00354: Child:
Starting 150 worker threads.
;[Wed May 13 14:37:27.042371 2015] [mpm_winnt:notice] [pid 5288] AH00418: Parent: Created
child process 3480
;format=[%{a %b %d %H:%M:%S%f %Y}t] [%m:%{severity}i] [pid %P:tid %{thread_id}i] %E: %M
;format=[%{a %b %d %H:%M:%S%f %Y}t] [%m:%{severity}i] [pid %P] %E: %M

; Section to collect Apache ERROR logs
[filelog|clfflogs-error]
  directory=C:\xampp\apache\logs
  include=err*
  parser=clfparsers_apache_error
  enabled=yes

;Parser to parse Apache ERROR logs
[parser|clfparsers_apache_error]
  debug=yes
  base_parser=clf
  format=[%{a %b %d %H:%M:%S%f %Y}t] [%m:%{severity}i] [pid %P:tid %{thread_id}i] %E: %M
  next_parser=clfparsers_apache_error2

;Parser to parse Apache ERROR logs
[parser|clfparsers_apache_error2]
  debug=yes
```

```
base_parser=clf
format=[%{a %b %d %H:%M:%S%f %Y)t] [%m:%{severity}i] [pid %P] %E: %M
```

**注：** 提供された名前は、連結されたログフォーマットに由来しています。Apache のエラー ログは、Apache エラー ログフォーマットではなく、上記のフォーマット キーを使用しても記述されます。

次の手順を実行して、エラー ログ フォーマットを定義します。

## 1 エラー ログ フォーマット (httpd.conf) に対応するように Apache を構成します。

```
LogFormat "%h %l %u %{d-%b-%Y:%H:%M:%S}t \"%r\" %a %A %e %k %l %L %m %n %T %v %V %>s %b\n\"%{Referer}i\" \"%{User-Agent}i\" combined"
```

## 2 CLF パーサ構成を定義します。

```
;Parser to parse Apache ERROR logs
[parser|clfparsers_apache_error]
    debug=yes
    base_parser=clf
    format=[%{a %b %d %H:%M:%S%f %Y}t] [%m:%{severity}i] [pid %P] %E: %M
    next_parser=clfparsers_apache_error2

;Parser to parse Apache ERROR logs
[parser|clfparsers_apache_error2]
    debug=yes
    base_parser=clf
    format=[%{a %b %d %H:%M:%S%f %Y}t] [%m:%{severity}i] [pid %P:tid %{thread_id}i] %E: %M
```

ログ エントリ:

```
[Wed May 13 14:37:17.042371 2015] [mpm_winnt:notice] [pid 4488:tid 272] AH00354: Child:
Starting 150 worker threads.
```

CLF パーサはログ エントリ用の次のフィールドを返します (+0400 のタイムゾーンでパーサを使用している場合)。

```
timestamp=2015-05-13T10:37:17.042371
request_method=mpm_winnt
severity=notice
process_id=4488
thread_id=272
error_status=AH00354
log_message=Child: Starting 150 worker threads.
```

ログ エントリ:

```
[Wed May 13 14:37:27.042371 2015] [mpm_winnt:notice] [pid 5288] AH00418: Parent: Created
child process 3480
```

CLF パーサはログ エントリ用の次のフィールドを返します (+0400 のタイムゾーンでパーサを使用している場合)。

```
timestamp=2015-05-13T10:37:27.042371
request_method=mpm_winnt
severity=notice
process_id=5288
error_status=AH00418
log_message=Parent: Created child process 3480
```

## キー/値ペア パーサ

FileLog および WinLog の両方のコレクタに対応するようにキー/値ペア (KVP) パーサを構成できます。

### キー/値ペア (KVP) パーサ

kvp パーサを使用すると、任意のログ メッセージ テキストからすべての key=value の一致を検索して抽出することができます。次に、kvp パーサのフォーマットの例を示します。

```
[parser|kvp_parser]
base_parser=kvp
fields=*
```

たとえば、key-value ログには次の形式があります：scope=local; abstract=false; lazyInit=false; autowireMode=0; dependencyCheck=0;

kvp パーサでは、値を抽出するフィールドを指定する必要があります。たとえば、構成に fields=name,lastname,country という定義がある場合、指定されたキーを持つ値のみが解析され、サーバに送信されます。

スペースやその他の特殊文字を定義するには、キーと値の両方を二重引用符 (" ") で任意に囲むことができます。

キーまたは値に二重引用符を使用した場合は、バックスラッシュ文字 (\) をエスケープ文字として使用できます。バックスラッシュに続く文字はすべて、二重引用符文字やバックスラッシュ文字も含め、リテラルに定義されます。たとえば、“\\” のようにします。

以下の検討事項に注意してください。

- 等号の後にキー/値ペアのキーがなく、VALUE が指定されていない場合、オプションはフリー テキスト フィールドと同様にスキップされます。
- キーは空白にできませんが、値は空白にできます。
- 等号の後に値がない場合はフリー テキストと見なされ、スキップされます。
- 値は二重引用符で囲んで文字列にすることも、空にすることもできます。バックスラッシュを使用して、値の一部の特殊文字をエスケープ処理します。

### KVP パーサのオプション

kvp パーサの構造については、以下の情報に注意してください。

オプション	説明
fields	<p>データの単位として記述された、抽出対象の情報です。たとえば、<code>fields=name,lastname,country</code> など。</p> <p>特定のフィールド名が <code>fields</code> オプションによって定義されている場合、ログから抽出されたフィールド名にあるそれぞれの無効な文字がアンダースコアに置き換えられます。たとえば、<code>fields</code> オプションでフィールド「<code>x-A</code>」および「<code>a*(X+Y)</code>」というフィールドを検索する場合、パーサはこれらのフィールドをログから抽出し、それぞれ「<code>x_a</code>」および「<code>a__x_y</code>」というフィールドに置き換えます。これにより、名前に任意の文字を持つフィールドを抽出することができます。</p> <p><code>fields</code> オプションが「<code>*</code>」として指定されている場合、<code>kvp</code> パーサはフィールド/値のペアを自動的に認識し、「英数字 + アンダースコア」文字（LI サーバでサポートされる）のみを持つフィールドを検索します。その他のすべての無効な文字は、アンダースコアに変換される代わりに削除されます。そのため、パーサが不要な情報を静的フィールドに抽出するのを防ぐことができます。</p>
delimiter	<p>任意。</p> <p>デフォルトの区切り文字はスペース、タブ、改行文字、カンマ、セミコロンです。</p> <p>区切り文字が指定されていない場合、<code>kvp</code> パーサは解析にデフォルトの区切り文字を使用します。</p> <p>デフォルトの区切り文字を特定の区切り文字に変更するには、その区切り文字を二重引用符で囲んで定義する必要があります。たとえば、<code>delimiter = "#^ "</code> のようにします。この定義は、二重引用符で囲まれた各文字が区切り文字として使用されることを意味します。<code>kvp</code> パーサでは任意の文字を区切り文字と見なすことができます。定義には、デフォルトの区切り文字とその他の区切り文字を混在させることができます。たとえば、<code>delimiter = "#^ \t\r\n\s"</code> のステートメントには、区切り文字としてタブ、改行文字、およびスペースが含まれています。これらの文字を使用する場合は、先頭にエスケープ文字を付ける必要があります。たとえば、スペース文字を区切り文字として定義するには、<code>delimiter="\s"</code> のように、スペース文字の前にエスケープ文字（<code>\</code>）を入力します。</p>
field_decoder	<p>ネストされたパーサは JSON 文字列として指定されます。キーはネストされたパーサに適用されるフィールドの名前で、値はそのフィールドに使用されるパーサの名前です。</p> <p>ネストされたパーサは、ベースパーサによるデコードに従って、それぞれ対応するフィールドに適用されます。</p> <p>キー/値のペアがタイムスタンプやカンマ区切りのリストなどの複雑な場合、フィールドデコーダが便利です。</p>
debug =	<p>任意。 <code>debug</code> の値は <code>yes</code> または <code>no</code> になります。パーサに対する <code>debug</code> のデフォルト値は <code>debug=no</code> です。</p> <p>オプションが <code>yes</code> に設定されていると、パーサ取り込みの詳細なログが <code>liagent_&lt;date&gt;.log</code> で確認できます。</p>

## キー値の追加オプション

キー	定義
<code>KVP_MESSAGE = *(MESSAGE_ENTRY [WSPR])</code>	オプションのスペースで区切られたメッセージエントリのリストです。
<code>MESSAGE_ENTRY = KVP / FREE_TEXT</code>	エントリは、キー/値ペアまたはフリーテキストのみです。
<code>KVP = KEY ["=" VALUE]</code>	キー/値ペアです。KEY に続くのが等記号と VALUE ではない場合は、フリーテキストのようにスキップされます。
<code>KEY = BARE_KEY / QUOTED_KEY</code>	
<code>FREE_TEXT = "="</code>	単独の等記号はフリーテキストとして見なされてスキップされます。
<code>BARE_KEY = *1BARE_KEY_CHAR</code>	少なくとも 1 文字です。

キー	定義
BARE_KEY_CHAR = %0x00-08 / %0x10-19 / %0x21-3C / %3E-%FF	等記号、スペース、または TAB を除く任意の文字です。
QUOTED_KEY = 0x22 *1(QUOTED_STRING_CHAR / "\" CHAR) 0x22	二重引用符で囲まれた少なくとも 1 文字です。エスケープ文字としてバックスラッシュが使用されます。
QUOTED_STRING_CHAR = %0x00-21 / %0x23-FF	二重引用符を除く任意の文字です。
VALUE = BARE_VALUE / QUOTED_VALUE	
BARE_VALUE = *BARE_VALUE_CHAR	ゼロ個以上の文字です。
BARE_VALUE_CHAR = %0x00-08 / %0x10-19 / %0x21-FF	スペースまたはタブを除いた任意の文字です。
QUOTED_VALUE = 0x22 *(QUOTED_STRING_CHAR / "\" CHAR) 0x22	二重引用符で囲まれた文字列です。空白にすることもできます。エスケープ文字としてバックスラッシュが使用されます。

## KVP パーサの構成例

必要に応じて、fields=\* を使用してすべてのフィールドを解析することができます。

```
[parser|simple_kvp]
base_parser =kvp
fields=*
```

この例は、フィールド デコーダの指定方法を示します。

```
[parser|mykvp]
debug=no
base_parser=kvp
delimiter="#^|"
fields=*
;OR fields=scope,abstract,lazyInit,autowireMode,dependencyCheck
field_decoder={"field1":"field1_parser1"}

[parser|field1_parser1]
base_parser=clf
format=[%{value1}i]]
field_decoder={"value1":"field_parser2"}
```

次の KVP ログを解析するには：

```
Configuring transport... proto = cfapi server_hostname = LOCALHOST ssl = no port = 9000
reconnect = 30
```

KVP パーサ構成を定義します。

```
[parser|kvp_log_parser]
base_parser=kvp
fields=*
```

KVP パーサは以下のフィールドを返します。

```
proto=cfapi
server_hostname=LOCALHOST
ssl=no
port=9000
reconnect=30
```

### 例：単純な KVP パーサと複雑な KVP パーサの例

単純な KVP パーサの例

```
[filelog|MyLog]
directory=C:\<folder_name>\Parser_logs
include=*.log
parser=my_KVP_parser

[parser|my_KVP_parser]
base_parser=kvp
fields=*
```

複雑な KVP パーサの例

```
[filelog|MyLog]
directory=C:\<folder_name>\Parser_logs
include=*.log
parser=my_KVP_parser

[parser|my_KVP_parser]
base_parser=kvp
fields=*
field_decoder={"field1":" field1_parser1"}

[parser| field1_parser1]
base_parser=clf
format=[%{value1}i]]
field_decoder={"value1":" field1_parser2"}
```

### タイムスタンプ パーサ

timestamp パーサを実行してもフィールドは生成されず、入力が文字列から、1970 年 1 月 1 日（深夜、UTC/GMT）からの経過時間（ミリ秒）で表示された内部タイムスタンプ形式に変換されます。

サポートされている構成オプションは format のみです。たとえば、format=%Y-%m-%d %H:%M:%S など。

CLF パーサとは異なり、timestamp パーサは %A%B%d%H%M%S%Y%z などの時刻指定子間に区切り文字のない時刻を解析します。

timestamp パーサが使用するフォーマット指定子は、以下のとおりです。

```
'%a':    Abbreviated weekday name, for example: Thu
'%A':    Full weekday name, for example: Thursday
'%b':    Abbreviated month name, for example: Aug
'%B':    Full month name, for example: August
'%d':    Day of the month, for example: 23. strftime() expects zero-padded (01-31) digits
          for this specifier but Log Insight agents can parse space-padded and non-padded
          day numbers, too.
'%e':    Day of the month, for example: 13. strftime() expects space-padded ( 1-31) digits
          for this specifier but Log Insight agents can parse zero-padded and non-padded
          day numbers too.
'%f':    Fractional seconds of time, for example: .036 'f' specifier assumes that '.' or ','
          character should exist before fractional seconds and there is no need to mention
          that character in the format. If none of these characters precedes fractional
seconds,
          timestamp wouldn't be parsed.
'%H':    Hour in 24h format (00-23), for example: 14. Zero-padded, space-padded, and non-
padded hours
          are supported.
'%I':    Hour in 12h format (01-12), for example: 02. Zero-padded, space-padded and non-
padded hours
          are supported.
'%m':    Month as a decimal number (01-12), for example: 08. Zero-padded, space-padded
          and non-padded month numbers are supported.
'%M':    Minute (00-59), for example: 55
'%p':    AM or PM designation, for example: PM
'%S':    Second (00-61), for example: 02
'%s':    Total number of seconds from the UNIX epoch start, for example 1457940799
          (represents '2016-03-14T07:33:19' timestamp)
'%Y':    Year, for example: 2001
'%z':    ISO 8601 offset from UTC in timezone (1 minute=1, 1 hour=100) ., for example: +100
```

タイムスタンプ パーサは追加の指定子を受け取りますが、その値は無視され、解析される時刻には影響しません。

```
'%C':    Year divided by 100 and truncated to integer (00-99), for example: 20
'%g':    Week-based year, last two digits (00-99), for example, 01
'%G':    Week-based year, for example, 2001
'%j':    Day of the year (001-366), for example: 235
'%u':    ISO 8601 weekday as number with Monday as 1 (1-7), for example: 4
'%U':    Week number with the first Sunday as the first day of week one (00-53), for example:
33
'%V':    ISO 8601 week number (00-53), for example: 34
'%w':    Weekday as a decimal number with Sunday as 0 (0-6), for example: 4
'%W':    Week number with the first Monday as the first day of week one (00-53), for example:
34
'%y':    Year, last two digits (00-99), for example: 01
```

format パラメータが定義されていないと、Timestamp パーサはデフォルトのフォーマットを使用してタイムスタンプを解析します。



## 自動タイムスタンプ パーサ

自動タイムスタンプ パーサは、タイム スタンプ パーサの形式が定義されていない場合に呼び出されます。また、`field_decoder` で `timestamp` を使用して、タイムスタンプ パーサ定義なしで直接パーサを呼び出すことができます。例：

```
[parser|mycsv]
base_parser=csv
debug=yes
fields=timestamp,action,source_id,dest
field_decoder={"timestamp": "timestamp"}
```

### 例：デフォルト構成のタイムスタンプ パーサ

次の例に、デフォルト構成の `timestamp` パーサを示します。

```
[parser|tsp_parser]
base_parser=timestamp
debug=no
format=%Y-%m-%d %H:%M:%S%f
```

`timestamp` パーサを `CSV` パーサなどの他のパーサと統合するには、以下の構成を指定します。

```
[parser|mycsv]
base_parser=csv
fields=timestamp,action,source_id,dest
field_decoder={"timestamp": "tsp_parser"}
```

この構成が指定されていると、`mycsv` パーサは構成で指定された名前を持つフィールドを抽出し、`timestamp` フィールドの内容に `tsp_parser` を実行します。`tsp_parser` が有効なタイムスタンプを取得すると、サーバはそのタイムスタンプをログ メッセージに使用します。

## 自動ログ パーサ

自動パーサは、行の最初の 200 文字の中で、タイムスタンプを自動的に検出します。自動検出されるタイム スタンプの形式は、`timestamp` パーサのものと同じです。

自動パーサにはオプションがありません。タイムスタンプの自動検出に加えて、ログ エントリでキー/値パーサを実行し、ログにあるすべての既存のキー/値のペアを自動的に検出し、それに従ってフィールドを抽出します。次に例を示します。

```
[filelog|some_logs]
directory=/var/log
include=*
parser=auto
```

他のパーサと同じように、自動パーサのアクションを個別に定義することができます。

```
[filelog|kvplogs]
directory=C:\temp_logs\csv-itbm
include=*.txt
parser=myauto
[parser|myauto]
```

```
base_parser=auto
debug=yes
```

自動パーサで debug を有効にしていると、解析についての追加情報が表示されます。たとえば、どのログで自動パーサが実行されていたか、どのフィールドがログから抽出されたのかについての情報などです。

パーサに対する debug のデフォルト値は debug=no です。

## syslog パーサ

Syslog パーサは message\_decoder と extract\_sd オプションをサポートし、RFC-6587、RFC-5424、および RFC-3164 の 3 つのフォーマットを自動的に検出します。

### message\_decoder オプションの構成

syslog パーサには、すべての共通オプションと message\_decoder オプションを使用できます。デフォルトでは、timestamp と appname フィールドのみが抽出されます。次の例のように liagent.ini ファイルの設定値を設定して、message\_decoder オプションを有効にします。

```
[filelog|data_logs]
directory=D:\Logs
include=*.txt
parser=mysyslog

[parser|mysyslog]
base_parser=syslog
message_decoder=syslog_message_decoder
debug=yes

[parser|syslog_message_decoder]
base_parser=kvp
fields=*
```

### 例： message\_decoder オプションによる解析

次の例は、サンプルのイベントと、message\_decoder オプションを使用するように構成された syslog パーサによってイベントに追加されるフィールドを示しています。

#### ■ サンプルのイベント：

```
2015-09-09 13:38:31.619407 +0400 smith01 john: Fri Dec 5 08:58:26 2014 [pid 26123]
[jsmith.net] status_code=FAIL oper_
ation=LOGIN: Client "176.31.17.46"
```

#### ■ KVP パーサを実行するために message\_decoder オプションが適用されている syslog パーサによって返されます。

```
timestamp=2015-09-09T09:38:31.619407 appname=john status_code=FAIL operation=LOGIN:
```

## 構造化データの解析のための extract\_sd オプションの設定

構造化データを解析するには、次の例のように liagent.ini ファイルの設定値を設定して、extract\_sd オプションを有効にします。

```
[filelog|simple_logs]
directory=/var/log
include=*.txt
parser=syslog_parser

[parser|syslog_parser]
base_parser=syslog
extract_sd=yes
```

### 例：extract\_sd オプションによる解析

次の例は、サンプルのイベントと、extract\_sd オプションを使用するように構成された syslog パーサによってイベントに追加されるフィールドを示しています。

- サンプルのイベント：<165>1 2017-01-24T09:17:15.719Z localhost evntslog - ID47  
[exampleSDID@32473 iut="3" eventSource="Application" eventId="1011"]  
[examplePriority@32473 class="high"] Found entity IPSet, display name dummy ip set 1411
- syslog パーサによって、次のフィールドがイベントに追加されます。

```
timestamp=2017-01-24T09:17:15.719000
pri_facility=20
pri_severity=5
procid="-"
msgid="ID47"
iut="3"
eventsourc="Application"
eventid="1011"
class="high"
appname="evntslog"
```

### パーサによって抽出されるフィールド

パーサは、イベントから次のフィールドを自動的に抽出します。

RFC の分類	pri_facility	pri_severity	timestamp	appname	procid	msgid
RFC 以外			X	X		
RFC-3164	X	X	X	X		
RFC-5424	X	X	X	X	X	X

### syslog パーサのオプション

次の表に、使用可能な syslog オプションを示します。

オプション	説明
message_decoder	イベントのメッセージ本文を解析するために使用される追加のパーサを定義します。これには「auto」などの組み込みのパーサ、あるいは任意のカスタム定義のパーサがあります。
extract_sd	構造化データを解析します。 extract_sd オプションでは [yes] または [no] の値のみがサポートされます。オプションは、デフォルトでは無効です。extract_sd オプションを有効にすると、構造化データからすべてのキー/値ペアが抽出されます。

### 例：RFC-5424 標準の解析

次の例は、syslog インスタンスによって解析される 2 つのイベントを示します。このインスタンスは、コレクタ、サンプルのイベント、および syslog パーサがイベントに追加するフィールドに使用される設定を示します。

#### ■ 設定：

```
[filelog|simple_logs]
directory=/var/log
include=*.txt
parser=syslog
```

#### ■ 監視対象のファイル内で生成されるイベント：

```
<165>1 2017-01-24T09:17:15.719Z router1 mgd 3046 UI_DBASE_LOGOUT_EVENT
[junos@2636.1.1.1.2.18 username=\"regress\"] User 'regress' exiting configuration
mode - Juniper format
```

#### ■ syslog パーサによってイベントに追加されるフィールド。

```
The following fields will be added to the event by Syslog parser:
timestamp=2017-01-24T09:17:15.719000
pri_facility = 20
pri_severity = 5
procid = 3046
msgid = UI_DBASE_LOGOUT_EVENT
appname = mgd
```

### 例：RFC-3164 標準の解析

次の例は、コレクタに使用される設定、サンプルの RFC-3164 イベント、および syslog がイベントに追加するフィールドを示します。

#### ■ 設定：

```
[filelog|simple_logs]
directory=/var/log
include=*.txt
parser=syslog
```

- 監視対象のファイル内で生成される RFC-3164 イベント：

```
<13>2017-01-24T09:17:15.719Z router1 mgd 3046 UI_DBASE_LOGOUT_EVENT User 'regress' exiting
configuration mode - Juniper format
```

- syslog パーサによってイベントに追加されるフィールド。

```
timestamp=2017-01-24T09:17:15.719000
pri_facility=1
pri_severity=5
appname="mgd"
```

## ラベル付きタブ区切り値パーサ

ラベル付きタブ区切り値 (LTSV) 形式は、タブ区切り値 (TSV) のバリエーションです。

LTSV ファイル内の各レコードは 1 行で表されています。各フィールドは <TAB> で区切られていて、ラベルおよび値が含まれています。ラベルおよび値は : で区切られています。LTSV 形式を使用すると、(TSV 形式と同じように) <TAB> で各行を分割して解析し、一意のラベルの付いた任意のフィールドを任意の順序で拡張できます。LTSV の定義および形式の詳細については、<http://ltsv.org/> を参照してください。

### 例：LTSV パーサの構成

### 例：LTSV ログの例

LTSV パーサでは、特定の構成オプションを指定する必要がありません。LTSV パーサを使用するには、構成内で組み込み ltsv パーサの名前を指定します。

```
[parser|myltsv]
base_parser=ltsv
```

LTSV ファイルは、ABNF 形式の LTSV 本番環境に対応するバイト シーケンスである必要があります。

```
ltsv = *(record NL) [record]
record = [field *(TAB field)]
field = label ":" field-value
label = 1*lbyte
field-value = *fbyte

TAB = %x09
NL = [%x0D] %x0A
lbyte = %x30-39 / %x41-5A / %x61-7A / "_" / "." / "-" ;; [0-9A-Za-z_.-]
fbyte = %x01-08 / %x0B / %x0C / %x0E-FF
```

```
host:127.0.0.1<TAB>ident:-<TAB>user:frank<TAB>time:[10/Oct/2000:13:55:36 -0700]<TAB>req:GET /
apache_pb.gif HTTP/1.0<TAB>status:200<TAB>size:2326<TAB>referer:http://www.example.com/
start.html<TAB>ua:Mozilla/4.08 [en] (Win98; I ;Nav)
```

LTSV の構成例では、ログの解析によって次のフィールドが返されます。

```
host=127.0.0.1
ident=-
user=frank
time=[10/Oct/2000:13:55:36 -0700]
req=GET /apache_pb.gif HTTP/1.0
status=200
size=2326
referer=http://www.example.com/start.html
ua=Mozilla/4.08 [en] (Win98; I ;Nav)
```

## 構成のデバッグ

LTSV パーサに追加のデバッグを行うこともできます。デフォルトでは、LTSV デバッグは無効です。LTSV デバッグを有効にするには、`debug=yes` を入力します。

```
[parser|myltsv]
base_parser=ltsv
debug=yes
```

デバッグが有効な場合、LTSV パーサはログからすべての有効なラベルの値を抽出します。LTSV パーサでは、ラベル名を英数字、アンダースコア ('\_')、ピリオド ('.')、およびダッシュ ('-') 文字のみで構成する必要があります。ログ内に無効なラベル名が 1 つ以上存在する場合、解析は失敗します。ラベル名が有効な場合でも、エージェントはフィールド名を確認します。無効な名前が存在する場合は、ラベル名を有効なフィールド名に修正する必要があります。

## filelog セクションからの LTSV パーサの構成

LTSV パーサは `filelog` セクションから直接構成することもできます。

```
[filelog|simple_logs]
directory=/var/log
include=*
parser=ltsv
```

## regex パーサ

regex パーサを使用すると、収集済みデータに対していくつかの正規表現を使用できます。

vRealize Log Insight エージェントは、Perl 構文を持つ C++ Boost ライブラリ正規表現を使用します。regex パーサは、指定されたキャプチャ グループを含む正規表現パターンを指定することで定義できます。例：( ? <field\_1>\d{4}) [-] ( ? <field\_2>\d{4}) [-] ( ? <field\_3>\d{4}) [-] ( ? <field\_4>\d{4})

グループに指定された名前 (field\_1、field\_2、field\_3、および field\_4 など) は、対応する抽出済みフィールドの名前になります。名前には次の要件があります。

- 正規表現パターンで指定された名前は、vRealize Log Insight の有効なフィールド名でなければなりません。
- 名前には英数字とアンダースコア ( \_ ) 文字だけを含めることができます。
- 名前の先頭を数字にすることはできません。

無効な名前を指定すると、構成は失敗します。

## regex パーサのオプション

regex パーサに必要なオプションは、format オプションだけです。

追加のデバッグ情報が必要な場合は、debug オプションを使用できます。

### 構成

regex パーサを作成するには、base\_parser として regex を使用し、format オプションを指定します。

#### 例：regex の構成例

##### 例：Apache ログの解析例

次の例を使用して 1234-5678-9123-4567 を分析できます。

```
[parser|regex_parser]
base_parser=regex
format=(?<tag1>\d{4}) [-] (?<tag2>\d{4}) [-] (?<tag3>\d{4}) [-] (?<tag4>\d{4})
[filelog|some_info]
directory=D:\Logs
include=*.txt
parser=regex_parser
```

結果が表示されます。

```
tag1=1234
tag2=5678
tag3=9123
tag4=4567
```

regex パーサで Apache ログを解析するには、Apache ログ用の特定の regex 形式を指定します。

```
[parser|regex_parser]
base_parser=regex
format=(?<remote_host>.*)(?<remote_log_name>.*)(?<remote_auth_user>.*) \[(?<log_timestamp>.*)\] "(?<request>.*)" (?<status_code>.*)(?<response_size>.*)
```

結果が表示されます。

```
127.0.0.1 - admin [10/Oct/2000:13:55:36 -0700] "GET /apache_pb.gif HTTP/1.0" 200 2326
remote_host=127.0.0.1
remote_log_name=-
remote_auth_user=admin
log_timestamp=10/Oct/2000:13:55:36 -0700
request=GET /apache_pb.gif HTTP/1.0
status_code=200
response_size=2326
```

次のコードは、別の Apache ログの解析例を示します。

```
[parser|regex_parser]
base_parser=regex
format=(?<remote_host>.* (?<remote_log_name>.*)) (?<remote_auth_user>.*) \[(?<log_timestamp>.*)\] "(?<request>.* (?<resource>.*)(?<protocol>.*))" (?<status_code>.*)(?
```

```
<response_size>.*)
127.0.0.1 unknown - [17/Nov/2015:15:17:54 +0400] \"GET /index.php HTTP/1.1\" 200 4868
remote_host=127.0.0.1 unknown
remote_log_name=unknown
remote_auth_user=-
log_timestamp=17/Nov/2015:15:17:54 +0400
request=GET /index.php HTTP/1.1
resource=/index.php
protocol=HTTP/1.1
status_code=200
response_size=4868
```

## パフォーマンスについての考慮事項

regex パーサは、CLF パーサなどの他のパーサよりも多くのリソースを消費します。他のパーサを使用してログを解析できる場合には、より良いパフォーマンスを得るために、regex パーサの代わりにそれらのパーサを使用することを検討してください。

パーサが提供されておらず、regex パーサを使用する場合は、形式をできるだけ簡潔に定義します。次の例は、より良いパフォーマンスを提供する構成を示します。この例は、数値のフィールドを指定します。

```
(?<remote_host>\d+\.\d+\.\d+\.\d+) (?<remote_log_name>.*) (?<remote_auth_user>.*) \[(?<log_timestamp>.*)\] "(?<request>.*)" (?<status_code>\d+) (?<response_size>\d+)
```

## JSON パーサ

JSON ログを選択的に解析するように JSON パーサ構成をカスタマイズできます。

**FileLog** および **WinLog** の両方のコレクタに対応するようにカンマ区切り値 (CSV) パーサを構成できます。

Log Insight エージェント JSON パーサでは、有効な JSON ログのみが解析されます。無効な JSON ログパーサは空の結果を返します。

デフォルトの JSON パーサ構成では、Log Insight エージェントによって、JSON ログからすべてのフィールドが抽出されます。JSON ログが、JSON オブジェクトも含むことができる複雑な JSON オブジェクトとして自分自身を表す場合、パーサはネストされた上位階層の JSON オブジェクトの名前を連結するためにアンダースコア ( \_ ) 文字を使用します。これにより、対応する要素の情報フィールド名が生成されます。JSON ログに配列も含まれている場合、メンバー要素名には配列名が含まれ、その後に配列内の要素のインデックスが続きます。

また、JSON パーサは、**fields** と呼ばれる特定のオプションを提供します。

### JSON パーサの「fields」オプション

**fields** オプションを使用して、構成内のどのフィールドを解析するかを指定できます。このオプションの目的は、JSON ログの選択的な解析を有効にすることです。

**注：** 選択的に解析を行うには、目的の JSON 要素へのパスを指定する必要があります。異なる階層の JSON オブジェクトはドット (.) 文字で区切る必要があります。



次のリストは、必要に応じて JSON ログを選択的に解析するための構成例を示します。

- JSON ログから複数の要素を解析するには、必要な要素を **fields** オプションのパラメータとしてリストし、カンマで区切って指定する必要があります。次の例を参照してください。

```
{ "operation" : { "timestamp" :
    "2018-11-22T15:28:58.094000", "thread_id" : "0x05673", "initiator" : "connector",
    "log_severity" : "info", "log_message" : "Requested connection to the server."},
  "operation_result" : "success" }
```

- **timestamp**、**log\_severity**、および **log\_message** など、最も内側の JSON オブジェクトのみを解析するには、次の例を参照してください。この構成例では、次のようなフィールド結果が生成されます：  
operation\_timestamp = "2018-11-22T15:28:58.094000" and operation\_log\_severity = "info"

```
[parser|json_parser]
base_parser=json
fields=operation.timestamp,operation.log_severity, operation.log_message
```

- JSON オブジェクト全体を解析するには、オブジェクトへのパスと、その後にアスタリスク (\*) 文字を指定します。

```
{ "product_name" : "LI Agent",
  "operation" : { "timestamp" : "2018-11-22T15:28:58.094000", "thread_id" :
    "0x05673", "initiator" : "connector", "log_severity" : "info", "log_message" :
    "Requested connection to the server."}, "operation_result" :
    "success" }
```

- **operation** オブジェクトのみを解析するには、次の構成を使用します：

```
[parser|json_parser]
base_parser=json
fields=operation.*
```

- JSON ログに配列が含まれていて、配列の特定の要素のみを解析する場合は、構成内の配列の要素インデックスを次の例のように使用します：

```
{
  "Records": [{
    "object":{
      "key": "/events/mykey",
      "size": 502,
      "eTag": "091820398091823",
      "sequencer": "1123123"
    }
  },
  {
    "object":{
      "key": "/events/user_key",
      "size": 128,
      "eTag": "09182039000001",
      "sequencer": "1123231"
    }
  },
  ]
}
```

```
{
  "object": {
    "key": "/events/admin_key",
    "size": 1024,
    "eTag": "09182039547241",
    "sequencer": "1123213"
  }
}
```

- 同じログから **key** および **size** 要素のみを解析するには、次の構成を使用して次のフィールドを生成します：

```
records0_object_key="/events/mykey"
```

```
records0_object_size=502
```

```
records2_object_key="/events/admin_key"
```

```
records2_object_size=1024
```

```
[parser|json_parser]
base_parser=json
fields = Records0.object.key Records0.object.size, Records2.object.key,
Records2.object.size
```

- すべての配列要素の **key** フィールドを解析するには、次の構成を使用します：

```
[parser|json_parser]
base_parser=json
fields=Records.#.object.key
```

- すべてのフィールドを解析するには、フィールド オプションにアスタリスク (\*) 文字を指定します。この構成は、デフォルトの JSON パーサ構成と同じです。

```
[parser|json_parser]
base_parser=json
fields=*
```

# vRealize Log Insight エージェントのアンインストール

# 5

vRealize Log Insight Agent をアンインストールする必要がある場合には、インストールしたエージェント パッケージに該当する指示に従います。

この章には、次のトピックが含まれています。

- [Log Insight Windows Agent のアンインストール](#)
- [Log Insight Linux Agent RPM パッケージのアンインストール](#)
- [Log Insight Linux Agent DEB パッケージのアンインストール](#)
- [Log Insight Linux Agent bin パッケージのアンインストール](#)
- [Log Insight Linux Agent bin パッケージの手動アンインストール](#)

## Log Insight Windows Agent のアンインストール

Windows のコントロール パネルのプログラムと機能の画面から Log Insight Windows Agent をアンインストールできます。

### 前提条件

vRealize Log Insight Windows エージェントをインストールした Windows マシンにログインし、サービス マネージャを起動して vRealize Log Insight エージェント サービスがインストールされていることを確認します。

### 手順

- 1 [コントロール パネル] - [プログラムと機能] に移動します。
- 2 VMware vRealize Log Insight Windows Agent を選択し、[アンインストール] をクリックします。

### 結果

アンインストーラが VMware vRealize Log Insight Windows Agent サービスを停止し、ファイルをシステムから削除します。

## Log Insight Linux Agent RPM パッケージのアンインストール

Log Insight Linux Agent RPM パッケージをアンインストールできます。

**前提条件**

- root としてログインするか、または `sudo` を使用してコンソール コマンドを実行します。
- Log Insight Linux Agent がインストールされている Linux マシンにログインし、ターミナル コンソールを開き、`pgrep liagent` を実行して、VMware Log Insight Linux Agent がインストールされていて実行中であることを確認します。

**手順**

- ◆ 次のコマンドを、*VERSION* および *BUILD\_NUMBER* をインストールされているエージェントのバージョンおよびビルド番号に置き換えて実行します。

```
rpm -e VMware-Log-Insight-Agent-VERSION-BUILD_NUMBER
```

**結果**

アンインストーラによって VMware Log Insight Linux Agent のデーモンが停止され、そのログ以外のすべてのファイルがシステムから削除されます。

## Log Insight Linux Agent DEB パッケージのアンインストール

Log Insight Linux Agent DEB パッケージをアンインストールできます。

**前提条件**

- root としてログインするか、または `sudo` を使用してコンソール コマンドを実行します。
- Log Insight Linux Agent がインストールされている Linux マシンにログインし、ターミナル コンソールを開き、`pgrep liagent` を実行して、VMware Log Insight Linux Agent がインストールされていて実行中であることを確認します。

**手順**

- ◆ 次のコマンドを実行します。

```
dpkg -P vmware-log-insight-agent
```

**結果**

アンインストーラによって VMware Log Insight Linux Agent のデーモンが停止され、そのログ以外のすべてのファイルがシステムから削除されます。

## Log Insight Linux Agent bin パッケージのアンインストール

vRealize Log Insight スクリプトで Log Insight Linux Agent .bin パッケージをアンインストールすることができます。

**前提条件**

- root としてログインするか、または `sudo` を使用してコンソール コマンドを実行します。

- Log Insight Linux Agent がインストールされている Linux マシンにログインし、ターミナル コンソールを開き、`pgrep liagent` を実行して、VMware vRealize Log Insight Linux Agent がインストールされて実行中であることを確認します。

#### 手順

- 1 シェル プロンプトで、スクリプトを起動するために、次のコマンドを入力します。

```
loginsight-agent-uninstall
```

- 2 次のコマンドから返されるエラー コードが 0 であるかチェックすることで、アンインストールが正常に完了したことを確認することができます。

```
echo $?
```

## Log Insight Linux Agent bin パッケージの手動アンインストール

アンインストール スクリプトを使用しないことを選択した場合に、Log Insight Linux Agent .bin パッケージを手動でアンインストールできます。

#### 前提条件

#### Log Insight Linux Agent bin パッケージの手動アンインストール

- root としてログインするか、または `sudo` を使用してコンソール コマンドを実行します。
- Log Insight Linux Agent がインストールされている Linux マシンにログインし、ターミナル コンソールを開き、`pgrep liagent` を実行して、VMware vRealize Log Insight Linux Agent がインストールされて実行中であることを確認します。

#### 手順

- 1 次のコマンドを実行することによって、Log Insight Linux Agent daemon を停止します。

```
sudo service liagentd stop、または以前の Linux ディストリビューションの場合は sudo /  
sbin/service liagentd stop
```

- 2 手動で次の Log Insight Linux Agent ファイルを削除します。

- `/usr/lib/loginsight-agent` - デーモン バイナリおよびライセンス ファイルのディレクトリ。
- `/usr/bin/loginsight-agent-support` - Log Insight Linux Agent のサポート バンドルを生成するために使用。
- `/var/lib/loginsight-agent` - 構成ファイルおよびデータベース ストレージ ディレクトリ。
- `/var/log/loginsight-agent` - Log Insight Linux Agent のログ ディレクトリ。
- `/var/run/liagent/liagent.pid` - Log Insight Linux Agent PID ファイル。ファイルが自動的に削除されない場合、手動で削除します。
- `/etc/init.d/liagentd` - Log Insight Linux Agent デーモン用のスクリプト ディレクトリ。
- `/usr/lib/systemd/system/liagentd.service`

# vRealize Log Insight エージェントの トラブルシューティング

# 6

既知のトラブルシューティング情報は、vRealize Log Insight エージェントの運用に関連する問題の診断と解決に役立てることができます。

この章には、次のトピックが含まれています。

- Log Insight Windows Agent のサポート バンドルの作成
- Log Insight Linux Agent のサポート バンドルの作成
- Log Insight Agents のログの詳細レベルの定義
- 管理 UI に Log Insight Agents が表示されない
- vRealize Log Insight Agent でイベントを送信しない
- Log Insight Windows Agent に対する送信例外ルールの追加
- Windows ファイアウォールでの Log Insight Windows Agent からの送信接続の許可
- Log Insight Windows Agent の一括展開に失敗しました
- Log Insight Agents による自己署名証明書の拒否
- vRealize Log Insight サーバによる暗号化されていないトラフィックの接続の拒否

## Log Insight Windows Agent のサポート バンドルの作成

問題が発生したため Log Insight Windows Agent が予測どおりに動作しない場合は、ログおよび構成ファイルのコピーを VMware サポート サービスに送信できます。

### 手順

- 1 Log Insight Windows Agent をインストールしたターゲット マシンにログインします。
- 2 Windows の [スタート] ボタンをクリックし、[ VMware ] - [ Log Insight エージェント - サポート バンドルの収集 ] をクリックします。
- 3 (オプション) ショートカットがない場合、Log Insight Windows Agent のインストール ディレクトリに移動し、`loginsight-agent-support.exe` をダブルクリックします。

---

**注：** デフォルトのインストール ディレクトリは `C:\Program Files (x86)\VMware\Log Insight Agent` です。

---

**結果**

バンドルが生成され、.zip ファイルとしてマイ ドキュメント に保存されます。

**次のステップ**

必要に応じて VMware サポート サービスにサポート バンドルを転送します。

## Log Insight Linux Agent のサポート バンドルの作成

問題が発生したため Log Insight Linux Agent が予測どおりに動作しない場合は、ログおよび構成ファイルのコピーを VMware サポート サービスに送信できます。

**手順**

- 1 Log Insight Linux Agent をインストールしたターゲット マシンにログインします。
- 2 次のコマンドを実行します。

```
/usr/lib/loginsight-agent/bin/loginsight-agent-support
```

**結果**

バンドルが生成され、.zip ファイルとして現在のディレクトリに保存されます。

**次のステップ**

必要に応じて VMware サポート サービスにサポート バンドルを転送します。

## Log Insight Agents のログの詳細レベルの定義

vRealize Log Insight Agent の構成ファイルを編集してログ レベルを変更することができます。

**前提条件**

Log Insight Linux Agent の場合：

- root としてログインするか、または sudo を使用してコンソール コマンドを実行します。
- Log Insight Linux Agent がインストールされている Linux マシンにログインし、コンソールを開き、pgrep liagent を実行して、VMware vRealize Log Insight Linux Agent がインストールされて実行中であることを確認します。

Log Insight Windows Agent の場合：

- vRealize Log InsightWindows エージェントをインストールした Windows マシンにログインし、サービス マネージャを起動して vRealize Log Insight エージェント サービスがインストールされていることを確認します。

## 手順

- 1 `liagent.ini` ファイルを含むフォルダに移動します。

オペレーティング システム	パス
Linux	<code>/var/lib/loginsight-agent/</code>
Windows	<code>%ProgramData%\VMware\Log Insight Agent</code>

- 2 任意のテキスト エディタで `liagent.ini` を開きます。
- 3 `liagent.ini` ファイルの `[logging]` セクションでログ デバッグ レベルを変更します。

**注：** デバッグ レベルが高いほど、vRealize Log Insight Agent への影響も高くなります。デフォルト値（推奨値）は 0 です。デバッグ レベル 1 では詳しい情報が提供されるため、ほとんどの問題のトラブルシューティングの際には、このレベルをお勧めします。デバッグ レベル 2 ではさらに詳しい情報が提供されます。レベル 1 とレベル 2 は、VMware サポートから要求された場合にのみ使用してください。

```
[logging]
; The level of debug messages to enable: 0..2
debug_level=1
```

- 4 `liagent.ini` ファイルを保存して閉じます。

## 結果

ログ デバッグ レベルが変更されます。

## 管理 UI に Log Insight Agents が表示されない

Log Insight Agents インスタンスに関する情報が、管理 UI の [エージェント] ページに表示されません。

## 問題

Log Insight Agents をインストールした後、管理 UI の [エージェント] ページに Log Insight Agents が表示されません。

## 原因

最も一般的な原因は、ネットワーク接続の問題または `liagent.ini` 内の Log Insight Agents の構成が間違っていることです。

## 解決方法

- ◆ Log Insight Agents がインストールされている Windows または Linux システムが vRealize Log Insight サーバに接続されていることを確認します。
- ◆ Log Insight Agents が cfapi プロトコルを使用していることを確認します。

Syslog プロトコルを使用している場合は、UI に Log Insight Windows Agents が表示されません。



- ◆ 次のディレクトリにある Log Insight Agents ログ ファイルの内容を表示します。

- Windows - %ProgramData%\VMware\Log Insight Agent\log
- Linux - /var/log/loginsight-agent/

次のフレーズを含むログ メッセージを探します: 「構成転送エラー: ホスト名を解決できません」 および 「リゾルバーが失敗しました。そのようなホストは不明です」

- ◆ liagent.ini にターゲット vRealize Log Insight サーバの正しい構成が含まれていることを確認します。  
「[ターゲット vRealize Log InsightServer の設定](#)」 および 「[エージェントの宛先の指定](#)」 を参照してください。

## vRealize Log Insight Agent でイベントを送信しない

間違った構成では、vRealize Log Insight が vRealize Log Insight サーバにイベントを転送できません。フラット ファイル収集チャンネルが正しく構成されていない場合、「Invalid settings were obtained for channel 'CHANNEL\_NAME'. Channel 'CHANNEL\_NAME' will stay dormant until properly configured.」のようなメッセージが表示される場合があります。

### 問題

vRealize Log Insight Agent インスタンスが [管理] - [エージェント] ページに表示されますが、[インタラクティブ分析] ページに vRealize Log Insight Agent のホスト名からのイベントが表示されません。フラット ファイル収集チャンネルが正しく構成されていません。

### 原因

間違った構成では、vRealize Log Insight が vRealize Log Insight サーバにイベントを転送できません。

### 解決方法

- ◆ 有効な収集チャンネルを定義します。フラット ファイル収集チャンネルが正しく構成されていることを確認します。  
[4 章 vRealize Log Insight エージェントの構成](#) を参照してください。
- ◆ vRealize Log Insight Windows Agent に対して次の操作を実行してください。
  - Windows チャンネルが有効になっている場合、%ProgramData%\VMware\Log Insight Agent\log にある vRealize Log Insight Windows Agent ログ ファイルの内容を表示します。  
Subscribed to channel CHANNEL\_NAME というフレーズを含む、チャンネル構成に関連するログ メッセージを探します。通常使用されるチャンネルは、Application、System、および Security です。
  - チャンネルが正しく構成されていない場合、Could not subscribe to channel CHANNEL\_NAME events. Error Code: 15007. The specified channel could not be found. Check channel configuration のようなログ メッセージが表示される場合があります。15007 以外のエラー コード番号が表示される場合があります。
  - フラット ファイル収集チャンネルが正しく構成されていない場合、Invalid settings were obtained for channel 'CHANNEL\_NAME'. Channel 'CHANNEL\_NAME' will stay dormant until properly configured のようなメッセージが表示される場合があります。

- ◆ vRealize Log Insight Windows Agent および vRealize Log Insight Linux Agent の両方に対して次の操作を実行します。

- ◆ フラット ファイル収集チャネルが構成されていない場合、Cannot find section 'filelog' in the configuration.The flat file log collector will stay dormant until properly configured のようなメッセージが表示される場合があります。

vRealize Log Insight Agent ログ ファイルの内容は、次のディレクトリにあります。

- Windows - %ProgramData%\VMware\Log Insight Agent\log
- Linux - /var/log/loginsight-agent/

#### 次のステップ

vRealize Log Insight Agent の構成の詳細については、[Log Insight Windows Agent の構成](#) および [Log Insight Linux Agent の構成](#)を参照してください。

## Log Insight Windows Agent に対する送信例外ルールの追加

Windows ファイアウォールで Log Insight Windows Agent をブロック解除するための例外ルールを定義します。

この手順は Windows Server 2008 R2 以降および Windows 7 以降に適用されます。

#### 前提条件

- 管理者アカウントまたは管理権限があるアカウントを持っていることを確認します。

#### 手順

- 1 [開始] - [実行] を選択します。
- 2 wf.msc と入力して [OK] をクリックします。
- 3 左のペインにある [送信ルール] を右クリックし、[新規ルール] をクリックします。
- 4 [カスタム] を選択し、ウィザードに従って次のオプションを設定します。

オプション	説明
プログラム	liwinsvc.exe
サービス	LogInsightAgentService
プロトコルおよびポート	cfapi の場合は TCP 9000、syslog の場合は 514

- 5 [このルールを適用するプロファイルの指定] ページで、適切なネットワーク タイプを選択します。
  - ドメイン
  - パブリック

## ■ プライベート

**注：** すべてのネットワーク タイプを選択して、ネットワーク タイプに関係なく例外ルールをアクティブにすることもできます。

### 次のステップ

Log Insight Windows Agent ログ ディレクトリの %ProgramData%\VMware\Log Insight Agent\log に移動して、最新のログ ファイルを開きます。最近のイベントに、「構成送信エラー:ホスト名を解決できませんでした」および「リゾルバが失敗しました。そのようなホストは見つかりません」というメッセージが含まれていた場合は、Log Insight Windows Agent サービスと Windows マシンを再起動してください。

**注：** Log Insight Windows Agent サービスからサーバへの再接続には 5 分ほどかかります。

## Windows ファイアウォールでの Log Insight Windows Agent からの送信接続の許可

Windows ファイアウォールの設定を構成して、Log Insight Windows Agent の vRealize Log Insight サーバへの送信接続を許可します。

Log Insight Windows Agent サービスをインストールして開始した後、Windows ドメインまたはローカル ファイアウォールがターゲット vRealize Log Insight サーバへの接続を制限する場合があります。

この手順は Windows Server 2008 R2 以降および Windows 7 以降に適用されます。

### 前提条件

- 管理者アカウントまたは管理権限があるアカウントを持っていることを確認します。

### 手順

- 1 [開始] - [実行] を選択します。
- 2 wf.msc と入力して [OK] をクリックします。
- 3 [アクション] ペインで、[プロパティ] をクリックします。
- 4 [ドメイン プロファイル] タブで、[送信接続] ドロップダウン メニューから [許可 (既定)] を選択します。

コンピューターがドメインに接続されていない場合は、コンピューターが接続されているネットワークのタイプに応じて [プライベート プロファイル] または [パブリック プロファイル] を選択できます。

- 5 [OK] をクリックします。

### 次のステップ

Windows ファイアウォールで、Log Insight Windows Agent のブロック解除例外ルールを定義します。 [Log Insight Windows Agent に対する送信例外ルールの追加](#)を参照してください。

## Log Insight Windows Agent の一括展開に失敗しました

ターゲット マシン上で Log Insight Windows Agent の一括展開に失敗しました。

### 問題

グループ ポリシー オブジェクトを使用して Windows ドメイン マシンに一括展開を実行した後、Log Insight Windows Agent はローカル サービスとしてのインストールに失敗しました。

### 原因

グループ ポリシーの設定が Log Insight Windows Agent の正しいインストールを妨げた可能性があります。

### 解決方法

- 1 グループ ポリシーオブジェクト (GPO) 設定を編集し、Log Insight Windows Agent を再展開します。
  - a GPO を右クリックし、[編集] をクリックして [コンピューターの構成] - [ポリシー] - [管理用テンプレート] - [システム] - [ログオン] に移動します。
  - b [コンピューターの起動およびログオンで常にネットワークを待つ] ポリシーを有効にします。
  - c [コンピューターの構成] - [ポリシー] - [管理用テンプレート] - [システム] - [グループ ポリシー] に移動します。
  - d [スタートアップ ポリシー処理時の待機時間] を有効にし、[待機する時間 (秒) :] を 120 に設定します。
- 2 ターゲット マシン上で `gpupdate /force /boot` コマンドを実行します。

## Log Insight Agents による自己署名証明書の拒否

Log Insight Agents は自己署名証明書を拒否します。

### 問題

vRealize Log Insight エージェントが自己署名証明書を拒否し、サーバとの接続を確立できません。

---

**注：** エージェントとの接続で問題が発生した場合は、エージェントのデバッグ レベルを 1 に変更することで、詳細ログを生成して確認できます。詳細については、[Log Insight Agents のログの詳細レベルの定義](#)を参照してください。

---

### 原因

エージェント ログに表示されるメッセージには、具体的な理由が記載されます。

メッセージ	原因
ピアの自己署名証明書を拒否します。パブリック キーが以前に保存された証明書のキーと一致しません。	<ul style="list-style-type: none"> <li>■ これは、vRealize Log Insight の証明書が置換された場合に起こることがあります。</li> <li>■ これは、HA 対応のクラスタ内環境が、vRealize Log Insight ノード上の異なる自己署名証明書で構成されている場合に発生する場合があります。</li> </ul>
ピアの自己署名証明書を拒否します。信頼される CA によって署名された、以前受信した証明書を使用してください。	CA 署名付き証明書がエージェント側に格納されています。

#### 解決方法

- ◆ ターゲットのホスト名が、信頼される vRealize Log Insight インスタンスであることを確認してから、以前の証明書を vRealize Log Insight Agent の cert ディレクトリから手動で削除します。
  - Log Insight Windows Agent 用の証明書は、C:\ProgramData\VMware\Log Insight Agent\cert にあります。
  - Log Insight Linux Agent 用の証明書は、/var/lib/loginsight-agent/cert にあります。

**注：**一部のプラットフォームでは、信頼される証明書の保存先として、標準以外のパスを使用している場合があります。Log Insight Agents には、`ssl_ca_path=<fullpath>` 構成パラメータを設定することで、信頼される証明書ストアへのパスを構成するオプションがあります。<fullpath> は、信頼されるルート証明書のバンドル ファイルのパスに置き換えてください。[Log Insight Agent の SSL パラメータの構成](#)を参照してください。

## vRealize Log Insight サーバによる暗号化されていないトラフィックの接続の拒否

vRealize Log Insight サーバは、暗号化されていないトラフィックを送信しようとする Log Insight Agents との接続を拒否します。

非 SSL 接続を受け入れるように vRealize Log Insight サーバを構成するか、SSL cfapi プロトコル接続を介してデータを送信するように Log Insight Agents を構成できます。

#### 問題

cfapi を使用して暗号化されていないトラフィックを送信しようとする、vRealize Log Insight サーバによってその接続が拒否されます。次のエラー メッセージがエージェントのログに表示されます: 403 Forbidden または 403 Only SSL connections are allowed.

#### 原因

vRealize Log Insight は SSL 接続のみを受け入れるように構成されていますが、Log Insight Agents は非 SSL 接続を使用するように構成されています。

## 解決方法

- 1 非 SSL 接続を受け入れるように vRealize Log Insight サーバを構成します。
  - a [管理] タブに移動します。
  - b [構成] で [SSL] をクリックします。
  - c [API サーバの SSL] ヘッダーで、[SSL 接続が必要] の選択を解除します。
  - d [保存] をクリックします。
- 2 SSL Cfapi プロトコル接続を介してデータを送信するように vRealize Log Insight エージェントを構成します。

- a liagent.ini ファイルを含むフォルダに移動します。

オペレーティング システム	パス
Linux	/var/lib/loginsight-agent/
Windows	%ProgramData%\VMware\Log Insight Agent

- b 任意のテキスト エディタで liagent.ini を開きます。
- c liagent.ini ファイルの [server] セクションにある ssl キーの値を yes に変更し、プロトコルを cfapi に変更します。

```
proto=cfapi
ssl=yes
```

- d liagent.ini ファイルを保存して閉じます。