

使用 vRealize Log Insight 代理程式

2022 年 5 月 24 日
vRealize Log Insight 8.1

您可以在 VMware 網站上找到最新的技術文件，網址如下：

<https://docs.vmware.com/tw/>

VMware, Inc.
3401 Hillview Ave.
Palo Alto, CA 94304
www.vmware.com

Copyright © 2022 VMware, Inc. 保留所有權利。 [版權與商標資訊](#)。

目錄

使用 vRealize Log Insight 代理程式 5

1 vRealize Log Insight 代理程式概觀 6

2 記錄檔輪替配置的類型 8

3 安裝或升級 vRealize Log Insight 代理程式 9

下載代理程式安裝檔案 10

安裝 Windows 代理程式 11

使用安裝精靈安裝或更新 vRealize Log Insight Windows 代理程式 11

從命令列安裝或更新 vRealize Log Insight Windows 代理程式 12

將 Log Insight Windows Agent 部署到多台機器 13

安裝或更新 vRealize Log Insight Linux 代理程式 RPM 套件 17

安裝或更新 vRealize Log Insight Linux 代理程式 DEB 套件 18

針對 Debian Linux 自訂代理程式安裝 19

安裝 Log Insight Linux Agent 二進位套件 21

Linux 上 vRealize Log Insight 代理程式安裝的命令列選項 22

vRealize Log Insight 代理程式的自動更新 23

停用或啟用個別代理程式的自動更新 24

4 設定 vRealize Log Insight 代理程式 25

設定 Log Insight Windows Agent 26

Log Insight Windows Agent 的預設組態 26

從 Windows 事件通道收集事件 28

從記錄檔收集事件 32

將事件轉送到 Log Insight Windows Agent 37

設定 Log Insight Linux Agent 37

vRealize Log Insight Linux 代理程式的預設組態 37

從記錄檔收集事件 39

篩選來自 vRealize Log Insight 代理程式的事件 46

從 vRealize Log Insight 代理程式轉送資訊 47

設定目標 vRealize Log Insight 伺服器 48

指定代理程式的目標 51

vRealize Log Insight 代理程式的集中式組態 54

組態合併範例 55

針對代理程式組態使用一般值 56

剖析記錄 58

設定記錄剖析器 58

5 解除安裝 vRealize Log Insight 代理程式 86

- 解除安裝 Log Insight Windows Agent 86
- 解除安裝 Log Insight Linux 代理程式 RPM 套件 86
- 解除安裝 Log Insight Linux 代理程式 DEB 套件 87
- 解除安裝 Log Insight Linux 代理程式 bin 套件 87
- 手動解除安裝 Log Insight Linux 代理程式 bin 套件 88

6 對 vRealize Log Insight 代理程式進行疑難排解 89

- 為 Log Insight Windows Agent 建立支援服務包 89
- 為 Log Insight Linux Agent 建立支援服務包 90
- 定義 Log Insight Agents 中的記錄詳細資料層級 90
- 管理 UI 不顯示 Log Insight Agents 91
- vRealize Log Insight 代理程式不傳送事件 92
- 為 Log Insight Windows Agent 新增輸出例外狀況規則 93
- 在 Windows 防火牆中允許來自 Log Insight Windows Agent 的輸出連線 93
- Log Insight Windows Agent 的大量部署不成功 94
- Log Insight Agents 拒絕自我簽署的憑證 95
- vRealize Log Insight 伺服器拒絕非加密流量的連線 95

使用 vRealize Log Insight 代理程式

《使用 vRealize Log Insight 代理程式》說明如何安裝和設定 vRealize™ Log Insight™ Windows 和 Linux 代理程式。其也包含疑難排解提示。

此資訊適用於要安裝、設定或疑難排解 Log Insight Agents 的任何人。該資訊是針對熟悉虛擬機器技術和資料中心作業且富有經驗的 Windows 或 Linux 系統管理員而撰寫。

有關如何使用 vRealize Log Insight 伺服器為代理程式建立組態類別的相關資訊，請參閱《管理 vRealize Log Insight》。

vRealize Log Insight 代理程式概觀

1

vRealize Log Insight 代理程式會從記錄檔收集事件，並將其轉送至 vRealize Log Insight 伺服器或任何第三方 syslog 目的地。

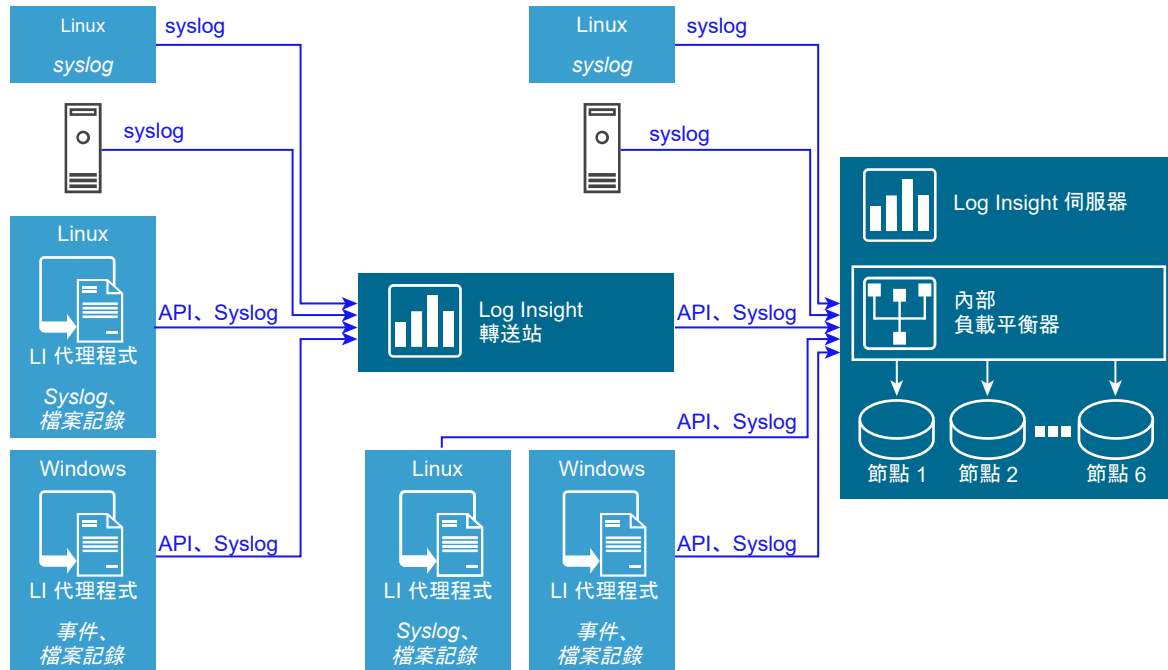
代理程式支援 Syslog 和 vRealize Log Insight 擷取 API (cfapi 通訊協定)，且可搭配 Linux 或 Windows 平台使用。您可以透過 Web 介面、使用伺服器和用戶端上的 liagent.ini 檔案，或是在安裝過程中設定代理程式。

代理程式包含下列功能：

- 單一或群組部署
- 自動升級
- 對記錄訊息執行剖析並擷取結構化資料。您可以為 FileLog 和 (或) WinLog 收集器設定剖析器。
- 多行訊息的支援
- 多個記錄輪替配置的原生支援
- 包含用戶端壓縮、加密，並且能夠將中繼資料新增至事件的大量擷取 API

vRealize Log Insight 伺服器支援集中式組態管理，以及代理程式群組的建立和管理。

下圖顯示代理程式部署組態的元素。



vRealize Log Insight 轉送站是一個主要負責將事件轉送至遠端目的地的專用 vRealize Log Insight 伺服器執行個體。通常，用作轉送站的伺服器執行個體不會用於查詢。轉送站會使用內部負載平衡器，且其結構會類似於 vRealize Log Insight 伺服器。

代理程式會寫入其本身的作業記錄。針對 Windows，這些記錄位於 `C:\ProgramData\VMware\Log Insight Agent\logs` 目錄。針對 Linux，作業記錄的路徑為 `/var/log/loginsight-agent/liagent_*.log`。當代理程式重新啟動或檔案的大小達到 10 MB 時，記錄檔會進行輪替。輪替會保持 50 MB 的檔案合併限制。您無法使用 vRealize Log Insight 代理程式本身來收集代理程式記錄。

代理程式會用於即時記錄收集。使用 vRealize Log Insight Importer 來匯入歷史記錄收集，包括支援服務包。

針對 Windows 和 Linux 作業系統會提供個別的安裝下載項目。

在 Windows 系統上，此代理程式會以 Windows 服務的形式執行，並在安裝後立即啟動。代理程式會監控應用程式記錄檔和 Windows 事件通道，以及用於收集相關 Windows 系統事件的集區。收集的事件會轉送到 vRealize Log Insight 伺服器或第三方 syslog 目的地。

在 Linux 系統上，此代理程式會以精靈的形式執行，並在安裝後立即啟動。vRealize Log Insight Linux 代理程式會從 Linux 機器上的記錄檔中收集事件，並將其轉送至 vRealize Log Insight 伺服器或 syslog 目的地。目前有 Debian、Red Hat 和 Linux 二進位安裝套件可供使用。

vRealize Log Insight 代理程式支援的記錄輪替配置

2

vRealize Log Insight 代理程式支援數個記錄輪替配置。

記錄輪替可確保記錄檔不會無限成長。目前有數個記錄輪替配置可供使用，每個配置都是為一組特定的使用案例而設計的。vRealize Log Insight 包含下列配置的原生支援。

表 2-1. vRealize Log Insight 代理程式所支援的記錄輪替配置

記錄輪替配置	說明
<code>create-new</code>	達到大小或時間限制後，會建立新的記錄檔。記錄器程序會停止寫入到目前的記錄檔，並將記錄輸出導向至新建立的檔案。任何現有檔案皆無法以任何其他方式重新命名或更改。
<code>rename-recreate</code>	在達到大小或時間限制時， <code>logrotate</code> 之類的外部公用程式會將記錄檔重新命名。然後，記錄器程序會以以前的名稱建立記錄檔。
<code>copy-truncate</code>	在達到大小或時間限制時， <code>logrotate</code> 之類的外部公用程式會複製記錄檔。記錄器程序會將複製的檔案重新命名，並截斷原始的檔案，使得其大小變成 0。記錄器程序可以繼續將記錄寫入至原始檔案。

安裝或升級 vRealize Log Insight 代理程式

3

您可以在 Windows 或 Linux 機器上安裝或升級 vRealize Log Insight 代理程式，包括具有第三方記錄管理系統的這類機器。

代理程式會收集事件，並將其轉送至 vRealize Log Insight 伺服器。安裝期間，您可以指定伺服器、連接埠和通訊協定等設定的參數，或選擇保留預設設定。

您可以使用安裝時使用的相同方法來升級代理程式，也可以使用自動升級。自動升級會在您部署新版的 vRealize Log Insight 時，將升級內容散佈至代理程式。如需詳細資訊，請參閱 [vRealize Log Insight 代理程式的自動更新](#)。升級不適用於 Linux 二進位套件。

硬體支援

若要安裝和執行 vRealize Log Insight 代理程式，您的硬體必須支援主機/機器為了支援 x86 和 x86_64 架構以及 MMX、SSE、SSE2 和 SSE3 指令集所需的最低參數。

平台支援

作業系統	處理器架構
Windows 7、Windows 8、Windows 8.1 和 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 和 Windows Server 2019	x86_64
RHEL 5、RHEL 6、RHEL 7 和 RHEL 8	x86_64、x86_32
SuSE Enterprise Linux (SLES) 11 SP3 和 SLES 12 SP1	x86_64
Ubuntu 14.04 LTS、Ubuntu 16.04 LTS 和 Ubuntu 18.04	x86_64
VMware Photon (第 1 版，修訂 2 版)、第 2 版和第 3 版	x86_64

Linux 注意事項

如果您沒有可用根權限之使用者實作 Log Insight Linux 代理程式的預設安裝，預設組態可能會產生資料收集的相關問題。代理程式不會記錄訂閱通道不成功，以及收集中檔案沒有讀取權限的警告。訊息 `Inaccessible log file ... will try later` 會重複地新增至記錄。您可以標註造成此問題的預設組態，或變更使用者權限。

如果您使用 rpm 或 DEB 套件來安裝 Linux 代理程式，則會在套件安裝過程中安裝名為 `liagentd` 的 `init.d` 指令碼。bin 套件會新增指令碼，但不會進行登錄。您可以手動登錄指令碼。

您可以執行 `(/sbin/)service liagentd status` 命令來確認安裝是否成功。

本章節討論下列主題：

- [下載代理程式安裝檔案](#)
- [安裝 Windows 代理程式](#)
- [安裝或更新 vRealize Log Insight Linux 代理程式 RPM 套件](#)
- [安裝或更新 vRealize Log Insight Linux 代理程式 DEB 套件](#)
- [針對 Debian Linux 自訂代理程式安裝](#)
- [安裝 Log Insight Linux Agent 二進位套件](#)
- [Linux 上 vRealize Log Insight 代理程式安裝的命令列選項](#)
- [vRealize Log Insight 代理程式的自動更新](#)

下載代理程式安裝檔案

設定 vRealize Log Insight 代理程式的第一步是為您的平台下載代理程式安裝套件。

從 vRealize Log Insight 伺服器代理程式頁面下載的所有套件，都會包含附加至套件名稱的目的地主機名稱。`server.hostname` 會在 MSI、RPM 及 DEB 代理程式的初始安裝期間加以套用。如果組態檔中有主機名稱存在，或您是透過主機名稱參數來執行套件，則會忽略內嵌式伺服器主機名稱。

程序

- 1 導覽至 vRealize Log Insight Web 使用者介面的**管理**索引標籤。
- 2 在 [管理] 區段中，按一下**代理程式**。
- 3 捲動至畫面底部，然後按一下**下載 Log Insight 代理程式**。
- 4 從快顯功能表中選取安裝套件，並按一下**儲存**，以下載該套件。

選項	說明
Windows MSI	Windows 平台 (32 位元/64 位元) 的安裝套件
Linux RPM	Linux Red Hat、openSUSE (32 位元/64 位元) 或 VMware Photon Platform 的安裝套件

選項	說明
Linux DEB	Linux Debian 平台 (32 位元/64 位元) 的安裝套件
Linux BIN	Linux (32 位元/64 位元) 的自動安裝套件。不需要套件管理系統。

後續步驟

使用下載的檔案來部署 vRealize Log Insight 代理程式。

安裝 Windows 代理程式

您可以透過安裝精靈或命令列將代理程式安裝在單一機器上，或使用指令碼部署代理程式的多個執行個體。

升級 Windows 代理程式

您可以使用任何可用來安裝的方法套用升級檔案，以升級 Windows 代理程式。您也可以選擇使用自動升級功能，在背景中升級您的代理程式。

使用安裝精靈安裝或更新 vRealize Log Insight Windows 代理程式

您可以使用安裝精靈在單一機器上安裝或升級 Windows 代理程式。

必要條件

- 確認您有 vRealize Log Insight Windows 代理程式 .msi 檔案的複本。請參閱[下載代理程式安裝檔案](#)。
- 確認您擁有在 Windows 電腦上執行安裝和啟動服務的權限。

程序

- 1 登入用來安裝 vRealize Log Insight Windows 代理程式的 Windows 機器。
- 2 變更為具有 vRealize Log Insight Windows 代理程式 .msi 檔案的目錄。
- 3 按兩下 vRealize Log Insight Windows 代理程式 .msi 檔案，接受授權合約的條款，然後按下一步。
- 4 輸入 vRealize Log Insight 伺服器的 IP 位址或主機名稱，然後按一下**安裝**。
精靈會以「本機系統」服務帳戶安裝或更新做為自動 Windows 服務的 vRealize Log Insight Windows 代理程式。
- 5 按一下**完成**。

後續步驟

透過編輯 `liagent.ini` 檔案來設定 vRealize Log Insight Windows 代理程式。請參閱[設定 Log Insight Windows Agent](#)。

從命令列安裝或更新 vRealize Log Insight Windows 代理程式

您可以從命令列安裝或更新 Windows 代理程式。

您可以使用預設值或指定服務帳戶，並使用命令列參數來指定伺服器、連接埠和通訊協定資訊。關於 MSI 命令列選項，請參閱 Microsoft Developer Network (MSDN) Library 網站，並搜尋 MSI 命令列選項。

必要條件

- 確認您有 vRealize Log Insight Windows 代理程式 .msi 檔案的複本。請參閱[下載代理程式安裝檔案](#)。
- 確認您擁有在 Windows 電腦上執行安裝和啟動服務的權限。
- 如果使用無訊息安裝選項 `/quiet` 或 `/qn`，請確認您以管理員身分執行安裝。如果您不是管理員並執行無訊息安裝，安裝不會提示需要管理員權限並失敗。使用記錄選項和參數 `/l*xv* file_name`，以達到診斷目的。

程序

- 1 登入要安裝或更新 vRealize Log Insight Windows 代理程式的 Windows 機器。
- 2 開啟**命令提示字元**視窗。
- 3 變更為具有 vRealize Log Insight Windows 代理程式 .msi 檔案的目錄。
- 4 透過下列格式的命令，使用預設值進行安裝或更新。請以您的版本和組建編號取代 *version-build_number*。

`/quiet` 選項會以無訊息的方式執行命令，而 `/l*xv` 選項會在現行目錄中建立記錄檔。

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

- 5 (選擇性) 指定用於執行 vRealize Log Insight Windows 代理程式服務的使用者服務帳戶。

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

備註 SERVICEACCOUNT 參數中提供的帳戶被授與以**服務方式**登入權限以及對 `%ProgramData%\VMware\Log Insight Agent` 目錄的完整寫入權限。如果提供的帳戶不存在，系統會加以建立。使用者名稱不可超過 20 個字元。如果您未指定 SERVICEACCOUNT 參數，則 vRealize Log Insight Windows 代理程式服務會透過 LocalSystem 服務帳戶安裝或更新。

6 (選擇性) 您可以視需要指定下列命令列選項的值。

選項	說明
SERVERHOST=hostname	vRealize Log Insight 虛擬應用裝置的 IP 位址或主機名稱。
SERVERPROTO=protocol	代理程式用於傳送事件至 vRealize Log Insight 伺服器的通訊協定。可能的值為 cfapi 和 syslog。 預設值為 cfapi。
SERVERPORT=portnumber	代理程式用來將事件傳送至 vRealize Log Insight 或第三方伺服器的通訊連接埠。 依預設，代理程式會根據針對 SSL 和通訊協定而設定的選項使用適當的連接埠。請參閱下列清單中提供的預設連接埠值。只有在連接埠選項與這些預設值不同時，才需要指定此選項。 <ul style="list-style-type: none"> ■ 已啟用 SSL 的 cfapi : 9543 ■ 已停用 SSL 的 cfapi : 9000 ■ 已啟用 SSL 的 Syslog : 6514 ■ 已停用 SSL 的 Syslog : 514
SERVICEACCOUNT=account-name	用來執行 Log Insight Windows Agent 服務的使用者服務帳戶。 備註 SERVICEACCOUNT 參數中提供的帳戶必須擁有以服務方式登入的權限以及對 %ProgramData%\VMware\Log Insight Agent 目錄的寫入權限，以便安裝程式可以正確執行。如果您未指定 SERVICEACCOUNT 參數，則 vRealize Log Insight Windows 代理程式服務會安裝在 LocalSystem 服務帳戶下。
SERVICEPASSWORD=password	使用者服務帳戶的密碼。
AUTOUPDATE={yes no}	啟用或停用代理程式的自動更新。您也必須從 vRealize Log Insight 伺服器啟用自動更新，如此才能完整啟用自動更新。預設值為 [是]。
LIAGENT_SSL={yes no}	啟用安全連線。如果啟用 SSL，代理程式會使用 TLS 1.2 通訊協定與伺服器通訊。預設值為 [是]。

結果

該命令會以 Windows 服務的形式來安裝或更新 vRealize Log Insight Windows 代理程式。vRealize Log Insight Windows 代理程式服務會在您啟動 Windows 機器時啟動。

後續步驟

確認您設定的命令列參數會在 liagent.ini 檔案中正確套用。請參閱[設定 Log Insight Windows Agent](#)。

將 Log Insight Windows Agent 部署到多台機器

您可以在 Windows 網域中的目標電腦上執行 Log Insight Windows Agent 的大型部署。

程序

1 建立可部署多個 vRealize Log Insight Windows 代理程式的轉換檔

在部署多個代理程式的過程中，您必須建立為部署指定組態參數的轉換檔。在安裝代理程式時，系統會將 .mst 轉換檔套用至 .msi 檔案。參數包含代理程式的目的地伺服器和通訊協定、連接埠，以及用於安裝和啟動 Log Insight 代理程式服務的使用者帳戶。

2 部署 vRealize Log Insight Windows 代理程式的多個執行個體

您可以在 Windows 網域的目標電腦上部署 vRealize Log Insight Windows 代理程式的多個執行個體。

建立可部署多個 vRealize Log Insight Windows 代理程式的轉換檔

在部署多個代理程式的過程中，您必須建立為部署指定組態參數的轉換檔。在安裝代理程式時，系統會將 .mst 轉換檔套用至 .msi 檔案。參數包含代理程式的目的地伺服器 and 通訊協定、連接埠，以及用於安裝和啟動 Log Insight 代理程式服務的使用者帳戶。

參數包含代理程式的目的地伺服器 and 通訊協定、連接埠，以及用於安裝和啟動 Log Insight 代理程式服務的使用者帳戶。

必要條件

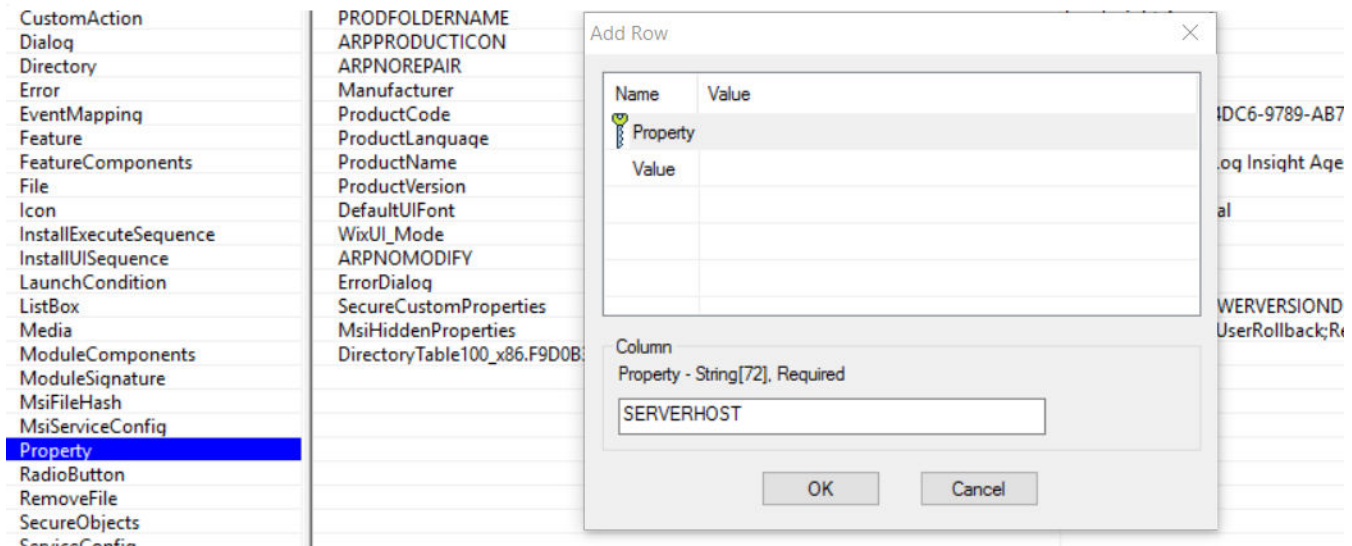
- 確認您有 vRealize Log Insight Windows .msi 檔案的複本。請參閱[下載代理程式安裝檔案](#)。
- 下載並安裝 Orca 資料庫編輯器。請參閱 <http://support.microsoft.com/kb/255905>。

程序

- 1 在 Orca 編輯器中開啟 vRealize Log Insight Windows 代理程式 .msi 檔案，然後選取**轉換 > 新增轉換**。

2 編輯內容資料表，並新增自訂安裝或升級所需的參數和值。

圖 3-1. 內容資料表



- 按一下 **內容**。
- 從 **表格** 下拉式功能表中，選取 **新增資料列**。
- 在 [新增資料列] 對話方塊中輸入內容名稱和值。

參數顯示於下表中。

參數	說明
SERVERHOST	vRealize Log Insight 虛擬應用裝置的 IP 位址或主機名稱。 預設值為 <code>loginsight</code> 。
SERVERPROTO	代理程式用於傳送事件至 vRealize Log Insight 伺服器的通訊協定。可能的值為 <code>cfapi</code> 和 <code>syslog</code> 。 預設值為 <code>cfapi</code> 。
SERVERPORT	代理程式用來將事件傳送至 vRealize Log Insight 或第三方伺服器的通訊連接埠。 依預設，代理程式會根據針對 SSL 和通訊協定而設定的選項使用適當的連接埠。請參閱下列清單中提供的預設連接埠值。只有在連接埠選項與這些預設值不同時，才需要指定此選項。 <ul style="list-style-type: none"> ■ 已啟用 SSL 的 <code>cfapi</code> : 9543 ■ 已停用 SSL 的 <code>cfapi</code> : 9000 ■ 已啟用 SSL 的 <code>Syslog</code> : 6514 ■ 已停用 SSL 的 <code>Syslog</code> : 514
SERVICEACCOUNT	用來執行 Log Insight Windows Agent 服務的使用者服務帳戶。 備註 SERVICEACCOUNT 參數中提供的帳戶必須擁有以 服務方式 登入的權限以及對 <code>%ProgramData%\VMware\Log Insight Agent</code> 目錄的寫入權限，以便安裝程式可以正確執行。如果您未指定 SERVICEACCOUNT 參數，則 vRealize Log Insight Windows 代理程式服務會安裝在 <code>LocalSystem</code> 服務帳戶下。

參數	說明
SERVICEPASSWORD	使用者服務帳戶的密碼。
AUTOUPDATE	啟用或停用代理程式的自動更新。您也必須從 vRealize Log Insight 伺服器啟用自動更新，如此才能完整啟用自動更新。預設值為 [是]。
LIAGENT_SSL	啟用安全連線。如果啟用 SSL，代理程式會使用 TLS 1.2 通訊協定與伺服器通訊。預設值為 [是]。

3 選取 **轉換 > 產生轉換**，然後儲存 .mst 轉換檔。

後續步驟

使用 .msi 和 .mst 檔案來部署 vRealize Log Insight Windows 代理程式。

部署 vRealize Log Insight Windows 代理程式的多個執行個體

您可以在 Windows 網域的目標電腦上部署 vRealize Log Insight Windows 代理程式的多個執行個體。

如需有關為何需要將用戶端機器重新開機兩次的詳細資訊，請參閱 <http://support.microsoft.com/kb/305293>。

必要條件

- 確保您擁有網域控制站的管理員帳戶或具有管理權限的帳戶。
- 確認您有 vRealize Log Insight Windows 代理程式 .msi 檔案的複本。請參閱 [下載代理程式安裝檔案](#)。
- 自行熟悉 <http://support.microsoft.com/kb/887405> 和 <http://support.microsoft.com/kb/816102> 中所述的程序。

程序

- 1 以管理員或具有管理權限之使用者的身分登入網域控制站。
- 2 建立發佈點並將 vRealize Log Insight Windows 代理程式 .msi 檔案複製到發佈點。
- 3 開啟 [群組原則管理主控台]，然後建立群組原則物件以部署 vRealize Log Insight Windows 代理程式 .msi 檔案。
- 4 編輯適用於軟體部署的群組原則物件並指派套件。
- 5 (選擇性) 如果在部署前已產生 .mst 檔案，請在 **GPO 內容** 視窗的 **修改索引標籤** 上選取 .mst 組態檔，並使用進階方法編輯群組原則物件來部署 .msi 套件。
- 6 (選擇性) 升級 vRealize Log Insight Windows 代理程式。
 - a 將升級 .msi 檔案複製到發佈點。
 - b 按一下 [群組原則物件內容] 視窗上的 **升級索引標籤**。
 - c 將最初安裝版本的 .msi 檔案新增到套件，此套件將升級區段。
- 7 將 vRealize Log Insight Windows 代理程式部署到包含網域使用者的特定安全群組。

- 關閉網域控制站上的所有 [群組原則管理主控台] 和 [群組原則管理編輯器] 視窗，然後重新開啟用戶端機器。

如果已啟用 [快速登入最佳化]，請將用戶端機器重新開機兩次。

- 確認 vRealize Log Insight Windows 代理程式已做為本機服務安裝在用戶端機器上。

如果您已設定 `SERVICEACCOUNT` 和 `SERVICEPASSWORD` 參數以使用 `.mst` 檔案部署 vRealize Log Insight Windows 代理程式的多個執行個體，請確認已使用您指定的使用者帳戶在用戶端機器上安裝 vRealize Log Insight Windows 代理程式。

後續步驟

若 vRealize Log Insight Windows 代理程式的多個執行個體不成功，請參閱 [Log Insight Windows Agent 的大量部署不成功](#)。

安裝或更新 vRealize Log Insight Linux 代理程式 RPM 套件

您可以用根使用者或非根使用者的身分安裝或更新 vRealize Log Insight Linux 代理程式，並且可以在安裝期間設定組態參數。安裝後，您可以確認已安裝的版本。

必要條件

- 閱讀 [Linux 上 vRealize Log Insight 代理程式安裝的命令列選項](#) 中安裝預設值以及如何進行變更的相關資訊。
- 以根使用者身分登入，或使用 `sudo` 執行主控台命令。
- vRealize Log Insight Linux 代理程式需要存取 `syslog` 和網路服務才能運作。安裝 vRealize Log Insight Linux 代理程式，並在執行層級 3 和 5 上加以執行。如果想要 vRealize Log Insight Linux 代理程式在其他執行層級下運作，請正確設定系統。

程序

- 您可以從主控台安裝或升級代理程式。
 - 若要以預設組態設定安裝 vRealize Log Insight Linux 代理程式，請開啟主控台並執行下列命令。

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

- 若要升級代理程式而不變更目前的組態設定，請開啟主控台並執行下列命令。

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

- (選擇性) 您可以在更新期間覆寫安裝的預設組態值或目前的組態值。若要這麼做，請將選項指定為安裝或升級命令的一部分。

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

- 3 (選擇性) 透過執行下列命令來確認已安裝的版本。

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

範例：Linux 代理程式安裝和更新範例

- 下列命令會安裝 Linux RPM 發行版的 vRealize Log Insight 代理程式。此命令會將代理程式安裝在個別的伺服器上、指派非預設連接埠號碼，並建立 vRealize Log Insight 代理程式使用者。

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

- 下列命令會使用指定的 rpm 檔案來更新代理程式。目前的代理程式組態則不會變更。

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

安裝或更新 vRealize Log Insight Linux 代理程式 DEB 套件

您可以從命令列或透過 debconf 資料庫來安裝或更新 vRealize Log Insight Linux 代理程式 DEB (Debian) 套件。安裝後，您可以確認已安裝的版本。

必要條件

- 在 [Linux 上 vRealize Log Insight 代理程式安裝的命令列選項](#) 中閱讀安裝預設值以及如何修改預設值的相關資訊。
- 以**根使用者**身分登入，或使用 `sudo` 執行主控台命令。
- 確認 vRealize Log Insight Linux 代理程式有權存取 Syslog 和網路服務，以便正常運作。依預設，vRealize Log Insight Linux 代理程式在執行層級 2、3、4 及 5 上執行，並在執行層級 0、1 和 6 上停止。
- 如需詳細資訊和範例，請參閱[針對 Debian Linux 自訂代理程式安裝](#)。

程序

- 1 若要安裝或更新 vRealize Log Insight Linux 代理程式，請開啟主控台，並執行 `dpkg -i package_name` 命令。

`package_name` 由首碼 `vmware-log-insight-agent-` 和您下載之版本的版本組件編號組成。下列命令格式會以預設值安裝套件。

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

- 2 (選擇性) 透過執行下列命令來確認已安裝的版本：

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

範例

- 從命令列設定連線通訊協定。

若要覆寫預設的連線通訊協定，請使用 SERVERPROTO 變數，如下列範例所示：

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

針對 Debian Linux 自訂代理程式安裝

您可以透過使用命令選項覆寫用於安裝的目前組態值，或設定 debconf 資料庫以自訂您的安裝。

從命令列自訂

若要從命令列設定您的安裝，請使用下列格式的命令：

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

如需完整選項清單，請參閱 [Linux 上 vRealize Log Insight 代理程式安裝的命令列選項](#)。

下列範例顯示從命令列完成的部分一般組態。

- 指定目標 vRealize Log Insight 伺服器。
- 若要在安裝期間設定目標伺服器，請執行 `sudo` 命令，並使用 vRealize Log Insight 伺服器的 IP 位址或主機名稱取代 hostname，如下列範例所示：

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

除非您在安裝期間啟用了 `--force-confold` 旗標，否則只要更新至較新版本，系統就會提示您保留或取代 liagent.ini 組態檔。將顯示下列系統訊息：

```
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] ?
```

若要保留現有組態，請使用 `[default=N]`。從命令列傳遞的其他參數仍然適用。

- 設定連線通訊協定。

若要覆寫預設的連線通訊協定，請使用 SERVERPROTO 變數，如下列範例所示：

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

■ 設定連線連接埠。

若要覆寫預設的連線連接埠，請將 SERVERPORT 變數的值提供給安裝程式，如下列範例所示：

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

■ 以非根使用者的身分執行代理程式。

若要以非根使用者身分執行 vRealize Log Insight Linux 代理程式，請執行 `sudo` 命令。

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

若指定的使用者不存在，vRealize Log Insight Linux 代理程式可在安裝期間建立此使用者帳戶。解除安裝後，不會刪除已建立的帳戶。如果您使用 `LIAGENTUSER=non_root_user` 參數安裝 Linux 代理程式，並嘗試使用 `LIAGENTUSER=non_root_user2` 參數進行升級，將會發生衝突。由於 `non_root_user2` 使用者沒有 `non_root_user` 使用者的權限，因此會出現警告。

適用於 debconf 資料庫的 DEB 套件自訂選項

代理程式 DEB 套件也可以透過 debconf 資料庫來設定。下表顯示支援的 debconf 選項和對應的 vRealize Log Insight 代理程式 DEB 安裝程式選項：

命令列選項	Debconf 選項	說明
<code>SERVERHOST=hostname</code>	<code>vmware-log-insight-agent/serverhost</code>	vRealize Log Insight 虛擬應用裝置的 IP 位址或主機名稱。 預設值為 <code>loginsight</code> 。
<code>SERVERPROTO={cfapi syslog}</code>	<code>vmware-log-insight-agent/serverproto</code>	代理程式用於傳送事件至 vRealize Log Insight 伺服器的通訊協定。可能的值為 <code>cfapi</code> 和 <code>syslog</code> 。 預設值為 <code>cfapi</code> 。
<code>SERVERPORT=portnumber</code>	<code>vmware-log-insight-agent/serverport</code>	代理程式用來將事件傳送至 vRealize Log Insight 或第三方伺服器的通訊連接埠。依預設，代理程式會根據針對 SSL 和通訊協定而設定的選項使用適當的連接埠。請參閱下列清單中提供的預設連接埠值。只有在連接埠選項與這些預設值不同時，才需要指定此選項。 <ul style="list-style-type: none"> ■ 已啟用 SSL 的 <code>cfapi</code>：9543 ■ 已停用 SSL 的 <code>cfapi</code>：9000 ■ 已啟用 SSL 的 <code>Syslog</code>：6514 ■ 已停用 SSL 的 <code>Syslog</code>：514
<code>LIAGENT_INITSYSTEM={init systemd}</code>	<code>log-insight-agent/init_system</code>	在安裝期間，代理程式會自動偵測您要安裝代理程式所在機器之 init 系統的類型。藉由使用此選項來指定系統類型的值，即可覆寫此行為。支援的 init 系統有兩種類型： <code>init</code> 和 <code>systemd</code> 。

命令列選項	Debconf 選項	說明
LIAGENT_AUTOUPDATE={ yes no}	vmware-log-insight-agent/auto_update	啟用或停用代理程式的自動更新。您也必須從 vRealize Log Insight 伺服器啟用自動更新，如此才能完整啟用自動更新。預設值為 [是]。 Linux BIN 套件不支援自動更新。
LI_AGENT_RUNSERVICES	vmware-log-insight-agent/init_system	依預設，服務 liagentd (代理程式) 和 liupdaterd (更新程式) 在安裝之後會立即啟動。您可以將 LIAGENT_RUNSERVICES debconf 參數設為 no ，以防止它們啟動。預設值為 [是]。接受的值只有 yes 和 no ；1 或 0 不是支援的值。
LIAGENT_SSL	vmware-log-insight-agent/ssl	C
LIAGENTUSER= <i>user-account-name</i>	vmware-log-insight-agent/liagentuser	指定用來執行代理程式的帳戶。如果使用者不存在，則安裝程式會將其建立為一般使用者。若指定的使用者帳戶不存在，則 vRealize Log Insight Linux 代理程式會在安裝期間建立使用者帳戶。解除安裝後，不會刪除已建立的帳戶。 依預設代理程式會安裝為使用根使用者身分來執行。 如果您使用 LIAGENTUSER= <i>non_root_user</i> 參數安裝代理程式，並嘗試使用 LIAGENTUSER= <i>non_root_user2</i> 進行升級，則會發生衝突。由於 <i>non_root_user2</i> 使用者沒有使用者 <i>non_root_user</i> 的權限，因此會出現警告。 解除安裝期間不會移除所建立的使用者。您可以手動進行移除。此參數僅用於代理程式服務。更新程式服務一律會以根使用者身分來執行。

安裝 Log Insight Linux Agent 二進位套件

安裝二進位套件包括將 .bin 檔案變更為可執行檔，然後再安裝代理程式。

升級 .bin 套件並不受到正式支援。如果您使用 .bin 套件安裝現有的 Log Insight Linux Agent，請為 /var/lib/loginsight-agent 目錄中的 liagent.ini 檔案建立備份複本以保留本機組態。建立備份複本後，手動解除安裝 Log Insight Linux Agent。請參閱[手動解除安裝 Log Insight Linux 代理程式 bin 套件](#)。

如果您使用 .bin 套件安裝 Linux 代理程式，將在套件安裝過程中安裝名為 liagentd 的 init.d 指令碼，但套件不會登錄此指令碼。您可以手動登錄指令碼。

您可以透過執行 (/sbin/)service liagentd status 命令確認安裝是否成功。

必要條件

- 下載 Log Insight Linux Agent.bin 套件，並將其複製到目標 Linux 機器。
- 確認 Log Insight Linux Agent 具有存取 syslog 和網路服務的權限。
- 閱讀預設組態值以及安裝時的變更方式的相關資訊。請參閱[Linux 上 vRealize Log Insight 代理程式安裝的命令列選項](#)。

程序

- 1 開啟主控台，然後執行 `chmod` 命令，將 `.bin` 檔案變更為可執行檔。

將 *filename-version* 取代為適當的版本。

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

- 2 從命令提示字元執行 `./filename-version.bin` 命令，以安裝代理程式。

將 *filename-version* 取代為適當的版本。

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

- 3 (選擇性) 若要在安裝期間設定目標 vRealize Log Insight 伺服器，請執行 `sudo SERVERHOST=hostname ./filename-version.bin` 命令。

使用 vRealize Log Insight 伺服器的 IP 位址或主機名稱取代 *hostname*。

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

- 4 (選擇性) 若要覆寫預設的連線通訊協定，請使用 `SERVERPROTO` 變數，如下列範例所示：

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

- 5 (選擇性) 若要覆寫預設的連線連接埠，請將 `SERVERPORT` 變數的值提供給安裝程式，如下列範例所示：

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

- 6 (選擇性) 若要以非根使用者身分執行 Log Insight Linux Agent，請執行 `sudo` 命令。

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

如果指定的使用者不存在，Log Insight Linux Agent 會在安裝期間建立使用者帳戶。解除安裝後，不會刪除已建立的帳戶。如果您使用 `LIAGENTUSER=non_root_user` 參數來安裝 Log Insight Linux Agent，並嘗試使用 `LIAGENTUSER=non_root_user2` 參數進行升級，將發生衝突並顯示警告，因為 `non_root_user2` 使用者沒有 `non_root_user` 使用者的權限。

Linux 上 vRealize Log Insight 代理程式安裝的命令列選項

從命令列安裝 vRealize Log Insight 代理程式時，您可以在安裝期間納入用來設定部署的選項。這些選項會對應於 `liagent.ini` 檔案中的設定。

下列選項可用來在安裝期間設定 Linux 系統上執行的 vRealize Log Insight 代理程式。

選項	說明
<code>SERVERHOST=hostname</code>	vRealize Log Insight 虛擬應用裝置的 IP 位址或主機名稱。 預設值為 <code>loginsight</code> 。
<code>SERVERPROTO={cfapi syslog}</code>	代理程式用於傳送事件至 vRealize Log Insight 伺服器的通訊協定。可能的值為 <code>cfapi</code> 和 <code>syslog</code> 。 預設值為 <code>cfapi</code> 。
<code>SERVERPORT=portnumber</code>	代理程式用來將事件傳送至 vRealize Log Insight 或第三方伺服器的通訊連接埠。依預設，代理程式會根據針對 SSL 和通訊協定而設定的選項使用適當的連接埠。請參閱下列清單中提供的預設連接埠值。只有在連接埠選項與這些預設值不同時，才需要指定此選項。 <ul style="list-style-type: none"> ■ 已啟用 SSL 的 <code>cfapi</code> : 9543 ■ 已停用 SSL 的 <code>cfapi</code> : 9000 ■ 已啟用 SSL 的 <code>Syslog</code> : 6514 ■ 已停用 SSL 的 <code>Syslog</code> : 514
<code>LIAGENT_INITSYSTEM={init systemd}</code>	在安裝期間，代理程式會自動偵測您要安裝代理程式所在機器之 <code>init</code> 系統的類型。藉由使用此選項來指定系統類型的值，即可覆寫此行為。支援的 <code>init</code> 系統有兩種類型： <code>init</code> 和 <code>systemd</code> 。
<code>LIAGENT_AUTOUPDATE={yes no}</code>	啟用或停用代理程式的自動更新。您也必須從 vRealize Log Insight 伺服器啟用自動更新，如此才能完整啟用自動更新。預設值為 [是]。 Linux BIN 套件不支援自動更新。
<code>LIAGENT_SSL={yes no}</code>	啟用安全連線。如果啟用 SSL，代理程式會使用 TLS 1.2 通訊協定與伺服器通訊。預設值為 [是]。
<code>LIAGENTUSER=user-account-name</code>	指定用來執行代理程式的帳戶。如果使用者不存在，則安裝程式會將其建立為一般使用者。若指定的使用者帳戶不存在，則 vRealize Log Insight Linux 代理程式會在安裝期間建立使用者帳戶。解除安裝後，不會刪除已建立的帳戶。 依預設代理程式會安裝為使用根使用者身分來執行。 如果您使用 <code>LIAGENTUSER=non_root_user</code> 參數進行安裝，並嘗試使用 <code>LIAGENTUSER=non_root_user2</code> 進行升級，則會發生衝突。由於 <code>non_root_user2</code> 使用者沒有使用者 <code>non_root_user</code> 的權限，因此會出現警告 解除安裝期間不會移除所建立的使用者。您可以手動進行移除。此參數僅用於代理程式服務。更新程式服務一律會以根使用者身分來執行。

vRealize Log Insight 代理程式的自動更新

vRealize Log Insight 代理程式的自動更新功能，可讓作用中的代理程式根據來自 vRealize Log Insight 伺服器的代理程式安裝套件進行檢查、下載及自動更新。

您可以從伺服器啟用所有代理程式的自動更新，或從用戶端啟用個別代理程式執行個體的自動更新。代理程式必須處於作用中狀態，且必須是 4.3 版或更新版本。

Linux BIN 套件不支援自動更新。

停用或啟用個別代理程式的自動更新

您可以編輯個別代理程式的用戶端組態檔，以啟用或停用該代理程式的自動更新。

依預設會從用戶端啟用代理程式的自動更新。

必要條件

代理程式必須是 4.3 版或更新版本。

程序

- 1 在編輯器中開啟本機 `liagent.ini` 檔案。
- 2 找出 `[update]` 區段。

其外觀會類似於下列範例。

```
[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 若要停用自動更新，取消註解 `auto_update=yes` 並將其變更為 `auto_update=no`。

備註 代理程式的自動更新依預設為啟用。因此，`auto_update` 的預設值為「是」，即使已加上註解。

- 4 儲存並關閉 `liagent.ini` 檔案。

設定 vRealize Log Insight 代理程式

4

部署代理程式後，可將其設定為向所選的 vRealize Log Insight 伺服器傳送事件、指定通訊協定以及設定其他參數。

根據需要使用這些指示設定您的代理程式以滿足需求。

- **設定 Log Insight Windows Agent**

安裝完 Log Insight Windows Agent 之後您可以加以設定。編輯 `liagent.ini` 檔案以將 Log Insight Windows Agent 設定為將事件傳送至 vRealize Log Insight、設定通訊協定和連接埠、新增 Windows 事件通道，以及設定一般檔案記錄收集。該檔案位於 `%ProgramData%\VMware\Log Insight Agent` 目錄。

- **設定 Log Insight Linux Agent**

安裝完 Log Insight Linux Agent 之後您可以加以設定。

- **篩選來自 vRealize Log Insight 代理程式的事件**

您可以利用本機 `liagent.ini` 檔案 `[server|<dest_id>]` 區段中的篩選選項，提供代理程式傳送到目的地的資訊。

- **從 vRealize Log Insight 代理程式轉送資訊**

您可以將代理程式收集的事件轉送至最多三個目的地。目的地可包含 vRealize Log Insight 伺服器或轉送站，或第三方記錄管理解決方案。

- **vRealize Log Insight 代理程式的集中式組態**

您可以設定多個 vRealize Log Insight 代理程式。

- **針對代理程式組態使用一般值**

您可以使用為 Windows 或 Linux 代理程式的每個代理程式組態區段套用的一般參數值，來覆寫代理程式組態檔的預設值。

- **剖析記錄**

代理程式端記錄剖析器會從原始記錄擷取結構化資料，然後傳遞至 vRealize Log Insight 伺服器。使用記錄剖析器，vRealize Log Insight 可從中分析記錄、擷取資訊並在伺服器上顯示這些結果。可針對 Windows 和 Linux vRealize Log Insight 代理程式設定記錄剖析器。

設定 Log Insight Windows Agent

安裝完 Log Insight Windows Agent 之後您可以加以設定。編輯 `liagent.ini` 檔案以將 Log Insight Windows Agent 設定為將事件傳送至 vRealize Log Insight、設定通訊協定和連接埠、新增 Windows 事件通道，以及設定一般檔案記錄收集。該檔案位於 `%ProgramData%\VMware\Log Insight Agent` 目錄。

Log Insight Windows Agent 的預設組態

安裝完成後，`liagent.ini` 檔案將包含為 Log Insight Windows Agent 預先設定的預設設定。

Log Insight Windows Agent `liagent.ini` 預設組態

如果您使用非 ASCII 名稱和值，將組態儲存為 UTF-8。

如果您使用中央組態，最終的組態即為此使用伺服器設定所加入的檔案以形成 `liagent-effective.ini` 檔案。

您會發現從伺服器的代理程式頁面進行設定會更有效。

```
; 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

```

參數	預設值	說明
hostname	LOGINSIGHT	vRealize Log Insight 虛擬應用裝置的 IP 位址或主機名稱。 預設值為 loginsight 。
central_config	yes	啟用或停用此代理程式的集中式組態。停用集中式組態時，代理程式會忽略 vRealize Log Insight 伺服器所提供的組態。接受的值為 yes、no、1 或 0。預設值為 yes。

參數	預設值	說明
proto	cfapi	代理程式用於傳送事件至 vRealize Log Insight 伺服器的通訊協定。可能的值為 cfapi 和 syslog。 預設值為 cfapi。
port	9543、9000、6514 和 514	代理程式用來將事件傳送至 vRealize Log Insight 或第三方伺服器的通訊連接埠。依預設，代理程式會根據針對 SSL 和通訊協定而設定的選項使用適當的連接埠。請參閱下列清單中提供的預設連接埠值。只有在連接埠選項與這些預設值不同時，才需要指定此選項。 <ul style="list-style-type: none"> ■ 已啟用 SSL 的 cfapi : 9543 ■ 已停用 SSL 的 cfapi : 9000 ■ 已啟用 SSL 的 Syslog : 6514 ■ 已停用 SSL 的 Syslog : 514
ssl	yes	啟用或停用 SSL。預設值為 yes。 當 ssl 設為「yes」時，如果您沒有為連接埠設定一個值，連接埠會自動選取為 9543。
max_disk_buffer	200	Log Insight Windows Agent 用於緩衝事件及其自身記錄檔的最大磁碟空間 (以 MB 為單位)。 達到指定的 max_disk_buffer 後，代理程式將開始捨棄新的傳入事件。
debug_level	0	定義記錄詳細資料層級。請參閱 定義 Log Insight Agents 中的記錄詳細資料層級 。
channel	應用程式, 安全性, 系統	依預設，會對應用程式、安全性和系統 Windows 事件記錄通道加上註解；Log Insight Windows Agent 不會收集來自這些通道的記錄。 請參閱 從 Windows 事件通道收集事件 。
raw_syslog	no	針對使用 Syslog 通訊協定的代理程式，允許代理程式收集和傳送原始 Syslog 事件。預設值為 [否]，表示收集的事件會使用使用者指定的 Syslog 屬性來轉換。啟用此選項以收集事件而不進行任何 Syslog 轉換。接受的值為 [是]/[1] 和 [否]/[0]。

從 Windows 事件通道收集事件

您可以將 Windows 事件通道新增至 Log Insight Windows Agent 組態。Log Insight Windows Agent 將收集事件，並將其傳送至 vRealize Log Insight 伺服器。

欄位名稱受到限制。下列名稱已保留，無法做為欄位名稱。

- event_type

- hostname
- source
- text

必要條件

登入已安裝 vRealize Log Insight Windows 代理程式的 Windows 機器，然後啟動服務管理員以確認 vRealize Log Insight 代理程式服務已安裝。

程序

- 1 導覽至 vRealize Log Insight Windows 代理程式的程式資料目錄。

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

- 2 在任一文字編輯器中開啟 liagent.ini 檔案。
- 3 新增下列參數，並設定您環境適用的值。

參數	說明
[winlog section_name]	組態區段的唯一名稱。
channel	事件通道的全名正如在事件檢視器內建 Windows 應用程式中顯示的名稱。若要複製正確的通道名稱，請在事件檢視器中的通道上按一下滑鼠右鍵，選取 內容 ，並複製 全名 欄位的內容。
enabled	用來啟用或停用組態區段的選擇性參數。可能的值為 yes 或 no (區分大小寫)。預設值為 yes。
tags	用於將自訂標籤新增至所收集事件之欄位的選擇性參數。使用 JSON 標記法定義標籤。標籤名稱可以包含字母、數字及底線。標籤名稱只能以字母或底線開頭，並且不能超過 64 個字元。標籤名稱不區分大小寫。例如，如果您使用 tags={"tag_name1": "tag value 1", "Tag_Name1": "tag value 2"}，則 Tag_Name1 將因為重複而被忽略。您不能將 event_type 和時間戳記用作標籤名稱。同一宣告內的任何重複項目都將被忽略。 如果目的地為 Syslog 伺服器，則標籤可以覆寫 APP-NAME 欄位。例如，tags={"appname": "VROPS"}。
whitelist, blacklist	用來明確包含或排除記錄事件的選擇性參數。 備註 blacklist 選項僅適用於欄位，無法用於封鎖文字。
exclude_fields	(選擇性) 用來從收集排除個別欄位的參數。您可以以分號分隔之清單的形式提供多個值。例如，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 儲存並關閉 liagent.ini 檔案。

範例：組態

請參閱下列 [winlog] 組態範例。

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

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

設定 Windows 事件通道的篩選

您可以針對 Windows 事件通道設定篩選器，以明確納入或排除記錄事件。

使用 `whitelist` 和 `blacklist` 參數評估篩選器運算式。篩選器運算式是一種布林運算式，由事件欄位和運算子所組成。

備註 `blacklist` 選項僅適用於欄位，無法用於封鎖文字。

- `whitelist` 參數僅收集篩選器運算式評估為非零的記錄事件。如果您省略此參數，則值為默許的 1。
- `blacklist` 參數會排除篩選器運算式評估為非零的記錄事件。預設值為 0。

如需 Windows 事件欄位和運算子的完整清單，請參閱[事件欄位和運算子](#)。

必要條件

登入已安裝 vRealize Log Insight Windows 代理程式的 Windows 機器，然後啟動服務管理員以確認 vRealize Log Insight 代理程式服務已安裝。

程序

- 1 導覽至 vRealize Log Insight Windows 代理程式的程式資料目錄。

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

- 2 在任一文字編輯器中開啟 `liagent.ini` 檔案。
- 3 在 [winlog] 區段新增 `whitelist` 或 `blacklist` 參數。

例如

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

- 4 從 Windows 事件欄位和運算子建立篩選器運算式。

例如

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

5 儲存並關閉 liagent.ini 檔案。

範例：篩選器組態

您可以將代理程式設定為僅收集錯誤事件，例如

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

您可以將代理程式設定為僅從「應用程式」通道收集 VMware 網路事件，例如

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

您可以將代理程式設定為從「安全性」通道收集所有事件，但特定的事件除外，例如

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

事件欄位和運算子

使用 Windows 事件欄位和運算子建立篩選器運算式。

篩選器運算式運算子

運算子	說明
==, !=	等於和不等於。同時搭配使用數字欄位和字串欄位。
>=, >, <, <=	大於或等於、大於、小於、小於或等於。僅搭配使用數字欄位。
&, , ^, ~	位元 AND、OR、XOR 以及補充運算子。僅搭配使用數字欄位。
和、或	邏輯 AND 和 OR。用於透過合併簡單運算式來建立複雜運算式。
not	一元邏輯 NOT 運算子。用於反向運算式的值。
()	在邏輯運算式中使用括號來變更評估的順序。

Windows 事件欄位

您可以在篩選器運算式中使用下列 Windows 事件欄位。

欄位名稱	欄位類型
主機名稱	字串
Text	字串
提供者名稱	字串
事件來源名稱	字串
事件識別碼	數字
事件記錄識別碼	數字

欄位名稱	欄位類型
Channel	字串
使用者識別碼	字串
Level	數字 您可以使用下列預先定義的常數 <ul style="list-style-type: none"> ■ WINLOG_LEVEL_SUCCESS = 0 ■ WINLOG_LEVEL_CRITICAL = 1 ■ WINLOG_LEVEL_ERROR = 2 ■ WINLOG_LEVEL_WARNING = 3 ■ WINLOG_LEVEL_INFO = 4 ■ WINLOG_LEVEL_VERBOSE = 5
Task	數字
OpCode	數字
Keywords	數字 您可以使用下列預先定義的位元遮罩 <ul style="list-style-type: none"> ■ WINLOG_KEYWORD_RESPONSETIME = 0x0001000000000000; ■ WINLOG_KEYWORD_WDICONTEXT = 0x0002000000000000; ■ WINLOG_KEYWORD_WDIDIAGNOSTIC = 0x0004000000000000; ■ WINLOG_KEYWORD_SQM = 0x0008000000000000; ■ WINLOG_KEYWORD_AUDITFAILURE = 0x0010000000000000; ■ WINLOG_KEYWORD_AUDITSUCCESS = 0x0020000000000000; ■ WINLOG_KEYWORD_CORRELATIONHINT = 0x0040000000000000; ■ WINLOG_KEYWORD_CLASSIC = 0x0080000000000000;

範例

收集所有重大、錯誤以及警告事件

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

僅從「安全性」通道收集「稽核失敗」事件

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

從記錄檔收集事件

您可以將 vRealize Log Insight Windows 代理程式設定為從一或多個記錄檔收集事件。

欄位名稱受到限制。下列名稱已保留，無法做為欄位名稱。

- event_type
- hostname
- source

■ text

您最多可為代理程式資訊指定三個目的地，並可先篩選資訊再加以傳送。請參閱從 [vRealize Log Insight 代理程式轉送資訊](#)。

備註

- 監控大量檔案 (例如一千個以上) 時，會導致代理程式使用過多資源，並且影響主機的整體效能。若要防止這種情形，請使用模式和 Glob 將代理程式設定為僅監控必要的檔案，或封存舊記錄檔。如果有監控大量檔案的需求，請考慮增加主機參數，例如 CPU 和 RAM。
- 代理程式由加密目錄的使用者執行時，才能從加密的目錄進行收集。
- 代理程式僅支援靜態目錄結構。如果已重新命名或新增目錄，且組態涵蓋目錄，則您必須重新啟動代理程式才能開始監控這些目錄。

必要條件

登入已安裝 vRealize Log Insight Windows 代理程式的 Windows 機器，然後啟動服務管理員以確認 vRealize Log Insight 代理程式服務已安裝。

程序

- 1 導覽至 vRealize Log Insight Windows 代理程式的程式資料目錄。

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

- 2 在任一文字編輯器中開啟 `liagent.ini` 檔案。

3 找到檔案的 [server|<dest_id>] 區段。新增組態參數，並設定您環境適用的值。

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

參數	說明
[filelog section_name]	組態區段的唯一名稱。
directory=full-path-to-log-file	<p>記錄檔目錄的完整路徑。支援 Glob 模式。範例組態：</p> <ul style="list-style-type: none"> ■ 若要從 D:\Logs\new_test_logs 目錄的所有子目錄中收集，請使用 directory=D:\Logs\new_test_logs* ■ 如果子目錄本身也有子目錄，請使用下列組態來監控所有子目錄 directory=D:\Logs\new_test_logs** <p>備註 為了限制檔案和目錄的數目，並避免耗用太多資源，您不能對第一層或第二層目錄定義目錄 Glob，例如：directory=c:/tmp/* 或 directory=c:\Logs*。目錄路徑必須至少有兩個層級。</p> <p>您可以定義不存在目錄的路徑，一旦建立目錄和檔案之後，代理程式即會在該目錄中收集記錄檔。</p> <p>您可以在一或多個不同組態區段下定義相同的目錄，以從同一個檔案多次收集記錄。此程序可將不同的標籤和篩選器套用至相同的事件來源。</p> <p>備註 如果您針對這些區段使用完全相同的組態，則在伺服器端會出現重複事件。</p>
include=file_name;...	<p>(選擇性) 要從中收集資料的檔案名稱或檔案遮罩 (全域模式)。您可以分號分隔之清單的形式提供值。預設值為 *，表示已包含所有檔案。參數區分大小寫。</p> <p>檔案遮罩 (Glob 模式) 可以用來分組遵循相同命名慣例的檔案以及單一檔案名稱內的檔案。例如，包含空格的檔案名稱，例如 vRealize Ops Analytics.log 和 vRealize Ops Collector.log，皆可使用 vRealize?Ops?Analytics*.log 或 vRealize*.log 來指定。透過使用檔案遮罩，您可以指定在 Linux 和 Windows 主機下的代理程式組態可接受的檔案名稱。</p> <p>依預設，.zip 和 .gz 檔案已排除在收集之外。</p> <p>重要 如果您正在收集輪替記錄檔，請使用 include 和 exclude 參數指定同時符合主要和輪替檔案的全域模式。如果全域模式僅符合主要記錄檔，則 vRealize Log Insight 代理程式可能會在輪替期間遺失事件。vRealize Log Insight 代理程式會自動決定輪替檔案的正確順序，並以正確的順序將事件傳送到 vRealize Log Insight 伺服器。例如，如果主要記錄檔名為 myapp.log，輪替記錄檔為 myapp.log.1 和 myapp.log.2 (以此類推)，您可以使用下列 include 模式：</p> <pre>include= myapp.log;myapp.log.*</pre>
exclude=regular_expression	<p>(選擇性) 排除在收集之外的檔案名稱或檔案遮罩 (全域模式)。您可以分號分隔之清單的形式提供值。預設值為空白，表示未排除任何檔案。</p>

參數	說明
event_marker=regular_expression	<p>(選擇性) 表示記錄檔中事件開始的規則運算式。如果省略，將預設為換行。您輸入的運算式必須使用 Perl 規則運算式語法。</p> <p>備註 系統不會將符號，例如引號 (" ")，視為規則運算式的壓縮包。系統會將其視為模式的一部分。</p> <p>由於 vRealize Log Insight 代理程式已針對即時收集最佳化，內部延遲寫入的部分記錄訊息可能會分割成多個事件。若記錄檔附加停止超過 200 毫秒且沒有新觀察到的 event_marker，則部分事件會被視作完成、已剖析和已傳遞。此計時邏輯無法設定且優先順序高於 event_marker 設定。記錄檔附加器應排清完整事件。</p>
enabled=yes no	(選擇性) 啟用或停用組態區段的參數。可能的值為 yes 或 no。預設值為 yes。
charset=char-encoding-type	<p>(選擇性) 代理程式監控之記錄檔的字元編碼。可能的值為：</p> <ul style="list-style-type: none"> ■ UTF-8 ■ UTF-16LE ■ UTF-16BE <p>預設值為 UTF-8。</p>
tags={"tag-name": "tag-value", ...}	<p>(選擇性) 將自訂標籤新增到已收集事件之欄位的參數。使用 JSON 標記法定義標籤。標籤名稱可以包含字母、數字及底線。標籤名稱只能以字母或底線開頭，並且不能超過 64 個字元。標籤名稱不區分大小寫。例如，如果您使用 tags={"tag_name1": "tag value 1", "Tag_Name1": "tag value 2"}，則 Tag_Name1 將因為重複而被忽略。您不能將 event_type 和時間戳記用作標籤名稱。同一宣告內的任何重複項目都將被忽略。</p> <p>如果目的地為 Syslog 伺服器，則標籤可以覆寫 APP-NAME 欄位。例如，tags={"appname": "VROPS"}。</p>
exclude_fields	<p>(選擇性) 用來從收集排除個別欄位的參數。您可以分號或逗號分隔的清單形式提供多個數值。例如，</p> <ul style="list-style-type: none"> ■ exclude_fields=hostname; filepath ■ exclude_fields=type; size ■ exclude_fields=type, size
raw_syslog=Yes No	<p>針對使用 Syslog 通訊協定的代理程式，此選項可讓代理程式收集和傳送原始 Syslog 事件。預設值為 [否]，這表示會以使用者指定的 Syslog 屬性來轉換已收集的事件。啟用此選項以收集事件而不進行任何 Syslog 轉換。</p>

範例：組態

```
[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]
```

設定 Windows 記錄檔通道篩選

您可以針對 Windows 記錄檔設定篩選器，以明確納入或排除記錄事件。

使用 `whitelist` 和 `blacklist` 參數評估篩選器運算式。篩選器運算式是一種布林運算式，由事件欄位和運算子所組成。

備註 `blacklist` 選項僅適用於欄位，無法用於封鎖文字。

- `whitelist` 參數僅收集篩選器運算式評估為非零的記錄事件。如果您省略此參數，則值為默許的 1。
- `blacklist` 參數會排除篩選器運算式評估為非零的記錄事件。預設值為 0。

如需 Windows 事件欄位和運算子的完整清單，請參閱[事件欄位和運算子](#)。

必要條件

登入已安裝 vRealize Log Insight Windows 代理程式的 Windows 機器，然後啟動服務管理員以確認 vRealize Log Insight 代理程式服務已安裝。

程序

- 1 導覽至 vRealize Log Insight Windows 代理程式的程式資料目錄。

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

- 2 在任一文字編輯器中開啟 `liagent.ini` 檔案。
- 3 在 `[filelog|]` 區段新增 `whitelist` 或 `blacklist` 參數。

例如：

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

- 4 從 Windows 事件欄位和運算子建立篩選器運算式。

例如

```
whitelist = myServer
```

- 5 儲存並關閉 `liagent.ini` 檔案。

範例：篩選器組態

您可以將代理程式設定為僅收集 Apache 記錄，其中 server_name 為

```
[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"
```

將事件轉送到 Log Insight Windows Agent

您可以將事件從 Windows 機器轉送到執行 Log Insight Windows Agent 的機器。

您可以使用 Windows 事件轉送將事件從多部 Windows 機器轉送到一部安裝有 Log Insight Windows Agent 的機器。隨後即可將 Log Insight Windows Agent 設定為收集所有轉送事件，並將其傳送到 vRealize Log Insight 伺服器。

熟悉 Windows 事件轉送。請參閱 <http://technet.microsoft.com/en-us/library/cc748890.aspx> 和 [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)。

必要條件

請參閱 [從 Windows 事件通道收集事件](#)。

程序

- 1 在 Log Insight Windows Agent 組態中新增一個區段，以便從接收轉送事件的 Windows 事件通道收集事件。

預設通道名稱為 ForwardedEvents。

- 2 設定 Windows 事件轉送。

後續步驟

前往 vRealize Log Insight Web 使用者介面，確認轉送事件已送達。

設定 Log Insight Linux Agent

安裝完 Log Insight Linux Agent 之後您可以加以設定。

您可以使用[集中式代理程式組態](#)將代理程式設定為將事件傳送至 vRealize Log Insight 伺服器、指定通訊協定和連接埠，以及設定一般檔案記錄收集。如需 liagent.ini 檔案的位置，請參閱[定義 Log Insight Agents 中的記錄詳細資料層級](#)。

vRealize Log Insight Linux 代理程式的預設組態

安裝完成後，liagent.ini 檔案將包含為 Log Insight Windows Agent 預先設定的預設設定。

vRealize Log Insight Linux 代理程式 liagent.ini 預設組態

如果您使用非 ASCII 名稱和值，將組態儲存為 UTF-8。

如果您使用中央組態，最終的組態即為此使用伺服器設定所加入的檔案以形成 liagent-effective.ini 檔案。

您會發現從伺服器的代理程式頁面進行設定會更有效。

```
[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.?
```

參數	預設值	說明
hostname	LOGINSIGHT	vRealize Log Insight 虛擬應用裝置的 IP 位址或主機名稱。 預設值為 <code>loginsight</code> 。
central_config	yes	啟用或停用此代理程式的集中式組態。停用集中式組態時，代理程式會忽略 vRealize Log Insight 伺服器所提供的組態。接受的值為 <code>yes</code> 、 <code>no</code> 、 <code>1</code> 或 <code>0</code> 。預設值為 <code>yes</code> 。
proto	cfapi	代理程式用於傳送事件至 vRealize Log Insight 伺服器的通訊協定。可能的值為 <code>cfapi</code> 和 <code>syslog</code> 。 預設值為 <code>cfapi</code> 。
port	9543、9000、6514 和 514	代理程式用於傳送事件至 vRealize Log Insight 伺服器的通訊連接埠。對於啟用 SSL 的 <code>cfapi</code> ，預設值為 9543；對於停用 SSL 的 <code>cfapi</code> ，預設值為 9000；對於啟用 SSL 的 <code>Syslog</code> ，預設值為 6514；對於停用 SSL 的 <code>Syslog</code> ，預設值為 514。
ssl	yes	啟用或停用 SSL。預設值為 <code>yes</code> 。 當 <code>ssl</code> 設為「 <code>yes</code> 」時，如果您沒有為連接埠設定一個值，連接埠會自動選取為 9543。
max_disk_buffer	200	Log Insight Windows Agent 用於緩衝事件及其自身記錄檔的最大磁碟空間 (以 MB 為單位)。 達到指定的 <code>max_disk_buffer</code> 後，代理程式將開始捨棄新的傳入事件。
debug_level	0	定義記錄詳細資料層級。請參閱 定義 Log Insight Agents 中的記錄詳細資料層級 。

從記錄檔收集事件

您可以將 vRealize Log Insight Linux 代理程式設定為從一或多個記錄檔收集事件。

依預設，vRealize Log Insight Linux 代理程式會收集應用程式或編輯器所建立的隱藏檔案。隱藏的檔案名稱會以句點開頭。您可以加入排除參數 `exclude=.*`，以防止 vRealize Log Insight Linux 代理程式收集隱藏檔案。

欄位名稱受到限制。下列名稱已保留，無法做為欄位名稱。

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

您最多可為代理程式資訊指定三個目的地，並可先篩選資訊再加以傳送。請參閱[從 vRealize Log Insight 代理程式轉送資訊](#)

備註 監控大量檔案 (例如一千個以上) 時，會導致 vRealize Log Insight 代理程式使用過多資源，並且影響主機的整體效能。若要防止這種情形，請使用模式和 Glob 將代理程式設定為僅監控必要的檔案，或封存舊記錄檔。如果有監控大量檔案的需求，請考慮增加主機參數，例如 CPU 和 RAM。

必要條件

- 以根使用者身分登入，或使用 `sudo` 執行主控台命令。
- 確認 vRealize Log Insight Linux 代理程式已安裝並執行中。登入您安裝 vRealize Log Insight Linux 代理程式的 Linux 機器、開啟主控台，並執行 `pgrep liagent`。

程序

- 1 在任何文字編輯器中開啟 `/var/lib/loginsight-agent/liagent.ini` 檔案。
- 2 找到檔案的 `[server|<dest_id>]` 區段。新增組態參數，並設定您環境適用的值。

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

參數	說明
<code>[filelog section_name]</code>	組態區段的唯一名稱。
<code>directory=full-path-to-log-file</code>	<p>記錄檔目錄的完整路徑。支援 Glob 模式。範例組態：</p> <ul style="list-style-type: none"> ■ 若要從 <code>D:\Logs\new_test_logs</code> 目錄的所有子目錄中收集，請使用 <code>directory=D:\Logs\new_test_logs*</code> ■ 如果子目錄本身也有子目錄，請使用下列組態來監控所有子目錄 <code>directory=D:\Logs\new_test_logs**</code> <p>備註 為了限制檔案和目錄的數目，並避免耗用太多資源，您不能對第一層或第二層目錄定義目錄 Glob，例如：<code>directory=c:/tmp/*</code> 或 <code>directory=c:\Logs*</code>。目錄路徑必須至少有兩個層級。</p> <p>您可以定義不存在目錄的路徑，一旦建立目錄和檔案之後，代理程式即會在該目錄中收集記錄檔。</p> <p>您可以在一或多個不同組態區段下定義相同的目錄，以從同一個檔案多次收集記錄。此程序可將不同的標籤和篩選器套用至相同的事件來源。</p> <p>備註 如果您針對這些區段使用完全相同的組態，則在伺服器端會出現重複事件。</p>

參數	說明
<code>include= <i>file_name</i>; ...</code>	<p>(選擇性) 要從中收集資料的檔案名稱或檔案遮罩 (全域模式)。您可以分號分隔之清單的形式提供值。預設值為 <code>*</code>，表示已包含所有檔案。參數區分大小寫。</p> <p>檔案遮罩 (Glob 模式) 可以用來分組遵循相同命名慣例的檔案以及單一檔案名稱內的檔案。例如，包含空格的檔案名稱，例如 <code>vRealize Ops Analytics.log</code> 和 <code>vRealize Ops Collector.log</code>，皆可使用 <code>vRealize?Ops?Analytics*.log</code> 或 <code>vRealize*.log</code> 來指定。透過使用檔案遮罩，您可以指定在 Linux 和 Windows 主機下的代理程式組態可接受的檔案名稱。</p> <p>依預設，<code>.zip</code> 和 <code>.gz</code> 檔案已排除在收集之外。</p> <p>重要 如果您正在收集輪替記錄檔，請使用 <code>include</code> 和 <code>exclude</code> 參數指定同時符合主要和輪替檔案的全域模式。如果全域模式僅符合主要記錄檔，則 vRealize Log Insight 代理程式可能會在輪替期間遺失事件。vRealize Log Insight 代理程式會自動決定輪替檔案的正確順序，並以正確的順序將事件傳送到 vRealize Log Insight 伺服器。例如，如果主要記錄檔名為 <code>myapp.log</code>，輪替記錄檔為 <code>myapp.log.1</code> 和 <code>myapp.log.2</code> (以此類推)，您可以使用下列 <code>include</code> 模式：</p> <pre>include= myapp.log;myapp.log.*</pre>
<code>exclude= <i>regular_expression</i></code>	<p>(選擇性) 排除在收集之外的檔案名稱或檔案遮罩 (全域模式)。您可以分號分隔之清單的形式提供值。預設值為空白，表示未排除任何檔案。</p>
<code>event_marker= <i>regular_expression</i></code>	<p>(選擇性) 表示記錄檔中事件開始的規則運算式。如果省略，將預設為換行。您輸入的運算式必須使用 Perl 規則運算式語法。</p> <p>備註 系統不會將符號，例如引號 (" ")，視為規則運算式的壓縮包。系統會將其視為模式的一部分。</p> <p>由於 vRealize Log Insight 代理程式已針對即時收集最佳化，內部延遲寫入的部分記錄訊息可能會分割成多個事件。若記錄檔附加停止超過 200 毫秒且沒有新觀察到的 <code>event_marker</code>，則部分事件會被視作完成、已剖析和已傳遞。此計時邏輯無法設定且優先順序高於 <code>event_marker</code> 設定。記錄檔附加器應排清完整事件。</p>
<code>enabled=yes no</code>	<p>(選擇性) 啟用或停用組態區段的參數。可能的值為 <code>yes</code> 或 <code>no</code>。預設值為 <code>yes</code>。</p>
<code>charset= <i>char-encoding-type</i></code>	<p>(選擇性) 代理程式監控之記錄檔的字元編碼。可能的值為：</p> <ul style="list-style-type: none"> ■ UTF-8 ■ UTF-16LE ■ UTF-16BE <p>預設值為 UTF-8。</p>
<code>tags={"tag-name": "tag-value", ...}</code>	<p>(選擇性) 將自訂標籤新增到已收集事件之欄位的參數。使用 JSON 標記法定義標籤。標籤名稱可以包含字母、數字及底線。標籤名稱只能以字母或底線開頭，並且不能超過 64 個字元。標籤名稱不區分大小寫。例如，如果您使用 <code>tags={"tag_name1": "tag value 1", "Tag_Name1": "tag value 2"}</code>，則 <code>Tag_Name1</code> 將因為重複而被忽略。您不能將 <code>event_type</code> 和時間戳記用作標籤名稱。同一宣告內的任何重複項目都將被忽略。</p> <p>如果目的地為 Syslog 伺服器，則標籤可以覆寫 APP-NAME 欄位。例如，<code>tags={"appname": "VROPS"}</code>。</p>

參數	說明
exclude_fields	(選擇性) 用來從收集排除個別欄位的參數。您可以分號或逗號分隔的清單形式提供多個數值。例如， <ul style="list-style-type: none"> ■ exclude_fields=hostname; filepath ■ exclude_fields=type; size ■ exclude_fields=type, size
raw_syslog=Yes No	針對使用 Syslog 通訊協定的代理程式，此選項可讓代理程式收集和傳送原始 Syslog 事件。預設值為 [否]，這表示會以使用者指定的 Syslog 屬性來轉換已收集的事件。啟用此選項以收集事件而不進行任何 Syslog 轉換。

3 儲存並關閉 liagent.ini 檔案。

範例：組態

```
[filelog|messages]
directory=/var/log
include=messages;messages.?

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

[filelog|Apache]
directory=/var/log/apache2
include=*
```

篩選事件

您可以在 vRealize Log Insight Linux 代理程式上依照其欄位值篩選所有收集的事件，以指定要選取或捨棄的記錄事件。您可以使用 **whitelist** 和 **blacklist** 收集器選項來定義篩選器。

提示 依預設，vRealize Log Insight Linux 代理程式會收集程式或編輯器建立的隱藏檔案。隱藏檔案的名稱以句點開頭。您可阻止 vRealize Log Insight Linux 代理程式收集隱藏檔案，方式是新增排除 **exclude=* 參數**。

針對每個事件，收集器會評估 **whitelist** 和 **blacklist** 篩選器運算式。如果 **whitelist** 運算式評估為 **true**，且 **blacklist** 運算式評估為 **false** 或無法評估，則會將事件移至佇列進行進一步處理。在任何其他情況下，收集器會捨棄該事件。**whitelist** 運算式的預設值為 **true**，且 **blacklist** 運算式的預設值為 **false**。

提示 **Filelog** 收集器提供用於篩選的欄位較少。若要取得用於篩選的欄位，您可以剖析記錄檔。如需詳細資訊，請參閱[剖析記錄](#)。

`whitelist` 或 `blacklist` 篩選器是評估為單一邏輯或整數值的一組變數、常值與運算子。您可以使用事件欄位做為變數，以及雙引號的字串和數字做為常值。如需您可以在篩選器運算式內使用的運算子相關資訊，請參閱 [事件欄位和運算子](#)。

備註

- 如果您將數字與字串比較，或如果比較牽涉到數值字串，則會將每個字串轉換為數字，並且以數值形式執行比較。例如：
 - 運算式 `whitelist = 123.0 == "000123"` 會評估為 `true`。
 - 運算式 `whitelist = "00987" == "987.00"` 會評估為 `true`。
 - 在運算式 `whitelist = response_size >= "12.12"` 中，如果 `response_size` 欄位有數值，則以數值形式評估運算式。如果回應大小大於 12.12，則運算式為 `true`，否則為 `false`。
 - 在運算式 `whitelist = "09123" < "234"` 中，字串常值會轉換為數值並且運算式會評估為 `false`。
- 如果其中一個字串運算元無法轉換為數值，則會將這兩個運算元轉換為字串。執行簡易的區分大小寫字典式比較。例如：
 - 運算式 `whitelist = "1234a" == "1234A"` 是評估為 `false` 的字串比較。
 - 運算式 `whitelist = 4 < "four"` 會將 4 轉換為「4」，並評估為 `true`。
 - 在運算式 `whitelist = response_size > "thousand"` 中，`response_size` 欄位的值會轉換為字串值，它會將運算式評估為 `false`。
- 如果篩選器運算式評估為整數值，則如果它是 0，會被視為 `false`，否則為 `true`。
 例如，如果 `some_integer` 欄位已設定最低有效位元集，則運算式 `whitelist = some_integer & 1` 會評估為 `true`，否則為 `false`。

如需事件欄位和運算子的完整清單，請參閱[從記錄檔收集事件](#)。

在此範例中，您必須從檔案 `/var/log/httpd/access` 收集 Apache 存取記錄。來自該檔案的部分範例記錄為：

- 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

必要條件

- 以**根使用者**身分登入，或使用 `sudo` 執行主控台命令。
- 登入安裝有 vRealize Log Insight Linux 代理程式的 Linux 機器，開啟主控台並執行 `pgrep liagent`，以確認 vRealize Log Insight Linux 代理程式已安裝且正在執行。

程序

- 1 為記錄定義剖析器，如下列程式碼片段所示：

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

您已定義的剖析器會為從檔案 `/var/log/httpd/access` 收集的每個事件擷取 `remote_host`、`remote_log_name`、`remote_auth_user`、`timestamp`、`request`、`status_code` 以及 `response_size` 欄位。您可以使用這些欄位來篩選事件。

- 2 在任何文字編輯器中開啟 `/var/lib/loginsight-agent/liagent.ini` 檔案。
- 3 在檔案中定義 `Filelog` 區段來收集和剖析記錄，如下列程式碼片段所示：

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

- 4 根據您的需求篩選事件。

- 若要收集 HTTP 狀態為 200 的記錄，您可以在 `Filelog` 區段中定義 `whitelist`，如下列程式碼片段所示：

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

`whitelist` 運算式只會將來自範例記錄的第一個和第二個事件評估為 `true`，並且收集器會選擇這些事件。

如果事件中不存在 `status_code` 欄位，因為它不存在於記錄中或未剖析，則無法評估 `whitelist` 運算式，這表示它評估為 `false`，並且收集器會捨棄事件。

- 若要捨棄您不感興趣的事件，您可以使用 `blacklist` 選項。例如，如果您對本機流量不感興趣，可以將本機 IP 封鎖，如下列程式碼片段所示：

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

收集器會從範例記錄選取第二個、第三個和第四個事件。

- 若要根據多個述詞篩選所有事件，您可以使用 `or` 和 `and` 運算子。例如，您可以捨棄從本機 IP 所產生的事件或測試使用者從您不需要的任何主機產生的事件，如下列程式碼片段所示：

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

使用 `or` 運算子將 `blacklist` 運算式評估為 `true`，可略過不必要的事件。運算式會指示收集器在 `remote_host` 欄位值為「127.0.0.1」或 `remote_auth_user` 欄位值為「test」時捨棄事件。

收集器會從範例記錄選取第二個和第三個事件。

- 若要捨棄測試使用者從本機 IP 產生的事件，您可以在 `blacklist` 運算式中使用 `and`，如下列程式碼片段所示：

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

收集器會捨棄來自範例記錄的第五個事件。

- 您可以同時使用 `whitelist` 和 `blacklist` 篩選器。例如，如果您需要回應大小大於 1024 位元組的事件，但您不需要來源是本機主機的事件，您可以使用下列程式碼片段：

```
[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"
```

收集器會選取來自範例記錄的第二個事件。

從 journald 收集事件

從 vRealize Log Insight 4.6 開始，針對執行 `systemd` 的 Linux 發行版中的記錄資料，代理程式可以讀取來自 `journald` 系統服務的記錄。現在 `journald` 是登入 `systemd` 型 Linux 平台的預設標準。`journald` 組態區段支援下列選項：

journal_files

要監控的日誌檔案。支援的值如下：

值	說明
全部	開啟並監控所有可用的日誌檔案。
本機	僅監控並讀取本機電腦上產生的日誌檔案。
執行階段	僅監控並讀取可變更的日誌檔案，排除永久儲存區中的檔案。
系統	僅監控並讀取系統服務和核心日誌檔案。
個使用者	僅監控和讀取目前使用者的日誌檔案。

fetch_fields

要從日誌記錄項目擷取訊息的欄位。此選項的值是不區分大小寫、以逗號分隔的欄位名稱清單。支援的值如下：

值	說明
pri_severity、pri_facility、syslog_identifier	此選項的預設值。
*	擷取所有欄位。
全部	不擷取任何欄位。

篩選來自 vRealize Log Insight 代理程式的事件

您可以利用本機 `liagent.ini` 檔案 `[server|<dest_id>]` 區段中的篩選選項，提供代理程式傳送到目的地的資訊。

選項為以下其中一個格式：

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

篩選類型	說明
collector_type	定義收集器類型而以逗號分隔的清單。支援的值為 <code>filelog</code> 或 <code>winlog</code> 。如果未提供值，則會使用所有收集器類型。
collector_filter	以 <code>regex</code> 格式指定收集器區段的名稱。例如， <code>vcops_.*</code> 表示開頭為「 <code>vcops_</code> 」的所有收集器區段。
event_filter	篩選與收集器區段中允許名單或封鎖名單使用相同語法的事件欄位。代理程式僅會傳送將運算式評估為 <code>True</code> 或為非零值的事件。空白的 <code>event_filter</code> 一律會評估為 <code>True</code> 。若要對事件使用 <code>event_filter</code> ，您必須在適當的收集器區段中，針對欄位擷取定義剖析器。如果由於收集的事件中缺乏欄位而無法評估運算式，則會捨棄事件。

以逗號分隔即可指定多個篩選器運算式，如下列範例所示：

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

如果訊息符合目的地目標的多組篩選器條件組合，則僅會傳送一次。

表 4-1. 語法範例

篩選器	意義
filter= {winlog;Microsoft.*;}	僅在事件名稱開頭為「Microsoft」時，才會傳送來自 winlog 收集器的事件。
filter= {winlog;Microsoft.*; eventid == 1023}	僅在事件名稱開頭為「Microsoft」且事件識別碼等於 1023 時，才會傳送來自 winlog 收集器的事件。
filter= {.*;}	預設的篩選值。傳送來自所有來源的所有事件。
filter= {winlog;.*;}	傳送來自 winlog 區段的所有事件。
filter= {filelog;syslog;facility<5}	如果設施小於 5，則傳送來自 [filelog syslog] 區段的事件。[filelog syslog] 區段必須具有擷取設施欄位的剖析器。否則，系統會略過所有事件。
filter= {;;}	不比對任何事件。使用此語法以停用事件轉送。

下列範例會將篩選新增到前一個範例中第二個目的地的組態。

```
; 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; }
```

下一個範例使用更複雜的篩選器運算式。

```
; 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"}
```

以逗號分隔即可指定多個篩選器運算式，如下列範例所示。

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

從 vRealize Log Insight 代理程式轉送資訊

您可以將代理程式收集的事件轉送至最多三個目的地。目的地可包含 vRealize Log Insight 伺服器或轉送站，或第三方記錄管理解決方案。

例如，您可以將稽核或系統記錄傳送至安全團隊的伺服器、將應用程式記錄傳送至開發維運團隊伺服器，以及將度量記錄傳送至 IT 管理系統。您可以使用篩選器指定要將哪些資訊傳至目的地。您可從單一 vRealize Log Insight 代理程式將資訊轉送至最多三個目的地。

代理程式設定可透過本機 `liagent.ini` 檔案的 `[server|<dest_id>]` 區段來完成。對於 vRealize Log Insight 伺服器或轉送站，請使用 `cfapi` 通訊協定，對其他目標或目的地則使用 `syslog` 通訊協定。

為代理程式指定多個目的地時，第一個目的地會使用預設 `loginsight` 位置。您必須指定其他目的地的位置資訊。

下一個範例顯示指定兩個目的地之 `liagent.ini` 檔案的一部分。依預設會對第一個目的地隱含使用預設伺服器名稱 `loginsight`，而不會加以指定。第二個 `[server|<dest_id>]` 區段會指定目的地。

```
; 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
```

如需為代理程式建立篩選器的相關資訊，請參閱[篩選來自 vRealize Log Insight 代理程式的事件](#)。

設定目標 vRealize Log Insight 伺服器

您可以為在 Windows 上執行的 vRealize Log Insight 代理程式設定或變更目標 vRealize Log Insight 伺服器。您最多可將事件傳送至三個目的地，並篩選每個目的地的輸出。

預設目的地可透過 `liagent.ini` 檔案的 `[server]` 區段來設定。預設目的地一律存在，且依預設會將主機名稱設為 `loginsight`。若要新增多個目標目的地，請為每個目標建立一個 `[server|<dest_id>]` 區段。您必須指定唯一的主機名稱，做為每個其他連線的目的地識別碼。您可以針對預設的 `[server]` 區段將相同選項用於其他目的地。請不要為自動升級設定其他目的地，或將這些目的地用於代理程式設定。您可以指定兩個額外目的地。

依預設，代理程式會將所有收集的事件傳送至所有目的地。您可以使用 `file` 選項來篩選事件，以將不同的事件傳送至不同的目的地。如需詳細資訊，請參閱[篩選來自 vRealize Log Insight 代理程式的事件](#)。

必要條件

- 登入已安裝 vRealize Log Insight Windows 代理程式的 Windows 機器，然後啟動服務管理員以確認 vRealize Log Insight 代理程式服務已安裝。
- 如果您的 vRealize Log Insight 叢集已啟用整合式負載平衡器，請參閱[啟用整合式負載平衡器](#)，以了解自訂 SSL 憑證的特定需求。

程序

- 1 導覽至 vRealize Log Insight Windows 代理程式的程式資料目錄。

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

- 2 在任一文字編輯器中開啟 `liagent.ini` 檔案。

3 修改下列參數，並設定您環境適用的值。

參數	說明
proto	代理程式用於傳送事件至 vRealize Log Insight 伺服器的通訊協定。可能的值為 <code>cfapi</code> 和 <code>syslog</code> 。 預設值為 <code>cfapi</code> 。
hostname	vRealize Log Insight 虛擬應用裝置的 IP 位址或主機名稱。 您可以指定 IPv4 或 IPv6 位址。可以使用也可以不使用方括弧來指定 IPv6 位址。例如： <pre>hostname = 2001:cdba::3257:9652 or hostname = [2001:cdba::3257:9652]</pre> 如果主機同時支援 IPv4 和 IPv6 堆疊，且網域名稱已指定為主機名稱，則代理程式會根據名稱解析程式所傳回的 IP 位址，來選擇 IP 堆疊。如果解析程式同時傳回 IPv4 和 IPv6 位址，則代理程式會嘗試以指定順序依序連線至這兩個位址。
max_disk_buffer	Log Insight Windows 代理程式可用來對為此特殊伺服器收集的事件進行緩衝處理的磁碟空間上限 (以 MB 為單位)。這個選項會覆寫此伺服器的 <code>[storage].max_disk_buffer</code> 值。 預設值為 150 MB；您可以設定 50 到 8000 MB 的緩衝區大小。
port	代理程式用來將事件傳送至 vRealize Log Insight 或第三方伺服器的通訊連接埠。 依預設，代理程式會根據針對 SSL 和通訊協定而設定的選項使用適當的連接埠。請參閱下列清單中提供的預設連接埠值。只有在連接埠選項與這些預設值不同時，才需要指定此選項。 <ul style="list-style-type: none"> ■ 已啟用 SSL 的 <code>cfapi</code> : 9543 ■ 已停用 SSL 的 <code>cfapi</code> : 9000 ■ 已啟用 SSL 的 <code>Syslog</code> : 6514 ■ 已停用 SSL 的 <code>Syslog</code> : 514
ssl	啟用或停用 SSL。預設值為 <code>yes</code> 。 當 <code>ssl</code> 設為「是」時，連接埠會設為 9543，除非您另行指定。
reconnect	強制重新連線至伺服器的時間 (以分鐘為單位)。預設值為 30。
filter	指定代理程式傳送至目的地的資訊。此選項會使用三個引數： <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 儲存並關閉 liagent.ini 檔案。

範例

下列組態範例會設定使用受信任憑證授權機構的目標 vRealize Log Insight 伺服器。

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

下列範例說明包含每個目的地之篩選訊息的多目的地組態。

```
; 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

```

後續步驟

您可以針對 vRealize Log Insight 代理程式設定其他 SSL 選項。請參閱[設定伺服器與 Log Insight 代理程式之間的 SSL 連線](#)。

指定代理程式的目標

您可以指定最多三個目的地 vRealize Log Insight 以供 Linux 代理程式傳送事件。

多個目的地連線是透過 `li-agent.ini` 檔案的 `[server|<dest_id>]` 區段定義，其中 `<dest_id>` 是唯一的每一組態連線識別碼。您可以針對預設的 `[server]` 區段將相同選項用於其他目的地。但是，請不要為自動升級設定其他目的地，或將這些目的地用於代理程式設定。您可以指定兩個額外目的地。

您定義的第一個目標可以使用預設伺服器值 `loginsight`。在定義其他目標時，您必須在後續目標的 `[server]` 區段中指定主機名稱。未使用篩選時，代理程式會將所有收集到的事件傳送至所有目的地。這是預設行為。不過，您可以篩選事件，以將不同的事件傳送至不同的目的地。

必要條件

- 以**根使用者**身分登入，或使用 `sudo` 執行主控台命令。
- 登入安裝有 vRealize Log Insight Linux 代理程式的 Linux 機器，開啟主控台並執行 `pgrep liagent`，以確認 vRealize Log Insight Linux 代理程式已安裝且正在執行。
- 如果您的 vRealize Log Insight 叢集已啟用整合式負載平衡器，請參閱[啟用整合式負載平衡器](#)，以了解自訂 SSL 憑證的特定需求。

程序

- 1 在任何文字編輯器中開啟 `/var/lib/loginsight-agent/liagent.ini` 檔案。

2 修改下列參數，並設定您環境適用的值。

參數	說明
proto	代理程式用於傳送事件至 vRealize Log Insight 伺服器的通訊協定。可能的值為 <code>cfapi</code> 和 <code>syslog</code> 。 預設值為 <code>cfapi</code> 。
hostname	vRealize Log Insight 虛擬應用裝置的 IP 位址或主機名稱。 您可以指定 IPv4 或 IPv6 位址。可以使用也可以不使用方括弧來指定 IPv6 位址。例如： <pre>hostname = 2001:cdba::3257:9652 or hostname = [2001:cdba::3257:9652]</pre> 如果主機同時支援 IPv4 和 IPv6 堆疊，且網域名稱已指定為主機名稱，則代理程式會根據名稱解析程式所傳回的 IP 位址，使用 IP 堆疊。如果解析程式同時傳回 IPv4 和 IPv6 位址，則代理程式會嘗試以指定順序依序連線至這兩個位址。
max_disk_buffer	Log Insight Linux 代理程式可用來對為此特殊伺服器收集的事件進行緩衝處理的磁碟空間上限 (以 MB 為單位)。這個選項會覆寫此伺服器的 <code>[storage].max_disk_buffer</code> 值。 預設值為 150 MB；您可以設定 50 到 8000 MB 的緩衝區大小。
port	代理程式用來將事件傳送至 vRealize Log Insight 或第三方伺服器的通訊連接埠。依預設，代理程式會根據針對 SSL 和通訊協定而設定的選項使用適當的連接埠。請參閱下列清單中提供的預設連接埠值。只有在連接埠選項與這些預設值不同時，才需要指定此選項。 <ul style="list-style-type: none">■ 已啟用 SSL 的 <code>cfapi</code>：9543■ 已停用 SSL 的 <code>cfapi</code>：9000■ 已啟用 SSL 的 <code>Syslog</code>：6514■ 已停用 SSL 的 <code>Syslog</code>：514
ssl	啟用或停用 SSL。預設值為 <code>yes</code> 。 當 <code>ssl</code> 設為「yes」時，如果您沒有為連接埠設定一個值，連接埠會自動選取為 9543。
reconnect	強制重新連線到伺服器的時間 (以分鐘為單位)。預設值為 30。

```
[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 儲存並關閉 liagent.ini 檔案。

範例

下列組態範例會設定使用受信任憑證授權機構的目標 vRealize Log Insight 伺服器。

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

下列範例顯示多個目的地組態。

- 第一個 (預設) 目的地會接收所有收集的事件。

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

- 第二個目的地僅會透過一般 Syslog 通訊協定來接收 Syslog 事件。

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

- 第三個目的地會接收層級欄位等於「錯誤」或「警告」的 vRealize Operations Manager 事件，且會根據名稱開頭為「vrops-」的區段進行收集

```
[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
```

後續步驟

您可以針對 vRealize Log Insight Linux 代理程式設定其他 SSL 選項。請參閱[設定伺服器與 Log Insight 代理程式之間的 SSL 連線](#)。

vRealize Log Insight 代理程式的集中式組態

您可以設定多個 vRealize Log Insight 代理程式。

每個 vRealize Log Insight 代理程式都具有本機組態和伺服器端組態。本機組態儲存在安裝有 vRealize Log Insight 代理程式之虛擬或實體機器的 `liagent.ini` 檔案中。可存取和編輯伺服器端組態，例如，從 Web 使用者介面中的**管理 > 代理程式**。每個 vRealize Log Insight 代理程式的組態均由區段和金鑰組成。金鑰具有可設定的值。

vRealize Log Insight 代理程式會定期輪詢 vRealize Log Insight 伺服器並接收伺服器端組態。伺服器端組態和本機組態會合併以產生有效的組態。每個 vRealize Log Insight 代理程式均使用有效組態做為其運作組態。這些組態會逐一合併區段與金鑰。伺服器端組態中的值會覆寫本機組態中的值。合併規則如下：

- 如果某個區段僅存在於本機組態或伺服器端組態，則該區段及其所有內容都將成為有效組態的一部分。
- 如果某個區段同時存在於本機組態和伺服器端組態，則該區段中的金鑰將根據下列規則進行合併：
 - 如果某個金鑰僅存在於本機組態或伺服器端組態，則該金鑰及其值將成為有效組態中此區段的一部分。
 - 如果某個金鑰同時存在於本機組態和伺服器端組態，則該金鑰將成為有效組態中此區段的一部分，並且會使用伺服器端組態中的值。

管理員 vRealize Log Insight 使用者可將集中式組態套用至所有 vRealize Log Insight 代理程式。例如，您可導覽至**管理**頁面，在**管理**區段中，按一下**代理程式**。在**代理程式組態**方塊中輸入組態設定，然後按一下**儲存所有代理程式的組態**。該組態將在下一個輪詢週期套用至所有可設定的作用中代理程式。

管理員 vRealize Log Insight 使用者也可以在代理程式群組中使用特定的篩選器 (例如依作業系統、代理程式版本、主機名稱或 IP 範圍)，並將組態套用至特定的 vRealize Log Insight 代理程式。如需代理程式群組的相關資訊，請參閱《使用代理程式群組》。

備註

- 您只能將集中式組態套用至使用 cfapi 通訊協定的 vRealize Log Insight 代理程式。
- 在下列任一案例中，無法設定 vRealize Log Insight 代理程式：
 - 目前 vRealize Log Insight 伺服器不是主要目的地。如需設定多個目的地的相關資訊，請參閱[指定代理程式的目標](#)。
 - 參數 `central_config = no` 會在代理程式組態中使用。如需適用於 Windows 的預設代理程式組態的相關資訊，請參閱[Log Insight Windows Agent 的預設組態](#)。

組態合併範例

合併 Log Insight Windows Agent 的本機和伺服器端組態的範例。

本機組態

您可能具有 Log Insight Windows Agent 的下列本機組態。

```
[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} - -
```

伺服器端組態

您可以使用 Web 使用者介面的**管理 > 代理程式**頁面向所有代理程式套用集中式組態。例如，您可以排除和新增收集通道，並變更預設的重新連線設定。

```
[server]
reconnect=20

[winlog|Security]
```

```
channel=Security
enabled=no

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

有效組態

有效組態是本機與伺服器端組態合併的結果。Log Insight Windows Agent 設定為：

- 每 20 分鐘與 vRealize Log Insight 伺服器重新連線一次
- 繼續收集應用程式和系統事件通道
- 停止收集安全性事件通道
- 開始收集 Microsoft-Windows-DeviceSetupManager/Operational 事件通道
- 繼續收集 ApacheAccessLogs

```
[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} - -
```

針對代理程式組態使用一般值

您可以使用為 Windows 或 Linux 代理程式的每個代理程式組態區段套用的一般參數值，來覆寫代理程式組態檔的預設值。

一般選項

liagent.ini 組態檔的 [common|global] 區段中指定的選項會散佈到所有區段，[common|filelog] 區段中指定的選項只會散佈到所有 filelog 區段，而 [common|winlog] 選項只會散佈到所有 winlog 區段。

您可以在一般區段中定義 tags, include, exclude, event_marker, charset, exclude_fields 和 parser 參數，如下列範例所示。這是 Windows 代理程式的範例：

```
[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

...
```

此範例會指定下列行為：

- 所有來自 filelog 區段的記錄都具有 log_source_vm 和 collector_type 標籤及其對應值。
- 系統會從所有傳送的記錄中排除 test_tag 和 some_other_tag 標籤。
- 系統會將 auto 剖析器套用至所有收集的記錄。
- 依預設，所有 filelog 收集器都會從監控中排除 *.trc 檔案。

系統也會將 [common|global] 中的選項套用至所有 winlog 區段。

合併和覆寫準則

如果已在多個區段中定義選項，則系統會合併或覆寫其值，且合併/覆寫時，具有較小範圍的區段會具有較高優先順序。這表示來自 [common|filelog] 的值會合併或覆寫來自 [common|global] 的值，而來自 [filelog|sample_section] 的值也會合併或覆寫該值。

合併和覆寫行為會遵循下列規則：

- 值代表值清單 (tags、include、exclude 和 exclude_fields) 的選項會與該選項來自具有較高優先順序之區段的值合併。如先前所述，在具有標籤的情況下，來自具有較高優先順序之區段的標籤值會覆寫來自具有較低優先順序之區段的相同標籤值。
- 選項來自具有較高優先順序之區段的值會覆寫可具有單值 (event_marker、charset 和 parser) 的選項值。

這表示來自 [filelog|sample_section] 的 charset=UTF-8 值會覆寫來自 [common|global] 的 charset= UTF-16LE 全域值。

因此，例如，如果您在 `[common|filelog]` 中具有 `tags={"app":"global-test"}`，且在 `[filelog|flg_test_section]` 中具有 `tags={"app":"local-test","section":"flg_test_section"}`，則來自 `[filelog|flg_test_section]` 區段的 "app" 標籤值會覆寫來自 `[common|filelog]` 的值。透過此 `filelog` 區段收集的所有記錄都會具有包含 "local-test" 值的額外 "app" 標籤，以及包含 "flg_test_section" 值的 "section" 標籤。winlog 區段的優先順序鏈結相同，任何 `[winlog|...]` 區段都具有最高優先順序，而 `[common|global]` 具有最低優先順序。

在一般區段中指定無效值時，系統通常會略過這些值，且不會將其與來自先前和對應之 `filelog/winlog` 區段的值合併。在標籤或 `exclude_fields` 選項具有無效值的情況下，代理程式會盡可能擷取較多的有效資料，並在遇到無效資料後略過其餘檔案。代理程式記錄檔中會報告所有異常。如果發生非預期的行為，請參閱記錄檔並修正代理程式報告的所有錯誤。

如果代理程式在 `filelog` 或 `winlog` 區段中偵測到選項的無效值，則不會合併來自該區段的選項值與來自一般區段的選項值，且不會啟用該區段。代理程式記錄檔中會報告所有錯誤。如果發生非預期的行為，請參閱記錄檔並修正代理程式報告的所有錯誤。

剖析記錄

代理程式端記錄剖析器會從原始記錄擷取結構化資料，然後傳遞至 vRealize Log Insight 伺服器。使用記錄剖析器，vRealize Log Insight 可從中分析記錄、擷取資訊並在伺服器上顯示這些結果。可針對 Windows 和 Linux vRealize Log Insight 代理程式設定記錄剖析器。

如果使用 Syslog 通訊協定，則根據 RFC5424，剖析器所擷取的欄位為 STRUCTURED-DATA 的一部分。

設定記錄剖析器

您可以為 FileLog 和 WinLog 收集器設定剖析器。

必要條件

對於 vRealize Log Insight Linux 代理程式：

- 以根使用者身分登入，或使用 `sudo` 執行主控台命令。
- 登入安裝有 Log Insight Linux 代理程式的 Linux 機器，開啟主控台並執行 `pgrep liagent`，以確認 Log Insight Linux 代理程式已安裝且正在執行。

對於 vRealize Log Insight Windows 代理程式：

- 登入安裝有 Log Insight Windows 代理程式的 Windows 機器，然後啟動服務管理員以確認 vRealize Log Insight 服務已安裝。

程序

- 1 導覽到包含 `liagent.ini` 檔案的資料夾。

作業系統	路徑
Linux	<code>/var/lib/loginsight-agent/</code>
Windows	<code>%ProgramData%\VMware\Log Insight Agent</code>

- 2 在任一文字編輯器中開啟 `liagent.ini` 檔案。
- 3 若要設定特定剖析器，請定義剖析器區段。[parser|myparser]

其中 `myparser` 是剖析器的屬性名稱，可從記錄來源進行參考。剖析器區段應參考任何內建 (或任何其他定義的) 剖析器，並設定剖析器的強制選項及設定非必要選項 (如果需要)。

例如，`base_parser=csv` 顯示 `myparser` 剖析器衍生自內建剖析器 `csv`。按照預期，輸入記錄包含兩個以分號分隔的欄位。

```
[parser|myparser]

base_parser=csv

fields=field_name1,field_name2

delimiter=";"
```

- 4 定義 `myparser` 後，可從記錄來源 `winlog` 或 `filelog` 進行參考。

```
[filelog|some_csv_logs]

directory=D:\Logs

include=*.txt;*.txt.*

parser=myparser
```

從 `some_csv_logs` 來源收集的記錄 (例如，從 `D:\Logs` 目錄) 由 `myparser` 剖析，擷取的事件分別以 `field_name1` 和 `field_name2` 顯示在伺服器上。

備註 代理程式未將 `D:\Logs` 目錄中的靜態記錄提取到 vRealize Log Insight。但是，可在 vRealize Log Insight 中取得在 `D:\Logs` 目錄中建立的新檔案。

- 5 儲存並關閉 `liagent.ini` 檔案。

剖析器的一般選項

您可以為產生具名欄位的所有剖析器設定一般選項。

欄位名稱的保留字

欄位名稱受到限制。下列名稱已保留，無法做為欄位名稱。

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

常用的剖析器選項

下表中的選項可用於所有支援的剖析器。

選項	說明
<code>base_parser</code>	此自訂剖析器延伸的基礎剖析器的名稱。它可以是內建剖析器名稱或其他自訂剖析器名稱。此組態金鑰為強制項目。
<code>field_decoder</code>	<p>指定為 JSON 字串的巢狀剖析器。索引鍵是要套用巢狀剖析器的欄位名稱，值是要用於該欄位的剖析器名稱。每個巢狀剖析器都會套用至基礎剖析器解碼的適當欄位。當欄位的值為複雜值 (例如時間戳記) 時，欄位解碼器十分有用。<code>field_decoder</code> 選項也支援使用更複雜的 JSON 物件來作為引數，這可讓您對特定欄位值使用條件，條件經檢查沒有問題後再套用巢狀剖析器。</p> <p>備註 如需使用方式和條件式組態的詳細資訊，請參閱下方的〈<code>field_decoder</code> 選項的條件式組態</p>
<code>field_rename</code>	重新命名擷取的欄位。請使用索引鍵是欄位的原始名稱、值是新欄位名稱的 JSON 字串。 <code>field_decoder</code> 選項一律會在 <code>field_rename</code> 之前套用。INI 檔案中這些選項的順序不重要。為了充分釐清，請先指定 <code>field_decoder</code> 。
<code>next_parser</code>	<p>下一個要執行的剖析器的名稱。允許多個剖析器針對相同的輸入依序執行。</p> <p>備註 剖析器將處理所有由 <code>next_parser</code> 關鍵字定義的後續剖析器，並且可能取代已由之前的剖析器擷取的欄位值。</p>
<code>exclude_fields</code>	在傳遞至伺服器之前，要從事件移除之以分號分隔的欄位名稱清單。在執行事件篩選之前會先移除欄位名稱，如此，您在剖析期間排除的欄位將無法在篩選條件中使用。
<code>debug</code>	<p>允許對特定剖析器偵錯的 [是] 或 [否] 選項。啟用偵錯後，剖析器將詳細記錄其接收的輸入、執行的作業及產生的結果。選項會依區段來套用，也就是，僅套用至特定區段定義的剖析器。</p> <p>對於剖析器，偵錯預設值為 <code>debug=no</code>。</p>

`field_decoder` 選項的條件式組態

對於具有相同通用格式但在特定欄位值方面存在極大差異的記錄 (例如，具有 **info** 和 **error** 嚴重性的記錄)，您可以使用條件式巢狀剖析器，以減少對已剖析記錄的對應欄位套用不必要的剖析器。

例如，使用下列記錄時：

```
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 --
```

您可以使用下列組態來剖析這些記錄：

```
[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"}
```

此組態會產生以下結果：

```
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.)"
```

此外，還會對 **info** 記錄剖析下列欄位：

```
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 --"
```

以逗點分隔的值記錄剖析器

您可以為 FileLog 和 WinLog 收集器設定以逗點分隔的值 (CSV) 剖析器。

csv 剖析器的可用選項為 `fields` 和 `delimiter`。

以逗點分隔的值剖析器選項

請注意下面關於 CSV 剖析器結構的資訊。

選項	說明
fields	<p><code>fields</code> 選項指定記錄中存在之欄位的名稱。列出的欄位名稱總數必須等於記錄中以逗號分隔的欄位總數。</p> <p><code>fields</code> 是 CSV 剖析器的必要選項。如果未指定，將不會進行任何剖析。視欄位內容而定，欄位值周圍的雙引號是選擇性的。</p> <p>欄位名稱必須以逗號分隔，例如</p> <pre>fields = field_name1, field_name2, field_name3, field_name4</pre> <p>此定義假設 <code>field_name1</code>、<code>field_name2</code>、<code>field_name3</code> 和 <code>field_name4</code> 的名稱已依序指派至擷取的欄位。</p> <p>如果 CSV 剖析器必須省略某些欄位，可在清單中省略其名稱。例如，</p> <pre>fields = field_name1, , field_name3, field_name4</pre> <p>在這種情況下，剖析器僅從事件擷取第一個、第三個和第四個欄位，之後為其指派名稱 <code>field_name1</code>、<code>field_name3</code> 和 <code>field_name4</code>。</p> <p>如果欄位選項未指定記錄中的完整欄位清單，剖析器將傳回空白清單。例如，如果記錄檔案包含 <code>field1</code>、<code>field2</code>、<code>field3</code>、<code>field4</code> 和 <code>field5</code>，但僅指定了 <code>fields= field1,field2,field3</code>，剖析器將傳回空白欄位清單。</p> <p>您無法針對 CSV 剖析器使用 <code>fields=*</code>，因為剖析器會傳回空白欄位清單。您必須指定完整欄位清單，除非您需要按照如上所述省略某些欄位。</p>
delimiter	<p><code>delimiter</code> 選項指定剖析器要使用的分隔符號。依預設，csv 剖析器會使用逗點做為分隔符號；但是，您可以將分隔符號變更為分號、空格或其他特殊字元。定義的分隔符號必須用雙引號括住。</p> <p>例如，<code>delimiter=","</code> 和 <code>delimiter=";"</code>。</p> <p>csv 剖析器支援用引號括住的任何一組字元做為分隔符號，例如「 」或「asd」。記錄中欄位值的分隔符號應與分隔符號參數所定義的模式完全相符，否則剖析器將會失敗。</p> <p>對於 csv 剖析器，可以將特殊字元 (如空格或 TAB) 定義為分隔符號，只要在對應特殊字元之前加上逸出字元 (<code>\</code>)、<code>\s</code>、<code>\t</code>) 即可。例如，<code>delimiter="\s"</code> 或 <code>delimiter=" "</code>。</p> <p><code>delimiter</code> 為選用的選項。</p>

CSV 記錄剖析器組態

若要剖析從 winlog 或 filelog 來源收集的記錄，請使用下列組態。

```
[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
```

在使用此組態的情況下，從 `some_csv_logs` 來源收集的記錄 (例如，從 `directory=D:\Logs` 目錄) 將由 `myparser` 剖析。如果收集的記錄包含三個以分號分隔的值，剖析的事件會按照順序接收 `field_name1`、`field_name2` 和 `field_name3` 名稱。

剖析以下 CSV 記錄：

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

定義 CSV 剖析器組態：

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

CSV 剖析器會傳回以下欄位：

```
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.
```

一般記錄格式 (Apache) 記錄剖析器

您可以同時為 `FileLog` 和 `WinLog` 收集器設定一般記錄格式 (CLF) Apache 剖析器。

一般記錄格式 (Apache) 剖析器

預設 CLF 剖析器會定義下列欄位順序和名稱。

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

剖析器名稱：`clf`

CLF 剖析器特定的選項為 `format`。

格式選項

`format` 選項會指定產生的 Apache 記錄所採用的格式。此選項並非強制性選項。

如果未指定格式，則系統會使用下列預設一般記錄格式。

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

CLF 剖析器格式字串不接受 `regex` 運算式。例如指定空格，而非運算式 `\s+`。

若要剖析其他記錄格式，請在代理程式的組態中指定該格式。剖析的欄位會使用下列名稱顯示在伺服器端上。

備註 在需要變數的情況下，如果組態中未提供 {VARNAME}，欄位將會忽略。

欄位	值
'%a':	"remote_ip"
'%A':	"local_ip"
'%B', '%b':	"response_size"
'%C':	取決於格式中指定的變數名稱
'%c':	取決於格式中指定的變數名稱
'%D':	"request_time_mcs"
'%E':	"error_status"
'%e':	取決於格式中指定的變數名稱
'%F', '%f':	"file_name"
'%h':	"remote_host"
'%H':	"request_protocol"
'%i':	取決於格式中指定的變數名稱
'%k':	"keepalive_request_count"
'%l':	"remote_log_name"
'%L'	"request_log_id"
'%M':	"log_message" (達到此規範後，剖析器會停止剖析輸入記錄)
'%m':	"request_method"
'%n':	取決於格式中指定的變數名稱
'%o':	取決於格式中指定的變數名稱
'%p':	"server_port" 其他格式可與此指定名稱搭配使用：%(format)p。支援的格式為 "canonical"、"local" 或 "remote"。使用 "canonical" 格式時，欄位名稱會保留為 "server_port"。使用 "local" 格式時，欄位名稱將為 "local_server_port"；而使用 "remote" 格式時，欄位名稱將為 "remote_server_port"。
'%P':	"process_id" 其他格式可與此指定名稱搭配使用：%(format)P。支援的格式為 "pid"、"tid" 和 "hexid"。如果使用 "pid" 作為格式，欄位名稱將為 "process_id"，而 "tid" 和 "hexid" 格式會產生名為 "thread_id" 的欄位
'%q':	"query_string"
'%r':	"request"

欄位	值
'%R':	"response_handler"
'%s':	系統也支援產生要求的最終狀態的 "status_code"。這會在伺服器上顯示於 "status_code"。
'%t':	<p>擷取時 "timestamp" 將作為事件時間戳記使用，以結合時間戳記剖析器。若要覆寫時間戳記自動偵測，您可以在大括號中指定日期和時間格式：%{ %Y-%m-%d %H:%M:%S}t；如需詳細資訊，請參閱時間戳記剖析器。</p> <p>CLF 剖析器的時間戳記格式開頭可以是 "begin:" 或 "end:" 前置詞。如果格式以 begin: 開頭 (預設)，則系統會採用要求處理開始的時間。如果以 end: 開頭，則為寫入記錄項目的時間，這會接近要求處理結束的時間。例如，CLF 剖析器支援下列類型的格式：%h %l %u [%{begin:%d/%b/%Y %T}t. %{msec_frac}t] \"%r\" %>s %b</p> <p>CLF 剖析器的時間戳記格式指定名稱也支援下列格式權杖：</p> <p>sec</p> <p>Epoch 之後的秒數。這等同於時間戳記剖析器的 %s 規範。</p> <p>msec</p> <p>Epoch 之後的毫秒數</p> <p>usec</p> <p>Epoch 之後的微秒數</p> <p>msec_frac</p> <p>毫秒分數 (等同於時間戳記剖析器的 %f 指定名稱)</p> <p>usec</p> <p>微秒分數 (等同於時間戳記剖析器的 %f 指定名稱)</p> <p>若要剖析透過格式權杖表示時間戳記的記錄，您可以在組態中使用下列格式：</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>這些權杖無法彼此合併，或與相同格式字串中的時間戳記剖析器格式合併。您可以改為使用多個 %{format}t 權杖。例如，若要使用包含毫秒的時間戳記，除了使用時間戳記剖析器的 %f 指定名稱以外，您也可以使用下列合併的時間戳記： %{ %d/%b/%Y %T}t. %{msec_frac}t 。</p>
'%T':	"request_time_sec"
'%u':	"remote_auth_user"
'%U':	"requested_url"
'%v':	"server_name"
'%V':	"self_referential_server_name"
'%X':	"connection_status" 取決於格式中指定的變數名稱
'%x':	取決於格式中指定的變數名稱
'%I':	"received_bytes"

欄位	值
'%O':	"sent_bytes"
'%S':	"transferred_size"

例如，若要使用 CLF 剖析器剖析從 winlog 或 filelog 來源收集的記錄，請指定下列組態：

```
[filelog|clfflogs]
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
```

在使用此組態的情況下，從 clfflogs 來源 (例如從 directory=D:\Logs 目錄) 收集的記錄將由 myclf 剖析。myclf 剖析器僅剖析採用組態中所述格式產生的記錄。

對於剖析器，偵錯預設值為 debug=no。

剖析使用 CLF 產生的記錄

若要剖析使用 CLF 產生的記錄，您必須在組態中定義相應的格式。例如，

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

使用規範 %{Referer}i 和 %{User_Agent}i 的非空欄位會顯示在 vRealize Log Insight 伺服器上，各自名稱稱為 referer 和 user_agent。

整合時間戳記剖析器與 CLF 剖析器

您可以剖析包含自訂時間格式的 Apache 記錄。

存取具有如下自訂時間格式的記錄。

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

如果未指定自訂時間，CLF 剖析器會嘗試透過執行自動時間戳記剖析器自動推斷時間格式，否則將使用自訂時間格式。

錯誤記錄支援的自訂時間格式為：

自訂時間格式	說明	組態格式
%{u}t	目前時間，包括微秒	format=[%{u}t] [%l] [pid %P] [client %a] %M
%{cu}t	採用精簡 ISO 8601 格式的目前時間，包括微秒	format=[%{cu}t] [%l] [pid %P] [client %a] %M

如需支援的時間戳記規範的完整清單，請參閱[時間戳記剖析器](#)。

範例：適用於 Windows 的 Apache 預設存取記錄組態**範例：適用於 Windows 的 Apache 預設錯誤記錄組態**

此範例介紹如何設定適用於 Windows 的 Apache v2.4 存取記錄組態的格式。

```
;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\"
```

定義存取記錄格式：**1 為存取記錄格式 (httpd.conf) 設定 Apache：**

```
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 定義 CLF 剖析器組態：

```
;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 剖析器會傳回以下內容：

```
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
```

此範例介紹如何設定適用於 Windows 的 Apache v2.4 錯誤記錄組態的格式。

```
;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|clflogs-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
```

備註 提供的名稱和合併的記錄格式相對應。此外，Apache 錯誤記錄還使用上述格式設定索引鍵加以說明，而非 Apache 錯誤記錄格式。

定義錯誤記錄格式：

1 為錯誤記錄格式 (httpd.conf) 設定 Apache：

```
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 定義 CLF 剖析器組態：

```
;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
```

記錄項目：

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

CLF 剖析器會傳回記錄項目的以下欄位 (若在 +0400 時區中使用剖析器)：

```
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.
```

記錄項目：

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

CLF 剖析器會傳回記錄項目的以下欄位 (若在 +0400 時區中使用剖析器)：

```
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
```

索引鍵/值配對剖析器

您可以為 FileLog 和 WinLog 收集器設定索引鍵/值配對 (KVP) 剖析器。

索引鍵/值配對 (KVP) 剖析器

kvp 剖析器會從任意記錄訊息文字尋找並擷取所有 key=value 相符項。下列範例顯示 kvp 剖析器格式。

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

例如，索引鍵-值記錄可以採用下列格式：scope=local; abstract=false; lazyInit=false; autowireMode=0; dependencyCheck=0;

若使用 kvp 剖析器，您必須指定要從中擷取值的欄位。例如，如果組態中存在定義 fields=name,lastname,country，僅具有指定索引鍵的值會進行剖析並傳送至伺服器。

索引鍵和值都可以選擇性地以雙引號 “ ” 括住，以定義空格或其他特殊字元。

針對索引鍵或值使用雙引號時，可以將反斜線字元 「\」 用作逸出字元。反斜線字元後面的任何字元都是照字面定義的，包括雙引號字元或反斜線字元。例如：“\\”

請注意下列考量事項。

- 如果索引鍵/值配對中的索引鍵後面未跟隨等號且未提供 VALUE，即如同任意文字，將略過此選項。
- 索引鍵不可為空白，值可以空白。
- 後面未跟隨值的等號會視為任意文字，將會被略過。
- 值可以是雙引號字元括住的字元字串，也可以是空的。使用反斜線逸出值中的特殊字元。

KVP 剖析器選項

請注意下列關於 kvp 剖析器結構的資訊。

選項	說明
fields	<p>您要擷取的資訊以資料單位形式說明。例如，<code>fields=name,lastname,country</code>。</p> <p>如果以 <code>fields</code> 選項定義特定欄位名稱，則會以底線取代從記錄中擷取之欄位名稱中的每個無效字元。例如，如果 <code>fields</code> 選項尋找「x-A」和「a*(X+Y)」欄位，則剖析器會從記錄中擷取這些欄位，並分別將其重新命名為「x_a」和「a__x_y」欄位。這可讓其擷取名稱中有任何字元的欄位。</p> <p>如果將 <code>fields</code> 選項指定為「*」（這意味著 <code>kvp</code> 剖析器會自動辨識欄位/值配對），則剖析器會尋找只具有「英數字元+底線」字元（受 LI 伺服器支援）的欄位。其他所有無效字元會遭到捨棄，而非轉換為底線。這可避免剖析器將不必要的資訊擷取至靜態欄位。</p>
delimiter	<p>選擇性。</p> <p>預設分隔符號包括空格字元、Tab、新行字元、逗點和分號字元。</p> <p>如果組態中未指定任何分隔符號，<code>kvp</code> 剖析器會使用預設分隔符號進行剖析。</p> <p>若要將預設分隔符號變更為特定分隔符號，您必須在兩個雙引號之間進行定義。例如：<code>delimiter = "#^ "</code>。此定義意味著用雙引號括住的每個字元都將用作分隔符號。對於 <code>kvp</code> 剖析器，任何字元都可視為分隔符號。您可以在定義中包含預設分隔符號和其他分隔符號。</p> <p>例如，<code>delimiter = "#^ \t\r\n\s"</code> 陳述式包含 Tab、新行字元及空格做為分隔符號。如果使用這些字元，它們必須置於逸出字元之後。例如，若要將空格字元做為分隔符號，請在將空格字元定義為分隔符號時在其前面輸入逸出字元「\」，例如 <code>delimiter="\s"</code>。</p>
field_decoder	<p>巢狀剖析器指定為 JSON 字串，其中索引鍵是套用至巢狀剖析器之欄位的名稱，值是用於此欄位之剖析器的名稱。</p> <p>每個巢狀剖析器會套用至基礎剖析器解碼的適當欄位。</p> <p>當索引鍵-值配對為複雜值（例如時間戳記或以逗點分隔的清單）時，欄位解碼器非常有用。</p>
debug =	<p>選擇性。 <code>debug =</code> 值可為 <code>yes</code> 或 <code>no</code>。對於剖析器，值錯預設值為 <code>debug=no</code>。</p> <p>當選項設定為 <code>yes</code> 時，您可以在 <code>liagent_<date>.log</code> 中檢視剖析器擷取的詳細記錄。</p>

其他索引鍵值選項

索引鍵	定義
<code>KVP_MESSAGE = *(MESSAGE_ENTRY [WSPR])</code>	以選擇性空格分隔的訊息項目清單
<code>MESSAGE_ENTRY = KVP / FREE_TEXT</code>	項目為索引鍵/值配對或僅為任意文字
<code>KVP = KEY ["=" VALUE]</code>	索引鍵/值配對。如果 KEY 後面未跟隨等號和 VALUE，則會將其視為任意文字加以略過。
<code>KEY = BARE_KEY / QUOTED_KEY</code>	
<code>FREE_TEXT = "="</code>	獨立的等號會被視為任意文字且會略過。
<code>BARE_KEY = *1BARE_KEY_CHAR</code>	至少一個字元
<code>BARE_KEY_CHAR = %0x00-08 / %0x10-19 / %0x21-3C / %3E-%FF</code>	除等號、空格或 TAB 之外的任何字元
<code>QUOTED_KEY = 0x22 *1(QUOTED_STRING_CHAR / "\" CHAR) 0x22</code>	至少一個以雙引號字元括住的字元。將反斜線用作逸出字元。
<code>QUOTED_STRING_CHAR = %0x00-21 / %0x23-%FF</code>	除雙引號之外的任何字元

索引鍵	定義
VALUE = BARE_VALUE / QUOTED_VALUE	
BARE_VALUE = *BARE_VALUE_CHAR	零或多個字元
BARE_VALUE_CHAR = %0x00-08 / %0x10-19 / %0x21-FF	除空格或 TAB 之外的任何字元
QUOTED_VALUE = 0x22 * (QUOTED_STRING_CHAR / "\" CHAR) 0x22	以雙引號字元括住的字元字串。這可以為空白。將反斜線用作逸出字元。

KVP 剖析器組態範例

您可以視需要使用 `fields=*` 剖析所有欄位。

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

此範例顯示如何指定欄位解碼器。

```
[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"}
```

剖析以下 KVP 記錄：

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

定義 KVP 剖析器組態：

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

KVP 剖析器會傳回以下欄位：

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

範例：簡單和複雜的 KVP 剖析器範例

簡單的 KVP 剖析器範例

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

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

複雜的 KVP 剖析器範例

```
[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"}
```

時間戳記剖析器

timestamp 剖析器不會產生欄位，而是會將其輸入從字串轉換成以毫秒顯示的內部時間戳記格式（自 UNIX epoch 起始，即，1970 年 1 月 1 日（午夜 UTC/GMT）起始）。

唯一支援的組態選項為 format。例如，format=%Y-%m-%d %H:%M:%S。

與 CLF 剖析器不同，timestamp 剖析器可以剖析時間規範之間沒有分隔符號的時間，例如 %A%B%d%H%M%S%Y%z。

timestamp 剖析器使用的格式規範如下：

```
'%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

```

其他規範可由時間戳記剖析器接受，但其值會忽略且不會影響剖析的時間。

```

'%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

```

如果未定義 format 參數，Timestamp 剖析器會使用預設格式剖析時間戳記。

自動時間戳記剖析器

未針對時間戳記剖析器定義格式時會叫用自動時間戳記剖析器，或透過在 `field_decoder` 中使用 `timestamp` 來直接叫用剖析器，而無需定義時間戳記剖析器。例如：

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

範例：具有預設組態的時間戳記剖析器

此範例顯示具有預設組態的 `timestamp` 剖析器。

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

若要將 `timestamp` 剖析器與其他剖析器 (例如 CSV 剖析器) 整合，請指定下列組態。

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

定義此組態後，`mycsv` 剖析器會擷取具有組態中指定之名稱的欄位，並針對 `timestamp` 欄位的內容執行 `tsp_parser`。如果 `tsp_parser` 擷取了有效時間戳記，伺服器會將此時間戳記用於記錄訊息。

自動記錄剖析器

自動剖析器可自動偵測一行中前 200 個字元內的時間戳記。自動偵測之時間戳記的格式與 `timestamp` 剖析器的格式相同。

自動剖析器沒有任何選項。除了自動偵測時間戳記，索引鍵/值剖析器還可針對記錄項目執行，自動偵測記錄中的任何現有索引鍵/值配對，並相應地擷取欄位。例如，

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

對於其他剖析器，您可為自動剖析器定義單獨動作。

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

base_parser=auto
debug=yes
```

如果您已為自動剖析器啟用 debug，將會列印關於剖析的其他資訊。例如，有關自動剖析器針對哪個記錄執行以及從記錄中擷取了哪些欄位的資訊。

對於剖析器，偵錯預設值為 debug=no。

Syslog 剖析器

Syslog 剖析器支援 message_decoder 和 extract_sd 選項，並且會自動偵測兩種格式：RFC 6587、RFC-5424 和 RFC-3164。

設定 message_decoder 選項

Syslog 剖析器可使用所有一般選項和 message_decoder 選項。預設只會擷取 timestamp 和 appname 欄位。透過將 liagent.ini 檔案中的組態值設定為類似下列範例，啟用 message_decoder 選項：

```
[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=*
```

範例：透過 message_decoder 選項進行剖析

下列範例顯示範例事件，以及由設定為使用 message_decoder 選項之 Syslog 剖析器新增至事件的欄位：

■ 範例事件：

```
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"
```

■ 套用 message_decoder 選項以執行 KVP 剖析器之 Syslog 剖析器傳回的項目：

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

設定 extract_sd 選項以剖析結構化資料

若要剖析結構化資料，請透過將 liagent.ini 檔案中的組態值設定為類似下列範例，啟用 extract_sd 選項：

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

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

範例：透過 extract_sd 選項進行剖析

下列範例顯示範例事件，以及由設定為使用 extract_sd 選項之 Syslog 剖析器新增至事件的欄位：

- 範例事件：<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 剖析器會將下列欄位新增至事件：

```
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"
```

剖析器擷取的欄位

剖析器會從事件中自動擷取下列欄位：

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

Syslog 剖析器選項

下表說明可用的 Syslog 選項。

選項	說明
message_decoder	定義用於剖析事件訊息內文的其他剖析器。其可為內建剖析器 (例如「auto」) 或任何自訂定義的剖析器。
extract_sd	剖析結構化資料。 extract_sd 選項只支援 yes 或 no 值。該選項預設為停用。啟用 extract_sd 選項時，其只會從結構化資料中擷取所有索引鍵-值配對。

範例：RFC-5424 標準剖析

下列範例顯示設定之 Syslog 執行個體所剖析的兩個事件，並顯示收集器使用的組態、範例事件和 Syslog 剖析器新增至事件的欄位。

■ 組態：

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

■ 監控之檔案所產生的事件：

```
<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 剖析器新增至事件的欄位：

```
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
```

範例：RFC-3164 標準剖析

下列範例顯示收集器使用的組態、RFC-3164 範例事件和 Syslog 新增至事件的欄位。

■ 組態：

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

■ 監控之檔案所產生的 RFC-3164 事件：

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

■ Syslog 剖析器新增至事件的欄位：

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

標記定位鍵分隔值剖析器

標記定位鍵分隔值 (LTSV) 格式是定位鍵分隔值 (TSV) 的變體。

LTSV 檔案中的每個記錄會以單線表示。每個欄位會以 <TAB> 分隔，且擁有標籤和值。標籤和值會以：分隔。藉由 LTSV 格式，您可以透過使用 <TAB> (與 TSV 格式相同) 分割行來剖析每一行，並使用唯一標籤延伸任意欄位 (無須按照特定順序)。如需有關 LTSV 定義和格式的詳細資訊，請參閱 <http://ltsv.org/>。

範例：LTSV 剖析器組態

範例：LTSV 記錄範例

LTSV 剖析器不需要特定組態選項。若要使用 LTSV 剖析器，請在組態中指定內建 `ltsv` 剖析器名稱。

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

LTSV 檔案必須是符合使用 ABNF 格式之 LTSV 生產的位元組順序。

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

藉由 LTSV 組態範例，記錄的剖析應傳回下列欄位：

```
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)
```


偵錯組態

額外偵錯也適用於 LTSV 剖析器。預設會停用 LTSV 偵錯。若要關閉 LTSV 偵錯，請輸入 `debug=yes`。

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

開啟偵錯後，LTSV 剖析器會從記錄擷取所有有效標籤的值。LTSV 剖析器要求標籤名稱僅由英數字元、底線（「_」）、圓點（「.」）以及破折號（「-」）字元構成。如果記錄中至少存在一個無效標籤名稱，則其剖析將失敗。即使標籤名稱有效，代理程式仍會檢查欄位名稱。如果存在無效名稱，則應將標籤名稱修正為有效的欄位名稱。

從 filelog 區段設定 LTSV 剖析器

您也可以直接從 `filelog` 區段設定 LTSV 剖析器。

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

Regex 剖析器

`regex` 剖析器允許針對收集的資料使用一些規則運算式。

vRealize Log Insight 代理程式會使用 C++ Boost 程式庫 `regex`，並採用 Perl 語法。可以透過指定包含具名擷取群組的規則運算式模式，來定義 `regex` 剖析器。例如：`(?<field_1>\d{4}) [-] (?<field_2>\d{4}) [-] (?<field_3>\d{4}) [-] (?<field_4>\d{4})`

群組中所指定的名稱（例如：`field_1`、`field_2`、`field_3` 及 `field_4`）將成為對應擷取的欄位的名稱。名稱具有下列需求：

- 規則運算式模式中指定的名稱必須是 vRealize Log Insight 的有效欄位名稱。
- 名稱可以僅包含英數字元和底線「_」字元。
- 名稱開頭不能為數位字元。

如果提供無效的名稱，則組態會失敗。

Regex 剖析器選項

`regex` 剖析器的唯一必要選項為 `format` 選項。

可以在需要其他偵錯資訊時使用 `debug` 選項。

組態

若要建立 `regex` 剖析器，請使用 `regex` 做為 `base_parser`，並提供 `format` 選項。

範例：Regex 組態範例**範例：剖析 Apache 記錄範例**

下列範例可用於分析 1234-5678-9123-4567：

```
[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
```

結果會顯示：

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

若要使用 regex 剖析器來剖析 Apache 記錄，請為 Apache 記錄提供特定的 regex 格式：

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

結果會顯示：

```
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
```

下列程式碼顯示剖析 Apache 記錄的另一個範例。

```
[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
```

效能考量事項

與其他剖析器 (如 CLF 剖析器) 相比，regex 剖析器會耗用更多資源。如果您可以使用其他剖析器來剖析記錄，請考慮使用這些剖析器而不使用 regex 剖析器，以獲得更佳效能。

如果未提供剖析器並且您要使用 regex 剖析器，請盡量定義清晰明確的格式。下列範例顯示提供更佳效能結果的組態。此範例指定具有數位值的欄位。

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

JSON 剖析器

您可以自訂 JSON 剖析器組態，以便有選擇性地剖析 JSON 記錄。

您可以為 **FileLog** 和 **WinLog** 收集器設定逗點分隔值 (CSV) 剖析器。Log Insight 代理程式 JSON 剖析器只會剖析有效的 JSON 記錄。無效的 JSON 記錄剖析器會傳回空值結果。

預設的 JSON 剖析器組態會透過 Log Insight 代理程式來擷取 JSON 記錄中的所有欄位。當 JSON 記錄本身以單一的複雜 JSON 物件 (也可包含多個 JSON 物件) 形式來呈現時，剖析器會使用底線 (_) 字元來串連巢狀 JSON 物件和更高層 JSON 物件的名稱。這會為對應的元素產生可傳達資訊的欄位名稱。如果 JSON 記錄還包含陣列，則成員元素名稱會包含陣列名稱，且後面會加上該元素在陣列中的索引。

JSON 剖析器也會提供稱為 **fields** 的特定選項。

JSON 剖析器「fields」選項

您可以使用 **fields** 選項在組態中指定要剖析的欄位。此選項的目的是要啟用選擇性剖析 JSON 記錄的功能。

備註 若要進行選擇性剖析，您必須指定所需 JSON 元素的路徑。不同層的 JSON 物件必須使用點 (.) 字元來隔開。

下列清單提供一些可讓您視需要有選擇性地剖析 JSON 記錄的範例組態。

- 若要剖析 JSON 記錄中的多個元素，您必須將所需元素列為 **fields** 選項的參數，並將這些元素以逗點隔開。請參閱下列範例：

```
{
  "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"
  }
}
```

- 若只要剖析最內層的 JSON 物件 (如 **timestamp**、**log_severity** 和 **log_message**)，請參閱以下範例。此範例組態會產生下列欄位結果：operation_timestamp = "2018-11-22T15:28:58.094000"，且 operation_log_severity = "info"

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

- 若要剖析整個 JSON 物件，請納入物件路徑，並於後面加上星號 (*) 字元。

```
{
  "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"
  }
}
```

- 若只要剖析 **operation** 物件，請使用下列組態：

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

- 如果 JSON 記錄包含陣列，且您只想剖析陣列的特定元素，請在組態中使用陣列的元素索引，如下列範例組態所示：

```
{
  "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"
        }
    }
}
]
}

```

- 若只要剖析同一記錄中的 **key** 和 **size** 元素，請使用下列組態來產生下列欄位：

```
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

```

- 若要剖析所有陣列元素的 **key** 欄位，請使用下列組態：

```

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

```

- 若要剖析所有欄位，請使用 **fields** 選項與星號 (*) 字元。此組態相當於預設的 JSON 剖析器組態。

```

[parser|json_parser]
base_parser=json
fields=*

```

解除安裝 vRealize Log Insight 代理程式

5

如果您需要解除安裝 vRealize Log Insight 代理程式，請遵循適用於您安裝的代理程式套件的指示進行操作。

本章節討論下列主題：

- 解除安裝 Log Insight Windows Agent
- 解除安裝 Log Insight Linux 代理程式 RPM 套件
- 解除安裝 Log Insight Linux 代理程式 DEB 套件
- 解除安裝 Log Insight Linux 代理程式 bin 套件
- 手動解除安裝 Log Insight Linux 代理程式 bin 套件

解除安裝 Log Insight Windows Agent

您可以從 Windows 控制台的 [程式和功能] 畫面解除安裝 Log Insight Windows Agent。

必要條件

登入已安裝 vRealize Log Insight Windows 代理程式的 Windows 機器，然後啟動服務管理員以確認 vRealize Log Insight 代理程式服務已安裝。

程序

- 1 前往**控制台 > 程式和功能**。
- 2 選取 VMware vRealize Log Insight Windows Agent，然後按一下**解除安裝**。

結果

解除安裝程式會停止 VMware vRealize Log Insight Windows Agent 服務並從系統移除其檔案。

解除安裝 Log Insight Linux 代理程式 RPM 套件

您可以解除安裝 Log Insight Linux Agent RPM 套件。

必要條件

- 以**根使用者**身分登入，或使用 `sudo` 執行主控台命令。

- 登入安裝有 Log Insight Linux Agent 的 Linux 機器，開啟終端機主控台並執行 `pgrep liagent`，確認 VMware Log Insight Linux Agent 已安裝且正在執行。

程序

- ◆ 執行下列命令，並將 `VERSION` 和 `BUILD_NUMBER` 取代為已安裝代理程式的版本和組建編號。

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

結果

解除安裝程式會停止 VMware Log Insight Linux Agent 精靈，並從系統移除其本身記錄以外的所有檔案。

解除安裝 Log Insight Linux 代理程式 DEB 套件

您可以解除安裝 Log Insight Linux Agent DEB 套件。

必要條件

- 以**根使用者**身分登入，或使用 `sudo` 執行主控台命令。
- 登入安裝有 Log Insight Linux Agent 的 Linux 機器，開啟終端機主控台並執行 `pgrep liagent`，確認 VMware Log Insight Linux Agent 已安裝且正在執行。

程序

- ◆ 執行下列命令

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

結果

解除安裝程式會停止 VMware Log Insight Linux Agent 精靈，並從系統移除其本身記錄以外的所有檔案。

解除安裝 Log Insight Linux 代理程式 bin 套件

您可以使用 vRealize Log Insight 指令碼來解除安裝 Log Insight Linux Agent .bin 套件。

必要條件

- 以**根使用者**身分登入，或使用 `sudo` 執行主控台命令。
- 登入安裝有 Log Insight Linux Agent 的 Linux 機器，開啟終端機主控台並執行 `pgrep liagent`，確認 VMware vRealize Log Insight Linux Agent 已安裝且正在執行。

程序

- 1 在殼層提示中，輸入下列命令以啟動指令碼。

```
loginsight-agent-uninstall
```

- 2 透過檢查以下命令傳回的錯誤碼為 0，即可確認解除安裝已成功完成。

```
echo $?
```

手動解除安裝 Log Insight Linux 代理程式 bin 套件

如果您選擇不要使用解除安裝指令碼，則可以手動解除安裝 Log Insight Linux Agent .bin 套件。

必要條件

手動解除安裝 Log Insight Linux 代理程式 bin 套件

- 以**根使用者**身分登入，或使用 `sudo` 執行主控台命令。
- 登入安裝有 Log Insight Linux Agent 的 Linux 機器，開啟終端機主控台並執行 `pgrep liagent`，確認 VMware vRealize Log Insight Linux Agent 已安裝且正在執行。

程序

- 1 透過執行下列命令停止 Log Insight Linux Agent 精靈：

```
sudo service liagentd stop 或 sudo /sbin/service liagentd stop (適用於舊 Linux 發行版)。
```

- 2 手動移除 Log Insight Linux Agent 檔案

- `/usr/lib/loginsight-agent` - 精靈二進位檔與授權檔案目錄。
- `/usr/bin/loginsight-agent-support` - 用於為 Log Insight Linux Agent 產生支援服務包。
- `/var/lib/loginsight-agent` - 組態檔與資料庫儲存區目錄。
- `/var/log/loginsight-agent` - Log Insight Linux Agent 的記錄目錄。
- `/var/run/liagent/liagent.pid` - Log Insight Linux Agent PID 檔案。如果該檔案未自動刪除，請手動將其移除。
- `/etc/init.d/liagentd` - Log Insight Linux Agent 精靈的指令碼目錄。
- `/usr/lib/systemd/system/liagentd.service`

對 vRealize Log Insight 代理程式進行疑難排解

6

已知的疑難排解資訊可協助您診斷並更正與 vRealize Log Insight 代理程式作業相關的問題。

本章節討論下列主題：

- 為 Log Insight Windows Agent 建立支援服務包
- 為 Log Insight Linux Agent 建立支援服務包
- 定義 Log Insight Agents 中的記錄詳細資料層級
- 管理 UI 不顯示 Log Insight Agents
- vRealize Log Insight 代理程式不傳送事件
- 為 Log Insight Windows Agent 新增輸出例外狀況規則
- 在 Windows 防火牆中允許來自 Log Insight Windows Agent 的輸出連線
- Log Insight Windows Agent 的大量部署不成功
- Log Insight Agents 拒絕自我簽署的憑證
- vRealize Log Insight 伺服器拒絕非加密流量的連線

為 Log Insight Windows Agent 建立支援服務包

如果 Log Insight Windows Agent 因出現問題而無法如預期運作，您可以將記錄檔和組態檔的複本傳送到 VMware 支援服務。

程序

- 1 登入安裝有 Log Insight Windows Agent 的目標機器。
- 2 按一下 Windows **開始** 按鈕，然後按一下 **VMware > Log Insight 代理程式 - 收集支援服務包**。
- 3 (選擇性) 如果捷徑不可用，則導覽至 Log Insight Windows Agent 的安裝目錄，然後連按兩下 `loginsight-agent-support.exe`。

備註 預設安裝目錄為：`C:\Program Files (x86)\VMware\Log Insight Agent`

結果

產生服務包並儲存為我的文件中的 `.zip` 檔案。

後續步驟

根據要求將支援服務包轉送給 VMware 支援服務。

為 Log Insight Linux Agent 建立支援服務包

如果 Log Insight Linux Agent 因出現問題而無法如預期運作，您可以將記錄檔和組態檔的複本傳送到 VMware 支援服務。

程序

- 1 登入安裝有 Log Insight Linux Agent 的目標機器。
- 2 執行下列命令。

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

結果

產生服務包並在目前目錄中儲存為 .zip 檔案。

後續步驟

根據要求將支援服務包轉送給 VMware 支援服務。

定義 Log Insight Agents 中的記錄詳細資料層級

您可以編輯 vRealize Log Insight 代理程式的組態檔以變更記錄層級。

必要條件

對於 Log Insight Linux Agent：

- 以**根使用者**身分登入，或使用 `sudo` 執行主控台命令。
- 登入安裝有 Log Insight Linux Agent 的 Linux 機器，開啟主控台並執行 `pgrep liagent`，確認 VMware vRealize Log Insight Linux Agent 已安裝且正在執行。

對於 Log Insight Windows Agent：

- 登入已安裝 vRealize Log Insight Windows 代理程式的 Windows 機器，然後啟動服務管理員以確認 vRealize Log Insight 代理程式服務已安裝。

程序

- 1 導覽到包含 `liagent.ini` 檔案的資料夾。

作業系統	路徑
Linux	<code>/var/lib/loginsight-agent/</code>
Windows	<code>%ProgramData%\VMware\Log Insight Agent</code>

- 2 在任一文字編輯器中開啟 `liagent.ini` 檔案。

3 變更 liagent.ini 檔案之 [logging] 區段中的記錄偵錯層級。

備註 偵錯層級越高，對 vRealize Log Insight 代理程式的影響越大。預設及建議值為 0。偵錯層級 1 可提供更多資訊，建議用於疑難排解多數問題。偵錯層級 2 可提供詳細資訊。僅當 VMware 支援提出要求時才使用層級 1 和 2。

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

4 儲存並關閉 liagent.ini 檔案。

結果

記錄偵錯層級已變更。

管理 UI 不顯示 Log Insight Agents

Log Insight Agents 執行個體的相關資訊不會出現在管理 UI 的 [代理程式] 頁面上。

問題

安裝 Log Insight Agents 之後，您就不會在管理 UI 的 [代理程式] 頁面中看到 Log Insight Agents。

原因

最常見的原因是網路連線問題或 liagent.ini 檔案中 Log Insight Agents 不正確的組態。

解決方案

- ◆ 確認安裝 Log Insight Agents 所在的 Windows 或 Linux 系統已連線至 vRealize Log Insight 伺服器。
- ◆ 確認 Log Insight Agents 使用 cfapi 通訊協定。
使用 syslog 通訊協定時，UI 不會顯示 Log Insight Windows Agents。
- ◆ 檢視位於下列目錄的 Log Insight Agents 記錄檔的內容。
 - Windows - %ProgramData%\VMware\Log Insight Agent\log
 - Linux - /var/log/loginsight-agent/

尋找包含片語組態傳輸錯誤：無法解析主機名稱和解析程式失敗。此類主機不明的記錄訊息。
- ◆ 確認 liagent.ini 包含目標 vRealize Log Insight 伺服器的正確組態。請參閱[設定目標 vRealize Log Insight 伺服器](#)和[指定代理程式的目標](#)。

vRealize Log Insight 代理程式不傳送事件

不正確的組態會導致 vRealize Log Insight 代理程式無法將事件轉送到 vRealize Log Insight 伺服器。如果未正確設定一般檔案收集通道，您可能會看到類似下列內容的訊息：為通道「CHANNEL_NAME」所取得的設定無效。正確設定之前，通道「CHANNEL_NAME」將保持休眠。

問題

vRealize Log Insight 代理程式執行個體顯示在**管理 > 代理程式**頁面上，但是無事件顯示於 vRealize Log Insight 代理程式主機名稱中的 **[互動式分析]** 頁面上。未正確設定一般檔案收集通道。

原因

不正確的組態會導致 vRealize Log Insight 代理程式無法將事件轉送到 vRealize Log Insight 伺服器。

解決方案

- ◆ 定義有效的收集通道。確認是否已正確地設定一般檔案收集通道。請參閱[第 4 章 設定 vRealize Log Insight 代理程式](#)。
- ◆ 對於 vRealize Log Insight Windows 代理程式，請嘗試以下操作。
 - 如果已啟用 Windows 通道，請檢視 vRealize Log Insight Windows 代理程式記錄檔的內容，位於 `%ProgramData%\VMware\Log Insight Agent\log`。尋找與包含片語 `Subscribed to channel CHANNEL_NAME` 的通道組態相關的記錄訊息。通常會使用的通道為 `Application`、`System` 及 `Security`。
 - 如果未正確設定通道，您可能會看到類似下列內容的記錄訊息：無法訂閱通道 `CHANNEL_NAME` 事件。錯誤碼：15007。找不到指定的通道。檢查通道組態。您可能會看到非 15007 的錯誤碼號碼。
 - 如果未正確設定一般檔案收集通道，您可能會看到類似下列內容的訊息：為通道 '`CHANNEL_NAME`' 所取得的設定無效。正確設定之前，通道「CHANNEL_NAME」將保持休眠
- ◆ 對於 vRealize Log Insight Windows 代理程式和 vRealize Log Insight Linux 代理程式，請嘗試以下操作。
 - ◆ 如果未設定一般檔案收集通道，您可能會看到類似下列內容的訊息：組態中找不到區段 '`filelog`'。未正確設定之前，一般檔案記錄收集器將保持休眠

vRealize Log Insight 代理程式記錄檔的內容位於下列目錄。

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

後續步驟

如需有關設定 vRealize Log Insight 代理程式的詳細資訊，請參閱[設定 Log Insight Windows Agent](#) 和 [設定 Log Insight Linux Agent](#)。

為 Log Insight Windows Agent 新增輸出例外狀況規則

定義在 Windows 防火牆中解除封鎖 Log Insight Windows Agent 的例外狀況規則。

此程序適用於 Windows Server 2008 R2 及更新版本，以及 Windows 7 及更新版本。

必要條件

- 確保您具有管理員帳戶或擁有管理權限的帳戶。

程序

- 1 選取**開始 > 執行**。
- 2 輸入 `wf.msc`，然後按一下**確定**。
- 3 在左窗格中的**輸出規則**上按一下滑鼠右鍵，然後按一下**新增規則**。
- 4 選取**自訂**並依照精靈設定下列選項。

選項	說明
程式	<code>liwinsvc.exe</code>
服務	LogInsightAgentService
通訊協定和連接埠	TCP 9000 (用於 cfapi) 和 514 (用於 syslog)

- 5 在 [指定套用此規則的設定檔] 頁面上，選取適當的網路類型。
 - 網域
 - 公用
 - 私人

備註 您可以選取所有網路類型，以確保不論網路類型為何，例外狀況規則均處於作用中狀態。

後續步驟

前往 Log Insight Windows Agent 記錄目錄 `%ProgramData%\VMware\Log Insight Agent\log` 並開啟最新的記錄檔。如果最近事件包含訊息 `Config transport error: Couldn't resolve host name` 和 `Resolver failed. No such host is known`，則重新啟動 Log Insight Windows Agent 服務和 Windows 機器。

備註 Log Insight Windows Agent 服務最多需要 5 分鐘來重新連線到伺服器。

在 Windows 防火牆中允許來自 Log Insight Windows Agent 的輸出連線

設定 Windows 防火牆設定，以允許 Log Insight Windows Agent 至 vRealize Log Insight 伺服器的輸出連線。

您安裝並啟動 Log Insight Windows Agent 服務之後，Windows 網域或本機防火牆可能會限制與目標 vRealize Log Insight 伺服器的連線。

此程序適用於 Windows Server 2008 R2 及更新版本，以及 Windows 7 及更新版本。

必要條件

- 確保您具有管理員帳戶或擁有管理權限的帳戶。

程序

- 1 選取**開始 > 執行**。
- 2 輸入 `wf.msc`，然後按一下**確定**。
- 3 在 [動作] 窗格中，按一下**內容**。
- 4 在**網域設定檔**索引標籤上，從**輸出連線**下拉式功能表選取**允許 (預設)**。

如果電腦未連線至網域，您可以選取**私人設定檔**或**公用設定檔**，具體取決於電腦連線的網路類型。

- 5 按一下**確定**。

後續步驟

定義在 Windows 防火牆中針對 Log Insight Windows Agent 解除封鎖的例外狀況規則。請參閱 [為 Log Insight Windows Agent 新增輸出例外狀況規則](#)。

Log Insight Windows Agent 的大量部署不成功

目標機器上的 Log Insight Windows Agent 大量部署不成功。

問題

透過使用群組原則物件在 Windows 網域機器上執行大量部署之後，Log Insight Windows Agent 無法做為本機服務安裝。

原因

群組原則設定可能會阻止 Log Insight Windows Agent 的正確安裝。

解決方案

- 1 編輯群組原則物件 (GPO) 設定，並重新部署 Log Insight Windows 代理程式。
 - a 在 GPO 上按一下滑鼠右鍵，按一下**編輯**，然後導覽至**電腦設定 > 原則 > 系統管理範本 > 系統 > 登入**。
 - b 啟用**永遠在電腦啟動及登入時等待網路啟動原則**。
 - c 導覽至**電腦設定 > 原則 > 系統管理範本 > 系統 > 群組原則**。
 - d 啟用**啟動原則處理等待時間**，並將**等待時間 (秒)**設定為 120。
- 2 在目標機器上執行 `gpupdate /force /boot` 命令。

Log Insight Agents 拒絕自我簽署的憑證

Log Insight Agents 拒絕自我簽署的憑證。

問題

vRealize Log Insight 代理程式拒絕自我簽署的憑證，且無法與伺服器建立連線。

備註 如果遇到代理程式連線問題，您可以將代理程式的偵錯層級變更為 1，以產生可檢查的詳細記錄。如需詳細資訊，請參閱[定義 Log Insight Agents 中的記錄詳細資料層級](#)。

原因

代理程式記錄中出現這些訊息源於某些特定原因。

訊息	原因
拒絕對等自我簽署的憑證。公開金鑰與之前儲存的憑證金鑰不符。	<ul style="list-style-type: none"> ■ 取代 vRealize Log Insight 憑證時可能會出現此情況。 ■ 如果在 vRealize Log Insight 節點上透過不同的自我簽署憑證來設定已啟用 HA 的叢集中環境，可能會出現此情況。
拒絕對等自我簽署的憑證。具有之前接收的憑證，該憑證是由信任的 CA 所簽署。	代理程式端會儲存 CA 簽署憑證。

解決方案

- ◆ 確認目標主機名稱是否為受信任的 vRealize Log Insight 執行個體，然後從 vRealize Log Insight 代理程式 `cert` 目錄手動刪除之前的憑證。
 - 對於 Log Insight Windows Agent，請前往 `C:\ProgramData\VMware\Log Insight Agent\cert`。
 - 對於 Log Insight Linux Agent，請前往 `/var/lib/loginsight-agent/cert`。

備註 某些平台可能會使用非標準路徑來儲存受信任的憑證。Log Insight Agents 可以選擇透過設定 `ssl_ca_path=<fullpath>` 組態參數來設定受信任憑證存放區的路徑。將 `<fullpath>` 取代為受信任之根憑證服務包檔案的路徑。請參閱[設定 Log Insight Agents SSL 參數](#)。

vRealize Log Insight 伺服器拒絕非加密流量的連線

當您嘗試傳送非加密流量時，vRealize Log Insight 伺服器會拒絕與 Log Insight Agents 的連線。

您可以將 vRealize Log Insight 伺服器設定為接受非 SSL 連線，或將 Log Insight Agents 設定為透過 SSL `cfapi` 通訊協定連線傳送資料。

問題

當您嘗試使用 `cfapi` 傳送非加密流量時，vRealize Log Insight 伺服器會拒絕您的連線。代理程式記錄中會顯示下列其中一個錯誤訊息：403 Forbidden 或 403 Only SSL connections are allowed。

原因

vRealize Log Insight 設定為僅接受 SSL 連線，但 Log Insight Agents 則設定為使用非 SSL 連線。

解決方案

- 1 將 vRealize Log Insight 伺服器設定為接受非 SSL 連線。
 - a 導覽至管理索引標籤。
 - b 在 [組態] 下，按一下 **SSL**。
 - c 在 [API 伺服器 SSL] 標頭下，取消選取 **需要 SSL 連線**。
 - d 按一下 **儲存**。
- 2 將 vRealize Log Insight 代理程式設定為透過 SSL Cfapi 通訊協定連線傳送資料。
 - a 導覽到包含 `liagent.ini` 檔案的資料夾。

作業系統	路徑
Linux	<code>/var/lib/loginsight-agent/</code>
Windows	<code>%ProgramData%\VMware\Log Insight Agent</code>

- b 在任一文字編輯器中開啟 `liagent.ini` 檔案。
- c 將 `liagent.ini` 檔案 [server] 區段中的 `ssl` 索引鍵值變更為 `yes`，並將通訊協定變更為 `cfapi`。

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

- d 儲存並關閉 `liagent.ini` 檔案。