

opentext™

LoadRunner Professional

ソフトウェア・バージョン: 24.1

インストール・ガイド

ヘルプセンター・オンラインにアクセス

<https://admhelp.microfocus.com/lr/>



ドキュメント・リリース日: 2024 年 4 月

フィードバックの送信



インストール・ガイドの内容についてお気づきになった点があればお知らせください。

電子メールの宛先: admdocteam@opentext.com

ご注意

© Copyright 2024 Open Text.

Open Textおよびその関連会社およびライセンサ(「Open Text」)の製品およびサービスの保証は、当該製品およびサービスに付随する明示的な保証文によってのみ規定されるものとします。ここでの記載で追加保証を意図するものは一切ありません。ここに含まれる技術的、編集上の誤り、または欠如について、Open Textはいかなる責任も負いません。ここに記載する情報は、予告なしに変更されることがあります。

免責事項

ここからアクセス可能なソフトウェアの一部には、Hewlett-Packard Company (現在のHP Inc.) および Hewlett Packard Enterprise Companyのブランドが含まれる場合があります。本ソフトウェアは、2017年9月1日にMicro Focusによって買収され、現在は所有と経営を別とする会社OpenTextによって提供されています。HPおよびHewlett Packard Enterprise/HPEマークの使用は歴史的なものであり、HPおよびHewlett Packard Enterprise/HPEマークはそれぞれの所有者に帰属します。

目次

LoadRunner Professional へようこそ	5
LoadRunner Professional ヘルプセンター	5
LRP コンポーネント	6
インストール・パッケージのダウンロード	7
システム要件と前提条件	8
システム要件	8
LRP での TLS(SSL) 証明書の使用	8
インストール前の作業	9
Windows への LRP コンポーネントのインストール	12
インストールのワークフロー	12
LRP のアップグレード	14
セットアップ・ウィザードの使用による LRP コンポーネントのインストール	14
インストールの実行	14
セットアップ・ウィザードに関する情報	16
LRP コンポーネントのサイレント・インストール	17
インストールのコマンド・ライン・オプション	20
ラッパー・ファイルでの LoadRunner Professional フル・パッケージのインストール	20
PFTW ファイルでのスタンドアロン・アプリケーションまたは追加コンポーネントのインストール	23
LoadRunner 言語パックのインストール	25
ユーザ・ログインの設定	26
Load Generator のインストール	27
Windows での Load Generator のインストール	27
Windows での Docker 化された Load Generator のデプロイ	28
前提条件	28
定義済みイメージを使用して Docker 化された Load Generator を実行する	29
カスタム・イメージを使用して Docker 化された Load Generator を実行する	29
Dockerfile の内容のカスタマイズ例	31
Load Generator コンテナの実行後	32
ヒントとガイドライン	33
Linux でのインストールのワークフロー	33
セットアップ・ウィザードによる Linux 上での Load Generator のインストール	34
Linux での Load Generator のサイレント・インストール	36
Linux での Docker 化された Load Generator のデプロイ	37
前提条件	38
定義済みイメージを使用して Docker 化された Load Generator を実行する	38
カスタム・イメージを使用して Docker 化された Load Generator を実行する	40

Load Generator コンテナの実行後	42
ヒントとガイドライン	42
Linux 環境の設定	43
環境変数の設定	43
Linux インストールの検証	44
verify_generator の実行	45
Controller の接続の確認	46
Load Generator のパフォーマンスの改善	48
ファイル記述子を増やす	48
プロセス・エントリを増やす	49
スワップ領域を増やす	50
Linux マシンからの Load Generator のアンインストール	50
Load Generator 12.56 以降のアンインストール	50
Load Generator 12.55 以前のアンインストール	51
Linux Load Generator のインストールに関するトラブルシューティング	52
LoadRunner Data Hub と Web Connector のインストール	58
Data Hub のインストール	58
セットアップ・ウィザードによる Data Hub のインストール	59
Data Hub のサイレント・インストール	59
サイレント・インストールの実行	60
サイレント・インストールのオプション	61
Data Hub のインストールの検証	62
Web Connector のインストール	62
セットアップ・ウィザードによる Web Connector のインストール	63
Web Connector のサイレント・インストール	63
サイレント・インストールの実行	64
サイレント・インストールのオプション	65
Web Connector のインストールの検証	66

LoadRunner Professional へようこそ

OpenText™ LoadRunner Professional インストール・ガイドへようこそ。

LoadRunner Professional (LRP) は、パフォーマンス・テストのためのツールです。このツールを使用して、アプリケーションに負荷をかけ、クライアント、ネットワークおよびサーバの潜在的なボトルネックの切り分けと特定を行います。

本書では、LRP のインストールとセットアップの方法について説明します。

LoadRunner Professional ヘルプセンター

LRP コンポーネント内の状況依存ヘルプについては、ダイアログ・ボックス内で F1 キーを押すか、[ヘルプ]メニューを使用します。

Web 上の LoadRunner Professional および VuGen のヘルプセンターにアクセスできます。

- <https://admhelp.microfocus.com/lr/>
- <https://admhelp.microfocus.com/vugen/>

ヘルプセンターへのリンクを開くときは、該当する製品バージョンが選択されていることを確認してください。

ローカル・バージョンのヘルプをダウンロードしてインストールすることもできます。詳細については、「[ヘルプセンターのダウンロード](#)」を参照してください。

オンライン・モードとローカル・モードを切り替えるには、[ヘルプ]>[ヘルプセンター オプション]>[オンラインで開く]または[ローカルで開く]を選択します。

注:

- ブラウザまたは VuGen 内でヘルプセンターを表示するには、ブラウザの設定で JavaScript (アクティブ・スクリプト) が有効になっていることを確認してください。
- LRP アプリケーション・セッションの開始時にオンライン・ヘルプセンターが利用できない場合、そのアプリケーションのセッション全体で[オンラインで開く]オプションが無効のままになることがあります。

LRP コンポーネント

このインストール・ガイドでは、以下のLRP コンポーネントのインストールについて説明します。これらの各コンポーネントの詳細については、[LoadRunner Professional ヘルプセンター](#)を参照してください。

Virtual User Generator (VuGen)	主に記録によって仮想ユーザ・スクリプトを作成するためのツールです。仮想ユーザ・スクリプトは、直接関数呼び出しを使用して、グラフィカルなユーザ・インタフェースなしでユーザをエミュレートします。
Controller	仮想ユーザ・スクリプトに基づいて負荷テストとパフォーマンス・テストの実行を制御します。テスト実行に関する情報を監視して表示する、オンライン・モニタが含まれます。
Analysis	負荷テストを分析するためのグラフとレポートを表示します。
Load Generator	仮想ユーザ(Windows ベースの GUI 仮想ユーザを含む)を実行して負荷を生成するためのコンポーネントです。
Network Virtualization	NV Insights レポートを生成できます。このレポートでは、さまざまな種類のネットワーク上でアプリケーションがどのように動作するかを詳細に分析できます。 詳細については、 Network Virtualization for LoadRunner ヘルプ を参照してください。
Monitor Over Firewall	ファイアウォール越しに監視するための、エージェント・マシン上のコンポーネントです。
MI Listener	ファイアウォール越しに仮想ユーザを実行して監視するのに使用する MI Listener マシン用のコンポーネントです。
LoadRunner Data Hub	LoadRunner コンポーネント間のより堅牢な通信をサポートする通信チャネル。
Data Hub Web Connector	Data Hub が Web 経由かつファイアウォール越しに通信できるようにします。

スタンドアロン・アプリケーション

上記のコンポーネントの一部は、スタンドアロンでインストールできます。これらのコンポーネントは、インストール・パッケージの **Standalone Applications** フォルダにあります。

追加コンポーネント

追加コンポーネント (Citrix Agent など) は、インストール・パッケージの **Additional Components** フォルダにあります。詳細とインストール手順については、LoadRunner Professional ヘルプセンターの「[追加のコンポーネントとアプリケーション](#)」を参照してください。

インストール・パッケージのダウンロード

インストール・パッケージには、[無料トライアル](#)・ページからアクセスするか、[ソフトウェアのライセンスとダウンロード](#)・ページでアカウントを選択してアクセスできます。

VuGen, LoadRunner Developer, VTS, LoadRunner Data Hub などを [AppDelivery Marketplace](#) から無料でダウンロードすることもできます。

システム要件と前提条件

本章の内容

• システム要件	8
• LRP での TLS(SSL) 証明書の使用	8
• インストール前の作業	9

システム要件

次の情報は「[サポート・マトリックス](#)」にあります。

- Windows システムで LRP コンポーネントを実行するためのシステム要件
- Windows または Linux システムで Load Generator を実行するためのシステム要件
- 同じ物理マシンに同時にインストールできるコンポーネントの完全なリスト

Network Virtualization のシステム要件とインストール手順については、[Network Virtualization for LoadRunner ヘルプ](#)を参照してください。Network Virtualization は、LRP のインストール時にインストールすることも、後からインストールすることもできます。

LRP での TLS(SSL) 証明書の使用

一部の LRP コンポーネントのインストール中に、CA および TLS(SSL) 証明書をインストールするオプションがあります。

これらの証明書は、認証とセキュア通信で使用されます。どちらの証明書も「*.cer」(X.509) 形式である必要があります。

- **CA 証明書**:既存の証明書のパスを入力するか、または空白のままにして証明書のインストールを省略します。CA 証明書は事前に生成しておく必要があります。
- **TLS 証明書**:既存の証明書ファイルから選択します。または、秘密鍵を含む CA 証明書を作成しておけば、証明書は自動的に作成されます。

CA 証明書の生成と LRP での証明書の操作の詳細については、LoadRunner Professional ヘルプセンターの「[TLS\(SSL\)を使用したセキュアな通信](#)」を参照してください。

インストール前の作業

LRP コンポーネントをインストールする前に、以下の情報を確認してください。

その他の注意事項と制限事項については、LoadRunner Professional ヘルプセンターの「[既知の問題](#)」を参照してください。

セキュリティ・レビュー	LRP コンポーネントは、機密情報を含まない、または機密情報へのアクセスを提供しない専用マシンにインストールすること、テスト環境でネットワークトポロジとアクセス・レベルの徹底したセキュリティ・レビューを行うことを強くお勧めします。
アクセス権の要件	<p>すべてのLRP コンポーネントのインストール・プロセスで、すべてのオペレーティング・システムのフル管理者アカウント (Linux システムの場合はルート・アカウント) が必要になります。</p> <p>UAC と DEP を有効にして LRP コンポーネントをインストールできますが、インストール中は無効にしておくことをお勧めします (または、必須の Windows プログラムとサービスに対してのみ DEP を有効にします)。</p> <p>インストール後、LRP のすべてのアプリケーションとコンポーネントは、UAC と DEP を有効にして、標準ユーザ・アカウントで実行できます (管理者アカウントは不要)。</p>
ウイルス対策アプリケーション	LRP をインストールする前に、すべてのウイルス対策アプリケーションを終了することをお勧めします。

Windows での Web サービス仮想 ユーザの実行	<p>Axis ツールキットと WSE は、LRP では自動的にインストールされなくなりました。スタンドアロンの Load Generator で Web サービス仮想ユーザを実行する場合、一部のレガシ構成では .NET 3.5 のアクティベーション(MSDN を参照) 、Axis の Java ファイルの組み込み、または WSE のインストール(WSE 2.0 SP3 および WSE 3.0) が必要になる場合があります。</p> <p>これらのコンポーネントの使用を避けるには、スクリプトを再作成し、標準設定の WCF ツールキットを使用して WSDL をインポートします。</p> <p>これらのコンポーネントを使用するには、次の手順を実行します。</p> <ul style="list-style-type: none">• WSE の場合、WSE コンポーネントを OneLG インストール・パッケージ・フォルダ(<code>DVD\lrunner\Common\wse20sp3</code> および <code>DVD\lrunner\Common\wse30</code>) からインストールするか、Microsoft Web サイトからダウンロードします。• Axis の場合、Axis の Java ファイルを <code>%LG_PATH%\bin\java\Axis</code> にコピーします。OneLG の場合、ファイルは抽出されたインストール・パッケージの <code>prerequisites\Axis</code> サブフォルダ(標準設定では <code><LoadRunner OneLG <バージョン>のルート・フォルダ>\prerequisites\Axis</code>) にあります。これにより、Axis ツールキットでレガシ・スクリプトを実行できるようになります。 <p>詳細については、「Web サービスの既知の問題」を参照してください。</p>
Windows での Java 仮想ユーザの 実行	<p>OpenJDK 32 ビットは、LoadRunner では自動的にインストールされなくなりました。Java プロトコルを使用するには、サポートされているバージョンの JDK を手動でインストールする必要があります。</p> <p>VuGen は、内部ロジックを使用して自動的に JRE のバージョンを検出します。詳細については、「Java プロトコル用の環境のセットアップ」を参照してください。</p>
LRP と UFT の共存	<p>LRP と UFT を同じマシンにインストールしてから、一方をアンインストールすると、もう一方のソフトウェア・プログラムが正しく機能しない場合があります。</p> <p>解決策:マシンで使用するソフトウェア・プログラムの修復を実行します。</p>

ライセンス	<p>LRP ライセンスに関しては、次の点に注意してください。</p> <ul style="list-style-type: none">• Controller で仮想ユーザを実行するには、Controller マシンに有効なライセンスがインストールされている必要があります。LRP は、LRPのセットアップ時にコミュニティ・ライセンス・バンドル(無料)を自動的にインストールし、50人の仮想ユーザを提供します。• 2020より前のバージョンからアップグレードする場合は、アップグレードしたLRPインストールで動作するように現在のライセンスを置き換える必要があります。コミュニティ・ライセンスを使用している場合、LRPのインストール時に自動的に置き換えられます。その他のライセンスについては、ライセンス・サポートに連絡して、互換性のある有効なライセンスを取得してください。 <p>LRPライセンスの管理と仮想ユーザ容量の拡張の詳細については、LoadRunner Professional ヘルプセンターの「LoadRunner Professional のライセンスについて」を参照してください。</p>
カスタム証明書	<p>12.55以前のバージョンからアップグレードする場合は、以前のLoadRunnerのインストールでカスタム証明書を使用した場合、セットアップ中に証明書の入力を求められたら、もう一度その証明書をインストールします。インストールしない場合、セットアップ・プログラムは標準設定を使用してそれらを上書きします。</p>
Network Virtualization	<p>NVを(LRPインストールの一部として)自動的にインストールすることを選択した場合、NVインストールを続行する前にWindows SmartScreenを無効にする必要があります。</p> <p>レジストリ・エディタでHKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorerを開き、SmartScreenEnabledの値データをOffに変更します。</p> <p>注:NVを手動でインストールする場合、SmartScreenを無効にする必要はありません。</p>
仮想環境	<p>仮想化ベンダが提供するアーキテクチャは急速に進化しています。サードパーティ・ベンダが仮想環境とLRP承認済みハードウェア要件との完全な互換性を保証する限り、LRPはこうした変化する環境でも設計通りに動作することが期待されます。システム要件とサポート・マトリックスに従って仮想マシンを作成すれば、LRPは正常に動作します。</p> <p>仮想マシン上で作業する場合、必要に応じて仮想化サーバのハードウェア/監視環境にアクセスし、仮想化サーバが飽和していないことを確認する必要があります。仮想化サーバが飽和すると、仮想マシンの測定値を正しく認識できず、間違った結果につながる可能性があります。</p>

Windows への LRP コンポーネントのインストール

本章では、LRP のフルバージョンまたは LRP コンポーネントを Windows プラットフォームにインストールする方法について説明します。

本章の内容

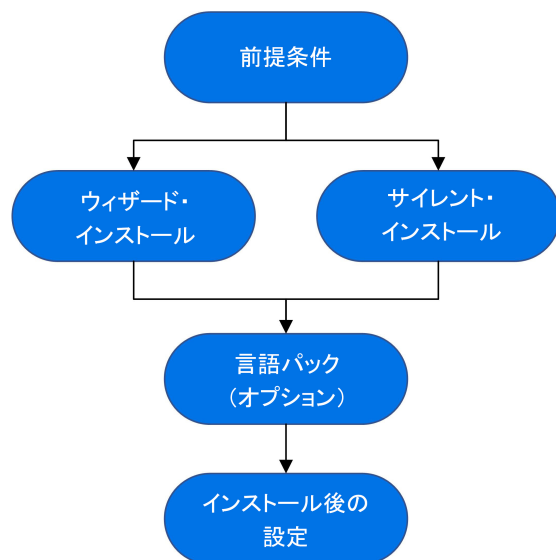
• インストールのワークフロー	12
• LRP のアップグレード	14
• セットアップ・ウィザードの使用による LRP コンポーネントのインストール	14
• LRP コンポーネントのサイレント・インストール	17
• インストールのコマンド・ライン・オプション	20
• LoadRunner 言語パックのインストール	25
• ユーザ・ログインの設定	26

インストールのワークフロー

LoadRunner Professional インストール・パッケージには、LRP のコンポーネントのインストール・プロセスを順を追って実行するためのセットアップ・ウィザードが収録されています。

このインストールの最終段階で、Network Virtualization をオプションでインストールできます。

Windows システムに LRP をインストールするには、以下の手順で行います。



1. 前提条件

- ご使用のシステムがハードウェアおよびソフトウェアの要件を満たしていることを確認します。詳細については、[「システム要件」\(8ページ\)](#)を参照してください。
- 次のセクションの情報を確認します。[「インストール前の作業」\(9ページ\)](#)
- インストールは目的のマシンで実行されなければなりません。LRP では、ターミナル・サービス経由でのインストールはサポートされません。

2. セットアップを実行する

- **ウィザード・インストール:** セットアップ・ウィザードを使用してインストールを実行するには、[「セットアップ・ウィザードの使用による LRP コンポーネントのインストール」\(14ページ\)](#)を参照してください。
- **サイレント・インストール:** サイレント・インストールの実行方法の詳細については、[「LRP コンポーネントのサイレント・インストール」\(17ページ\)](#)を参照してください。

セットアップを実行した後、必要に応じてパッチのインストールを実行します。

3. 言語パックをインストールする(オプション)

LRP のインストールが完了すると、ローカライズされたバージョンをインストールして、LRP, Analysis スタンドアロン、および VuGen のユーザ・インタフェースをユーザの言語で表示できるようになります。詳細については、[「LoadRunner 言語パックのインストール」\(25ページ\)](#)を参照してください。

4. インストール後の設定の完了

LRP は、ユーザが手動でログインしなくても Load Generator マシンで仮想ユーザを実行するように設定できます。詳細については、「[ユーザ・ログインの設定](#)」(26ページ)を参照してください。

LRP のアップグレード

LRP は、任意の LoadRunner または LRP インストール(バージョン 12.5x 以降)にインストール可能なフル・インストールです。

セットアップ・ウィザードを使用して Windows インストールのすべてのコンポーネントをアップグレードするには、「[セットアップ・ウィザードの使用による LRP コンポーネントのインストール](#)」(14ページ)で説明されているインストール手順に従います。インストール・プロセスで旧バージョンが検出されると、アップグレードするかどうかを選択できます。

サイレント・アップグレードについては、「[LRP コンポーネントのサイレント・インストール](#)」(17ページ)を参照してください。

セットアップ・ウィザードの使用による LRP コンポーネントのインストール

このセクションでは、セットアップ・ウィザードを使用して、LRP のフル・バージョン、スタンドアロン・コンポーネント、または追加コンポーネントをインストールする方法について説明します。

インストールの実行

Windows のインストール手順を次に示します。

LRP または追加コンポーネントをインストールするには、次の手順で行います。

1. マシンで Windows Update プロセスが実行されていないことを確認します。
2. インストール・パッケージのルート・フォルダにある **setup.exe** を実行します。

LRP インストール・プログラムが起動し、インストール・オプションが表示されます。

3. 必要なインストール・オプションを選択します。利用可能な各コンポーネントの詳細については、[「LRP コンポーネント」\(6ページ\)](#)を参照してください。

負荷テスト・シナリオを実行するマシンの場合は、**[LoadRunner Professional フルセットアップ]**を選択します。フル・セットアップには、次のコンポーネントが含まれています。

- VuGen
- Controller
- Analysis
- OneLG Load Generator
- Network Virtualization(任意)

注: LoadRunner Professional セットアップ・ウィザードでは、LRP または LoadRunner Enterprise(LRE) のいずれかのインストールが可能です。必ず **LoadRunner Professional** を選択してください。

4. LRP をインストールする前に、Windows に特定のソフトウェアをインストールする必要があります([サポート・マトリックス](#)を参照)。必須ソフトウェアがまだコンピュータにインストールされていない場合は、必須プログラムの一覧を表示するダイアログ・ボックスが開きます。

LRP のインストールを続行する前に一覧のソフトウェアをインストールするには、**[OK]**をクリックします。**[キャンセル]**をクリックすると、必須のソフトウェアがない状態では LoadRunner Professional をインストールできないため、LRP セットアップ・ウィザードは終了します。

必須ソフトウェアをインストールした後、LRPがマシンを再起動する場合があります。その場合は、**setup.exe** を再実行します。

5. インストールを実行します。

セットアップ・ウィザードが開き、ようこそページが表示されます。ウィザードの指示に従いインストールを完了します。

特定のコンポーネントのインストール中に、LoadRunner エージェントで使用する CA および TLS(SSL) 証明書をインストールするオプションがあります。これらの証明書の詳細と要件については、[「LRP での TLS\(SSL\) 証明書の使用」\(8ページ\)](#)を参照してください。

注:

- 標準設定のインストール・パスは次のとおりです。 **C:\Program Files (x86)\OpenText\LoadRunner**
- インストール・パスには英語以外の文字を含めることはできません。

6. インストールが完了したら、Network Virtualization (NV) のインストールを選択できません。

セットアップ・ウィザードで、次のいずれかを選択します。

- **標準モード**: 自動インストールを行う場合。
- **カスタム・モード**: インストール・フォルダ、データ・フォルダ、およびポートを変更する場合。

詳細については、[Network Virtualization for LoadRunner ヘルプ](#)のインストールの項を参照してください。

7. インストールしたバージョンに利用可能なパッチがある場合:

- a. インストール・パッケージのパッチ・インストール・ファイルを実行します。
- b. 画面の指示に従います。

注:

- インストールを変更、修復、または削除 (アンインストール) するには、LoadRunner Professional インストール・パッケージのルート・ディレクトリにある **setup.exe** ファイルを実行します。[**LoadRunner Professional フル セットアップ**]を選択し、[設定]ダイアログで関連する操作を選択します。
- **DLL** や **OCX** ファイルなど、セットアップ中に登録したコンポーネントのリストについては、インストールの **build_info** フォルダ内の **RegisteredComponents_OpenText LoadRunner CE 24.1.txt** ファイルを参照してください。

セットアップ・ウィザードに関する情報

セットアップ・ウィザードを使用して LRP コンポーネントを正しくインストールできるように、次の内容を確認してください。各コンポーネントの詳細については、[LoadRunner Professional ヘルプセンター](#)を参照してください。

LoadRunner エージェント	<p>[インストール後に LoadRunner エージェントを開始する] オプションを選択して、インストール直後に Load Generator で LoadRunner エージェントを開始します。</p> <p>これにより、エージェントで Load Generator と Controller との間の通信が可能になります。</p>
エージェント・モード	<p>OneLG Load Generator, MI Listener, Monitor Over Firewall のいずれかのコンポーネントのインストール中に、インストールする LoadRunner エージェントのモードを選択するように求められます。</p> <ul style="list-style-type: none">• LoadRunner Professional モード: エージェントは通常のプロセスとして実行され、プロセスを起動するにはログインが必要となります。 Business Process Monitor (BPM) で使用するために Load Generator をインストールする場合は、このモードを選択します。• LoadRunner Enterprise モード: エージェントはサービスとして実行され、オペレーティング・システムの起動時に自動的に起動されます。• LoadRunner Cloud モード: このモードは OneLG インストールの場合にのみ適用されます。LoadRunner Cloud で Load Generator を使用する場合に選択します。 <p>エージェントの設定は、インストール後に必要に応じて変更できません。</p>
Silk Performer エージェント	<p>Silk Performer スクリプトの実行に使用する Load Generator に Silk Performer エージェントをインストールするには、[インストール後に Silk Performer エージェントをインストールする] オプションを選択します。</p>

LRP コンポーネントのサイレント・インストール

サイレント・インストールは、ユーザとのやり取りを必要とせずに実行されるインストールです。コマンド・ラインを使用してセットアップ・ファイルを実行します。詳細については、「[インストールのコマンド・ライン・オプション](#)」(20ページ)を参照してください。

注:

- インストールのプロパティを定義するには、標準 MSI コマンド・ライン・オプションを使います。たとえば、別のインストール・フォルダを指定するには、INSTALLDIR を使います。

- Load Generator のインストール直後に LoadRunner エージェントを開始しないようにするには、コマンド・ラインに START_LGA=0 コマンドを追加します。これにより、エージェントで Load Generator と Controller との間の通信が可能になります。LoadRunner エージェントの詳細については、[LoadRunner Professional ヘルプセンター](#)を参照してください。

LRP のサイレント・インストールを行うには、次の手順を実行します。

1. コマンド・ラインから次のいずれかのコマンドを実行します。

- 単一のコマンドで、必須ソフトウェアを含む LRP コンポーネントをすべてインストールするには、次の手順を実行します。

```
<Installation_package>\lrunner\<<your_language_folder>\setup.exe /s
```

- 最初にすべての必須ソフトウェアをインストールしてから LRP をインストールするには、次のコマンドを実行します。

```
<Installation_package>\lrunner\<<language_folder>\setup.exe /InstallOnlyPrerequisite /s
```

```
msiexec.exe /qn /i "<Installation_package>\lrunner\MSI\LoadRunner_x64.msi"
```

- 標準設定では、LRP のインストール中に Network Virtualization がインストールされます。このインストールを容易にするため、インストール・コマンドに次のオプションを追加します。

```
REBOOT_IF_NEED=1
```

- インストール・プログラムで Network Virtualization がインストールされないようにするには、インストール・コマンドに次のオプションを追加します。

```
NVINSTALL=N
```

- OneLG インストールの一部として Gatling または JMeter をインストールするには、インストール・コマンドに次のオプションを追加します。

Gatling:

```
INSTALL_GATLING=1
```

JMeter:

```
INSTALL_JMETER=1
```

- LoadRunner 製品, OneLG, MiListener, または MoFW の TLS 証明書や双方向 TLS で環境を保護するには, インストール・コマンドに次の該当するオプションを追加します。

```
CACERT = <*.cer 形式のルート証明書のパス>  
CERT = <*.cer 形式のサーバ証明書のパス>  
CACERT_PK = <CA 証明書の秘密鍵>  
CACERT_PK_PWD = <CA 証明書の暗号化された秘密鍵のパスワード>  
CERT_PK = <TLS 証明書の秘密鍵>  
CERT_PK_PWD = <TLS 証明書の暗号化された秘密鍵のパスワード>  
TWO_WAY_SSL = High | Medium
```



例: 次のコマンドは, MI Listener のサイレント・インストールを実行します。

```
SetupMIListener.exe /s /a /s INSTALLDIR="C:\Program Files  
(x86)\OpenText\MI Listener" IS_RUNAS_SERVICE=0 START_LGA=1  
NVINSTALL=N CACERT=c:\temp\root.cer CERT=server.cer TWO_WAY_  
SSL=Medium
```

証明書の要件については, 「[LRP での TLS\(SSL\) 証明書の使用](#)」(8ページ)を参照してください。

- LRP スタンドアロン・アプリケーション(VuGen または OneLG 以外)をインストールするには, 次のコマンドを実行します。

```
<Installation_package>\Standalone  
Applications\Setup<component_name>.exe /s /a /s
```

- VuGen をインストールするには, 次のコマンドを実行します。

```
<Installation_package>\Standalone Applications\SetupVuGen.exe  
-s -sp"/s"
```

- 追加コンポーネントをインストールするには、次のコマンドを実行します。

```
<Installation_package>\Additional Components\<setup_file_path>  
/s /a /s
```

ヒント: その他の例については、[「例」\(24ページ\)](#)を参照してください。

2. アップグレードをインストールする場合は、次のコマンドを実行します。

```
msiexec.exe /update <msp ファイルのフル・パス> [/qn] [/l*vx <ログ・ファイルのフル・パス>]
```

msp ファイルは、インストール・パッケージ内にあります。

詳細については、[「LRP のアップグレード」\(14ページ\)](#)を参照してください。

インストールのコマンド・ライン・オプション

Windows のコマンド・ラインを使用して、LoadRunner Professional のフル・パッケージ、スタンドアロン・アプリケーション、および追加コンポーネントをインストールできます。CLI では、ラッパー・ファイル(setup.exe) または Web ファイルのパッケージ(<PFTW>.exe) を使用します。

- LRP のフル・インストールは、 setup.exe ファイルを使用して起動できます。詳細については、[「ラッパー・ファイルでの LoadRunner Professional フル・パッケージのインストール」\(20ページ\)](#)を参照してください。
- スタンドアロン・アプリケーションおよび追加コンポーネントのインストールは、 PFTW ファイルを介して起動できます。詳細については、[「PFTW ファイルでのスタンドアロン・アプリケーションまたは追加コンポーネントのインストール」\(23ページ\)](#)を参照してください。

ラッパー・ファイルでの LoadRunner Professional フル・パッケージのインストール

LoadRunner Professional インストール・パッケージの \runner\<使用言語> フォルダにあるインストール・ラッパー・ファイル(Setup.exe) で、LRP のフル・パッケージをインストールできま

す。次のコマンド・ライン・オプションが利用できます。

オプション	説明
/s	ユーザ操作なしにバックグラウンドで(サイレント) インストールを実行します。
/qb	ユーザ操作が制限された無人モードでインストールを実行します。
/InstallOnlyPrerequisite	<p>必須ソフトウェアのみをインストールします。LRP のコンポーネントはインストールされません。</p> <p>標準設定で、セットアップ・プログラムは LRP のコンポーネントをインストールする前に、マシンに必須ソフトウェアがインストールされていることを確認し、必要に応じて必須ソフトウェアをインストールします。</p>

次の構文を使用して、コマンド・ライン・インストール用のパブリック・プロパティを設定できます。

```
setup.exe PROPERTY_NAME="value"
```

次のプロパティが利用できます。

プロパティ名	説明
INSTALLDIR="your_path"	アプリケーションをインストールする場所を指定します。
REBOOT_IF_NEED	<p>1: 必要に応じて、インストール後にマシンを再起動します。NV(Network Virtualization) のインストールを含める場合は、このオプションを指定することをお勧めします。</p> <p>0: インストール後にマシンを再起動しません。</p> <p>標準設定:0</p>
NVINSTALL	<p>空の文字列: NV コンポーネントのインストールを含めません。</p> <p>Y:NV コンポーネントのインストールを含めます。</p> <p>標準設定:Y</p>

プロパティ名	説明
INSTALL_NV_MODE	1: NV インストール・モードを標準に設定します。 2: NV インストール・モードをカスタムに設定します。 標準設定:標準
INSTALL_DIR="your_path"	インストールのパスを指定します。
START_LGA	空の文字列: インストール後に Load Generator を起動しないようにマシンに指示します。 1:インストール後に Load Generator を起動するようにマシンに指示します。 標準設定:1
IS_RUNAS_SERVICE	0: LoadRunner Professional モード。Load Generator のエージェントをプロセスとして実行します。 1: LoadRunner Enterprise モード。Load Generator のエージェントをサービスとして実行します。 2: (OneLG にのみ関連) LoadRunner Cloud モード。Load Generator のエージェントをサービスとして実行し、インストールの最後に LRC エージェントの設定の起動をトリガします。 標準設定:1
INSTALL_SILK_PERFORMER_AGENT	1: Load Generator に Silk Performer エージェントをインストールします(Silk Performer スクリプトの実行に使用)。
INSTALL_GATLING	0:Load Generator に Gatling をインストールしません。 1:Load Generator に Gatling をインストールします。 標準設定:0
INSTALL_JMETER	0:Load Generator に JMeter をインストールしません。 1:Load Generator に JMeter をインストールします。 標準設定:0

プロパティ名	説明
SKIP_CHECK_PRIVILEGE	0: インストール・プロセスで、インストール・パスへのアクセス権限があるかどうかチェックされます。 1: インストール・プロセスで、インストール・パスのアクセス権限のチェックがスキップされます。 標準設定:0
IMPROVEMENTPROGRAM	0: LoadRunner 改善プログラムを無効にします。 1: LoadRunner 改善プログラムを有効にします。 標準設定:1

例:

次のコマンドは、サイレント・インストールを実行した後に、必要に応じてマシンを再起動します。

```
DVD\lrunner\en\setup.exe /s REBOOT_IF_NEED="1"
```

PFTW ファイルでのスタンドアロン・アプリケーションまたは追加コンポーネントのインストール

LRP スタンドアロン・アプリケーションまたはいずれかの追加コンポーネントを、LoadRunner Professional インストール・メディアに含まれる Web 用パッケージ (PFTW) ファイルを介してインストールできます。インストール・ファイルは、インストール・メディアの **Standalone Applications** または **Additional Components** フォルダ内にあります。

パッケージ・ファイルをダブルクリックするか、コマンド・ラインから次のオプションを指定して実行します。

オプション	説明
/s	ユーザ操作なしにバックグラウンドで(サイレント)インストールを実行します。
/e	インストール・ファイルの展開のみを行います。インストール・ファイルの実行は行われません。

オプション	説明
/f	ファイルを展開するための一時フォルダのパスを指定します。 例: /f "c:\my_temp_folder" このオプションを省略した場合、標準設定の一時フォルダが使用されます。
/a	パラメータやプロパティを、setup.exe などの自動実行ファイルに渡すことができます。 Setup.exe ファイルに対して、上記 (ラッパー・ファイル・セクション) のプロパティを使用します。

注: VuGen と OneLG のサイレント・インストール:

- **SetupVuGen.exe** と **SetupOneLG.exe** は、使用するインストール・コマンドが異なります(下の例を参照)。
- コマンド・ライン・パラメータに引用符文字を挿入する必要がある場合は、2つの連続した引用符を使用します。

例

次のコマンドにより、セットアップがサイレントで実行され、指定したフォルダにアプリケーションがインストールされます。必要に応じて、インストール後にエージェントも起動されます。

- **SetupVuGen.exe** と **SetupOneLG.exe** の例:

```
SetupOneLG.exe -s -sp"/s INSTALLDIR=""C:\Program Files  
(x86)\OpenText\LoadRunner OneLG"" IS_RUNAS_SERVICE=0 START_LGA=1  
NVINSTALL=N"
```

```
SetupVuGen.exe -s -sp"/s INSTALLDIR=C:\VuGen"
```

- 他のスタンドアロン・アプリケーションの例:

```
SetupAnalysis.exe /s /a /s INSTALLDIR="C:\Analysis"
```



```
SetupMIListener.exe /s /a /s INSTALLDIR="C:\MiListener"
```

```
SetupMoFW.exe /s /a /s INSTALLDIR="C:\MoFW"
```

LoadRunner 言語パックのインストール

言語パックを使用すると、LRP のユーザ・インタフェースをユーザの言語で表示できるようになります。関連する LRP コンポーネント (LoadRunner Professional フル、VuGen スタンドアロン、または Analysis スタンドアロン) の言語パックを、LoadRunner Professional インストール・パッケージからインストールします。

サポートされる言語のリストについては、[サポート・マトリックス](#)を参照してください。

言語パックは、英語以外のネイティブ・オペレーティング・システムでのみサポートされています。

重要: オペレーティング・システムのネイティブ言語は、インストールする言語パックと同じである必要があります。たとえば、スペイン語の言語パックは、スペイン語のネイティブ・オペレーティング・システムが稼働するマシンにインストールする必要があります。

言語パックをインストールするには、次の手順を実行します。

1. LRP の英語版がインストールされていることを確認します。
2. LoadRunner Professional インストール・パッケージのルート・フォルダにある **setup.exe** を実行します。LRP インストール・プログラムが起動し、インストール・オプションが表示されます。
3. **[言語パック]** をクリックします。インストール・パッケージの [Language Packs] フォルダが開きます。
4. インストールする言語およびコンポーネントのフォルダに移動し、インストール・ファイルを実行します。たとえば、フランス語の言語パックを VuGen スタンドアロンのマシンにインストールする場合、移動先のパスは **..\Language Packs\French\VugenSA\Vugen_FRA.msp** です。
画面の指示に従ってインストールを行います。

ユーザ・ログインの設定

標準設定では、コンピュータで LRP を使って仮想ユーザを実行するために、まずそのコンピュータに手動でログオンする必要があります。ただし LRP は、マシンに手動でログオンしなくても、Load Generator マシンで仮想ユーザを実行するように設定できます。

ユーザ・ログインの設定は、次の手順で行います。

1. Windows の[スタート]メニューで[**LoadRunner エージェント**]を検索し、
[**LoadRunner Agent Runtime Settings Configuration**]を選択します。
[LoadRunner エージェント の実行環境設定]ダイアログ・ボックスが開きます。
2. 次のオプションのいずれかを選択します。
 - [**ユーザ ログインなしでこのマシン上で仮想ユーザの実行を許可する**]: LRP は、Load Generator マシンからネットワークに自動的にログオンするため、仮想ユーザを手動による操作なしで実行できます。ユーザのコンピュータが所属するネットワーク・ドメインとユーザ名およびパスワードを入力します。

注: 作成されると、(指定したユーザ・アカウントではなく) **LocalSystem** アカウントで LoadRunner エージェント サービスが開始されます。スクリプトを実行して **mdrv.exe** プロセスを開始する場合、指定された資格情報が Agent サービスで使用されます。

自動ログオンを有効にするには、LRP をインストールした後に、最低 1 回はシステムを手動で再起動してログインする必要があります。

- **このマシンへの手動ログイン:** ユーザは、仮想ユーザを実行するセッションごとに、Load Generator マシンからネットワークに手動でログオンする必要があります。
3. [**OK**]をクリックします。

Load Generator のインストール

Load Generator は、仮想ユーザ・スクリプトを実行して負荷を生成するマシンです。Load Generator を作成するには、ホスト・コンピュータに LoadRunner の Load Generator をインストールします。Load Generator は、Windows ベースか Linux ベースのいずれかになります。Windows ベースの LoadRunner Controller を使用して、Windows ベースと Linux ベースの両方の仮想ユーザを制御します。

UI ベースのセットアップ・ウィザードを使用して Load Generator をインストールすることも、サイレント・インストールを実行することも、Docker コンテナを使用して Load Generator をデプロイすることもできます。または、クラウド・アカウントで Load Generator をプロビジョニングすることもできます。詳細については、LoadRunner Professional ヘルプセンターで「[クラウド上の Load Generator](#)」を参照してください。

本章の内容

- [Windows での Load Generator のインストール](#) 27
- [Windows での Docker 化された Load Generator のデプロイ](#) 28
- [Linux でのインストールのワークフロー](#) 33
- [セットアップ・ウィザードによる Linux 上での Load Generator のインストール](#) 34
- [Linux での Load Generator のサイレント・インストール](#) 36
- [Linux での Docker 化された Load Generator のデプロイ](#) 37
- [Linux 環境の設定](#) 43
- [Load Generator のパフォーマンスの改善](#) 48
- [Linux マシンからの Load Generator のアンインストール](#) 50
- [Linux Load Generator のインストールに関するトラブルシューティング](#) 52

Windows での Load Generator のインストール

Windows に Load Generator をインストールするには、他の LRP コンポーネントと同じインストール・プロセスを使用します。詳細については、「[Windows への LRP コンポーネントのインストール](#)」(12ページ)を参照してください。

Docker コンテナを使用して Windows に Load Generator をデプロイするには、「[Windows での Docker 化された Load Generator のデプロイ](#)」(28ページ)を参照してください。

Windows での Docker 化された Load Generator のデプロイ

本項では、Windows プラットフォームで Docker 化された Load Generator を実行する方法について説明します。Docker は、コンテナを介してアプリケーションを開発、出荷、および実行できるプラットフォームです。

Docker 化された Load Generator でサポートされているプロトコルについては、『[Supported Protocols](#)』ガイドを参照してください。

注: FTP は、Windows 上の Docker の Load Generator では完全にはサポートされていません。パッシブ・モードの FTP はサポートされていますが、アクティブ・モードの FTP はサポートされていません。

前提条件

- Docker をその依存関係とともにターゲット・マシンにインストールし、必要に応じてターゲット・マシン環境を設定します。現在のところ、64 ビット・バージョンのみがサポートされています。インストールの詳細については、Docker のオンライン・ドキュメントを参照してください。
- Docker Hub の「[performancetesting](https://hub.docker.com/u/performancetesting)」ページ (<https://hub.docker.com/u/performancetesting>) からアクセスできる関連ページから、Windows Load Generator の Docker イメージをダウンロードします。次のコマンドと適切な **<タグ・バージョン番号>** (23.1 など) を使用します。

```
docker pull performancetesting/opentext_one1g_windows:<tag version number>
```

注: OneLG Load Generator 用の Docker イメージは、以前の Windows スタンドアロン Load Generator の Docker イメージを置き換えるものです。

定義済みイメージを使用して Docker 化された Load Generator を実行する

既製のイメージを使用して、Docker for Windows で Load Generator (OneLG) を実行します。Java の場合や特定ユーザで実行する場合などで、コンテナのカスタマイズが必要な場合は、「[カスタム・イメージを使用して Docker 化された Load Generator を実行する](#)」(29 ページ)を参照してください。

Docker 化された Load Generator を実行するには、次の手順を実行します。

次のコマンドを使用して、Load Generator コンテナを実行します。

```
docker run -id -p <host_port>:54345 performancetesting/opentext_
onelg_windows:<tag version number>
```

注: マシン上で <host_port> が使用可能で、受信要求が許可されていることを確認します。この Load Generator に接続する場合は、Controller 側でこのポートを指定します。

カスタム・イメージを使用して Docker 化された Load Generator を実行する

お使いの環境でカスタマイズした設定を使用してコンテナを実行する必要がある場合は、Dockerfile を作成して Docker for Windows 用のカスタム・イメージをビルドできます。

カスタム・イメージの例:

- 仮想ユーザを実行するプロセスに特定のユーザ・アカウントを使用し、スクリプト・パラメータ・ファイルなどのネットワーク・リソースへのアクセス・サポートを提供する。実行後にコンテナでユーザを検証できる必要があります。
- Windows Load Generator のコンテナで Java プロトコルを実行する。
- プロキシ・サーバ・ホストとポートの環境変数を定義する。

カスタム・イメージを使用して Docker 化された Load Generator を実行するには、次の手順を実行します。

1. 新しいフォルダを作成し、その中に **dockerfile** という名前のファイルを作成します。このファイルに次の **FROM** 行を貼り付け、**<tag version number>** で LoadRunner Professional の該当するバージョンを指定し、関連するカスタマイズ行を追加します。

```
FROM performancetesting/opentext_one1g_windows:<tag version number>
<Customization lines>
```

カスタマイズの例については、「[Dockerfile の内容のカスタマイズ例](#)」(31ページ)を参照してください。

ヒント: Docker ファイルで使用できるコマンドについては、Docker のオンライン・ドキュメントを参照してください。

2. Dockerfile を保存します。
3. **dockerfile** フォルダ・パスでコマンド・ラインを開き、使用するカスタム・イメージの名前を指定して、次のコマンドを実行します。

```
docker build -t <custom image name> .
```

4. 次のコマンドを実行して、使用する Load Generator ごとにコンテナを1つずつ作成します(または、いずれかの Docker オーケストレータ・ツールを使用してコンテナを実行します)。

```
docker run -id -p <host_port>:54345 <custom image name>
```

ステップ 3 でタグを指定してカスタム・イメージをビルドしている場合は、コマンド内でタグを指定します。

```
docker run -id -p <host_port>:54345 <custom image name>:<tag version number>
```

注: マシン上で <host_port> が使用可能で、受信要求が許可されていることを

確認します。この Load Generator に接続する場合は、Controller 側でこのポートを指定します。

Dockerfile の内容のカスタマイズ例

指定したユーザ・アカウントで仮想ユーザを実行する例

以下に示すのは、共有の場所へのネットワーク・アクセスを持つ指定したユーザ・アカウントで仮想ユーザを実行するための Dockerfile の例です。<> で囲まれた値を、共有リソースへのネットワーク・アクセスを持つ、お使いの環境の有効なユーザ・アカウントの資格情報に置き換えます。



例:

```
#escape=`  
FROM performancetesting/opentext_onelg_windows:<tag version  
number>  
RUN c:\LG\launch_service\bin\magentservice.exe -remove  
RUN c:\LG\launch_service\bin\magentservice -install  
<domain>\<user name> <password>
```

Java/JMeter/Gatling プロトコルの実行に関する例

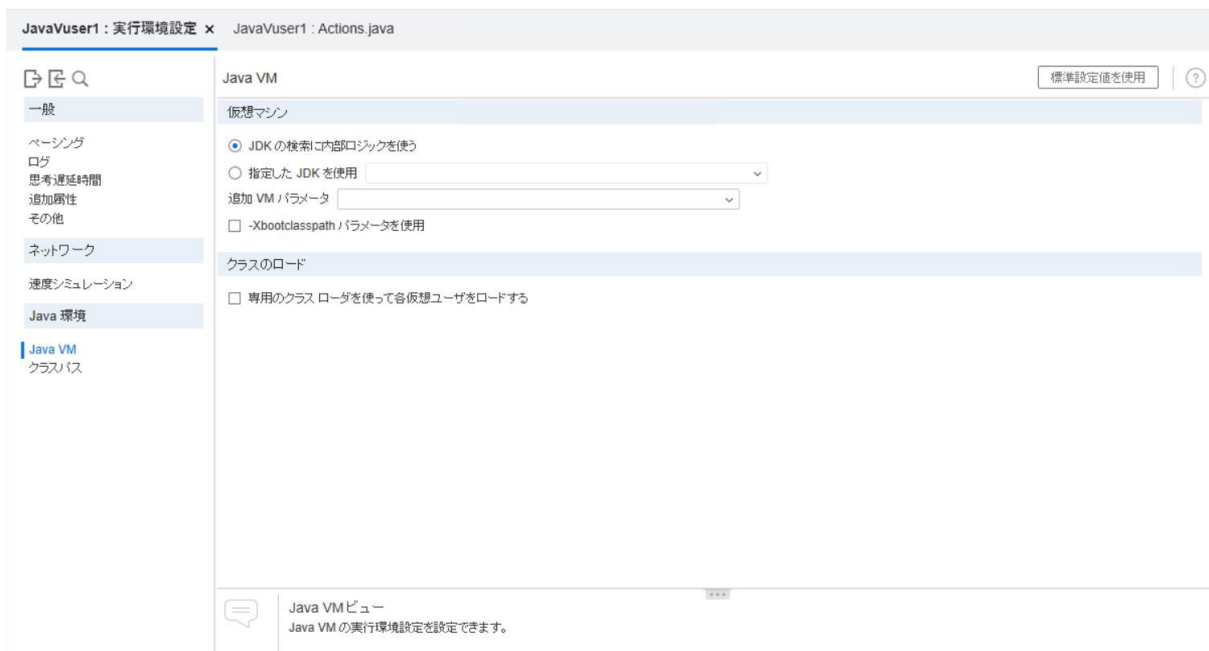
次に示すのは、Java, JMeter, または Gatling プロトコルを実行するための dockerfile の内容の例です。



例:

```
#escape=`  
FROM performancetesting/opentext_onelg_windows:<tag version  
number>  
COPY .\<folder contains JDK> <target path in the container>
```

- **COPY** 行の <target path in the container> で定義するターゲット JDK ディレクトリへのパスは、**Java VM** の実行環境設定ページにも追加する必要があります。



- **Java 64 ビット・プロトコルのテストの場合**は、マシンの PATH 環境変数に JDK 64 ビット bin フォルダへのパスを追加するため、Dockerfile に次のコマンド・ラインを追加します。

```
RUN powershell [Environment]::SetEnvironmentVariable(\"Path\",  
$env:Path + \";<target JDK path in the container>\bin\",  
[EnvironmentVariableTarget]::Machine)
```

- **JMeter/Gatling プロトコルのテストの場合**は、マシンの JAVA_HOME 環境変数に JDK/JRE のパスを追加するため、dockerfile に次のコマンド・ラインを追加します。

```
RUN powershell [Environment]::SetEnvironmentVariable(\"JAVA_<br>HOME\", \"<target JDK/JRE path in the container>\",  
[EnvironmentVariableTarget]::Machine)
```

Load Generator コンテナの実行後

Load Generator コンテナをシナリオに追加します。詳細については、LoadRunner Professional ヘルプセンターの「[Docker 化された Load Generator の設定](#)」を参照してください。

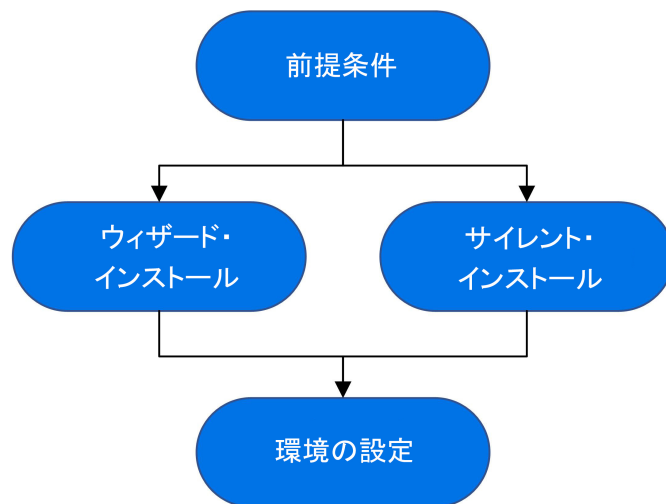
ヒントとガイドライン

- ファイアウォール越しに実行する場合、定義済みイメージを使用して実行される Docker 化された Load Generator はサポートされません。
- `docker ps` を使用して、実行中のコンテナをリストします。
- Load Generator サービスを停止するには、次の手順を実行します。
 - 同じ Load Generator を再利用する場合は、`docker stop <Load Generator コンテナ名または ID>` を使用します。
 - Load Generator コンテナを削除する場合は、`docker rm -f <Load Generator コンテナ名または ID>` を使用します。
- ホスト・ネットワークに直接アクセスするには、`-p <host_port>:54345` の代わりに `--net=host` を使用します。AUT によって大量のネットワーク・アクティビティが生成される場合は、このフラグを使用することをお勧めします。

Linux でのインストールのワークフロー

次の図は、標準のインストール・プログラムを使用して、Load Generator を Linux プラットフォームにインストールする主要な手順を示しています。

Docker コンテナを使用して Linux に Load Generator をデプロイする方法の詳細については、「[Linux での Docker 化された Load Generator のデプロイ](#)」(37ページ)を参照してください。



1. 前提条件

- システムが「[システム要件と前提条件](#)」(8ページ)に記載されている要件を満たしていることを確認します。
- **RHEL 8** オペレーティング・システムを使用する場合:標準設定の RHEL 8.0 には libnsl.so.1 がインストールされていないため、Load Generator のインストール前に手動でインストールする必要があります。これを行わないと、インストールは失敗します。インストールにはパッケージ・マネージャを使用します:`yum install libnsl.i686`
- Linux マシンで **ODBC プロトコル** スクリプトを再生する場合は、unixODBC v.2.3.1 以降をインストールします。
- このバージョンの Load Generator は、バージョン 12.56 以降の任意の Load Generator インストールにインストールできます。Load Generator バージョン 12.55 以前がインストールされている場合は、最初にそれをアンインストールする必要があります。詳細については、「[Load Generator 12.55 以前のアンインストール](#)」(51ページ)を参照してください。

2. セットアップを実行する

- **ウィザード・インストール**:セットアップ・ウィザードを使用してインストールを実行するには、「[セットアップ・ウィザードによる Linux 上での Load Generator のインストール](#)」(34ページ)を参照してください。
- **サイレント・インストール**:サイレント・インストールの実行方法の詳細については、「[Linux での Load Generator のサイレント・インストール](#)」(36ページ)を参照してください。

3. 環境を設定する

インストール後、Load Generator を使用する前に、環境を設定する必要があります。これには、該当する環境変数の設定、Load Generator へのアクセスの確認、インストールの検証が伴います。「[Linux 環境の設定](#)」(43ページ)を参照してください。

セットアップ・ウィザードによる Linux 上での Load Generator のインストール

本項では、セットアップ・ウィザードを使用して、Linux プラットフォームに Load Generator をインストールする方法について説明します。

セットアップ・ウィザードを使用して Load Generator をインストールするには、次の手順を実行します。

1. スーパー・ユーザに切り替えます。
2. Load Generator バージョン **12.55 以前** がマシンにインストールされている場合は、手動でアンインストールする必要があります。この Load Generator をアンインストールしないと、インストール中にエラーが発生します。詳細については、[「Load Generator 12.55 以前のアンインストール」\(51ページ\)](#)を参照してください。

Load Generator **12.56 以降** がインストールされている場合は、インストール・プロセスでセットアップ・ウィザードによってアンインストールされた後に再インストールされます。

3. Load Generator インストール・パッケージを解凍し、**.bin** ファイルの場所を開きます。
4. セットアップ・ウィザードを起動します。

対象:	次のように入力します。
[sh および bash シェル]	source ./installer.sh
[csh および tcsh シェル]	source ./installer.csh
source コマンドが現在のシェルでサポートされていない場合、「ドット」コマンドを使用します。	例: . ./installer.csh

注: セットアップ・ウィザードを実行する場合は、上記のように **source** コマンドを使用することをお勧めします。このコマンドを使用せずにセットアップ・ウィザードを実行する場合は、現在のシェル・セッションの環境変数を手動で設定する必要があります。詳細については、[「環境変数の設定」\(43ページ\)](#)を参照してください。

Load Generator セットアップ・ウィザードでは、必須のソフトウェアがコンピュータにインストールされているかどうかを確認されます。見つからない必須ソフトウェアがある場合、メッセージが表示され、セットアップ・ウィザードが終了します。必要なパッケージをインストールします。その後、セットアップ・ウィザードを再度実行します。

5. 画面の指示に従って Load Generator をインストールします。

Load Generator のインストール中に、CA および TLS(SSL) 証明書をインストールするオプションがあります。これらの証明書の詳細と要件については、「[LRP での TLS\(SSL\) 証明書の使用](#)」(8ページ)を参照してください。

インストール中の詳細なヘルプについては、「[Linux Load Generator のインストールに関するトラブルシューティング](#)」(52ページ)を参照してください。

注: 標準設定のインストール・パスは次のとおりです。/opt/OT/OT_LoadGenerator/_OT_LoadGenerator_Installation

6. スーパー・ユーザを終了するか、別のユーザに切り替えます。
7. 「[Linux 環境の設定](#)」(43ページ)に示すように環境を設定します。

Linux での Load Generator のサイレント・インストール

本項では、Linux マシンで Load Generator のサイレント・インストールを行う方法について説明します。

Load Generator のサイレント・インストールを行うには、次の手順で行います。

1. スーパー・ユーザに切り替えます。
2. Load Generator がマシンにインストールされている場合は、最初に Load Generator をアンインストールする必要があります。最初に Load Generator をアンインストールしておかないと、サイレント・インストールは失敗します。Load Generator のアンインストールについては、「[Linux マシンからの Load Generator のアンインストール](#)」(50ページ)を参照してください。
3. ディレクトリを /<path_to_installer_dvd>/load_generator_linux_x64/ に変更します。
4. 次のコマンドを実行して、Load Generator をサイレント・インストールします。以下の表に示すように、インストール・オプションを追加できます。

```
source ./installer.sh -i silent
```

インストール・オプション	指定可能な値	詳細
-DSTART_ PRODUCT_AFTER_ INSTALL	Yes, No	インストールの終了時に Load Generator を起動します。標準設定値:Yes
-DINSTALL_ GATLING	Yes, No	Load Generator に Gatling をインストールします。標準設定値: No
-DINSTALL_JMETER	Yes, No	Load Generator に JMeter をインストールします。標準設定値: No



例: 次のコマンドは、JMeter を使用する Load Generator のサイレント・インストールを実行します。Load Generator はインストールの終了時に起動しません。

```
source ./installer.sh -i silent -DINSTALL_JMETER=Yes -DSTART_  
PRODUCT_AFTER_INSTALL=No
```



注: サイレント・インストールを実行する場合は、上記のように **source** コマンドを使用することをお勧めします。**source** コマンドを使用せずにインストールを実行する場合は、Load Generator のインストール後に環境変数を設定する必要があります。詳細については、「[環境変数の設定](#)」(43ページ)を参照してください。

source コマンドが現在のシェルでサポートされていない場合、「ドット」コマンドを使用します。例: `./installer.sh - i silent`

インストール中にエラーが発生した場合は、「[Linux Load Generator のインストールに関するトラブルシューティング](#)」(52ページ)を参照してください。

Linux での Docker 化された Load Generator のデプロイ

本項では、Linux ディストリビューションで Docker 化された Load Generator を実行する方法について説明します。

Docker 化された Load Generator でサポートされているプロトコルについては、『[Supported Protocols](#)』ガイドを参照してください。

前提条件

注: OneLG Load Generator 用の Ubuntu イメージは、以前の Ubuntu Load Generator の Docker イメージを置き換えるものです。

OneLG の詳細については、LoadRunner Professional ヘルプセンターの「[Load Generator](#)」を参照してください。

- Docker をその依存関係とともにターゲット・マシンにインストールし、必要に応じてターゲット・マシン環境を設定します。現在のところ、64 ビット・バージョンのみがサポートされています。インストールの詳細については、Docker のオンライン・ドキュメントを参照してください。
- 定義済みの Load Generator Docker イメージを取得します。Linux には、Ubuntu 用と RHEL 用の 2 つのイメージが用意されています。

Docker Hub の「[performancetesting](#)」ページ

(<https://hub.docker.com/u/performancetesting>) からアクセスできる関連ページから、イメージをダウンロードします。

次のいずれかのコマンドと、適切な **<タグ・バージョン番号>** (24.1 など) を使用します。

Ubuntu

```
docker pull performancetesting/opentext_onelg_ubuntu:<tag version number>
```

RHEL

```
podman pull performancetesting/opentext_onelg_rhel:<tag version number>
```

定義済みイメージを使用して Docker 化された Load Generator を実行する

既製のイメージを使用して、Docker for Linux で Load Generator を実行します。プロキシ・サーバなどでコンテナのカスタマイズが必要な場合は、「[カスタム・イメージを使用して Docker 化された Load Generator を実行する](#)」(40ページ)を参照してください。

LoadRunner Professional 用の Ubuntu OneLG イメージを使用するには、実行コマンドに適切な環境変数を含める必要があります:ONELG_FLAVOR=1

さらに、必要に応じて次の環境変数を使用し、Load Generator で JMeter と Gatling を有効にすることができます。

- ENABLE_JMETER
- ENABLE_GATLING

Docker 化された Load Generator を実行するには、次の手順を実行します。

Ubuntu または RHEL の適切なコマンドを使用して、Load Generator コンテナを実行します (Gatling または JMeter を有効にするには、太字の変数を含めます)。

Linux マシン上で <host_port> が使用可能で、要求の受信が許可されていることを確認します。この Load Generator に接続する場合は、Controller 側でこのポートを指定します。

- **Ubuntu:**

```
docker run -id -p <host_port>:54345 -e "ONELG_FLAVOR=1" -e
"ENABLE_GATLING=1" -e "ENABLE_JMETER=1" --net=host
performancetesting/opentext_onelg_ubuntu:<tag version number>
```

- **RHEL:**

```
podman run -id -p <host_port>:54345 -e "ENABLE_GATLING=1" -e
"ENABLE_JMETER=1" performancetesting/opentext_onelg_rhel:<tag
version number>
```

SSH を使用する例

SSH を使用して複数の Load Generator コンテナを実行する簡単な C# コードの例を次に示します。Kubernetes, OpenShift, Docker Swarm などのコンテナ・オーケストレータ・ツールでは、これと同じことが行われます。

```
using (var client = new SshClient(dockerHost, dockerHostUserName,
dockerHostPasswd))
{
    client.Connect();
    for (int i = 0; i > numOfContainers; i++)
    {
```

```
        string command = "docker run -id -p " + lgInitialPort + i) +
        ":54345 -e \"ONELG_FLAVOR=1\" --net=host performancetesting/opentext_
        onelg_ubuntu:<tag version number>";
        var terminal = client.RunCommand(command);
        if (terminal.ExistStatus != 0)
        {
            throw new Exception("Failed to create new Docker container");
        }
        Console.WriteLine("Docker LG with external port" +
        lgInitialPort + i + "created.");
    }
    client.Disconnect();
}
```

カスタム・イメージを使用して Docker 化された Load Generator を実行する

お使いの環境のプロキシ・サーバなどでカスタマイズした設定を使用してコンテナを実行する必要がある場合は、以下の手順に従って Dockerfile を作成し、カスタム・イメージをビルドできます。

または、次のようにすることもできます。コンテナを起動します。コンテナが稼働したら、Load Generator の環境変数を設定し、コンテナ内で Load Generator を手動で起動します。

カスタム・イメージを使用して Docker 化された Load Generator を実行するための Dockerfile を作成するには、次の手順を実行します。

1. 新しいフォルダを作成し、その中に **dockerfile** という名前のファイルを作成します。このファイルに **FROM** 行を貼り付け、必要なカスタマイズ行を追加します。<tag version number> には LoadRunner Professional の該当するバージョンを指定します。

たとえば、Ubuntu の場合は次のようになります。

```
FROM performancetesting/opentext_oneig_ubuntu:<tag version
number>
ENV http_proxy http://my_proxy_name:port
```


注: 上記のカスタマイズ例はプロキシの場合です。この例では、ターゲット・イメージのプロキシ・サーバ・ホストとポートに対する環境変数を定義しています。

2. dockerfile を保存します。
3. dockerfile フォルダ・パスでコマンド・ラインを開き、使用するカスタム・イメージの名前を指定して、次のコマンドを実行します。

Linux-Ubuntu:

```
docker build -t <custom image name> .
```

RHEL:

```
podman build -t <custom image name> .
```

4. 次のコマンドを実行して、使用する Load Generator ごとにコンテナを1つずつ作成します。

Ubuntu:

```
docker run -id -p <host_port>:54345 -e "ONELG_FLAVOR=1" --net=host <custom image name>
```

RHEL:

```
podman run -id -p <host_port>:54345 <custom image name>
```

ステップ 3 でタグを指定してカスタム・イメージをビルドしている場合は、次のようにコマンド内でタグを指定します。

```
docker run -id -p <host_port>:54345 -e "ONELG_FLAVOR=1" --net=host <custom image name>:<tag version number>
```

注: Linux マシン上で <host_port> が使用可能で、要求の受信が許可されていることを確認します。この Load Generator に接続する場合は、Controller 側でこのポートを指定します。

Load Generator コンテナの実行後

Load Generator コンテナをシナリオに追加します。詳細については、LoadRunner Professional ヘルプセンターの「[Docker 化された Load Generator の設定](#)」を参照してください。

ヒントとガイドライン

- ファイアウォール越しに実行する場合、定義済みイメージを使用して実行される Docker 化された Load Generator はサポートされません。(上級ユーザ向けの回避策: MI Listener をサポートする独自の Docker イメージを作成できます。)
- `docker ps` を使用して、実行中のコンテナをリストします。
- Load Generator サービスを停止するには、次の手順を実行します。
 - 同じ Load Generator を再利用する場合は、`docker stop <Load Generator コンテナ名または ID>` を使用します。
 - Load Generator コンテナを削除する場合は、`docker rm -f <Load Generator コンテナ名または ID>` を使用します。
- Dockerfile コンテナには、ENTRYPOINT セクションがあります。コンテナは、最初 ENTRYPOINT でコマンドを実行します。これは環境を設定してから、Load Generator を起動します。このコマンドは、While ループを使用して入力を待機し、コンテナが終了しないようにします。この動作によって、コンテナの実行中、コンテナにアクセスできなくなります。コンテナの起動中、必ず `-i` を追加してください。そうしない場合、While ループによって CPU が過剰に消費されます。
- コンテナに入る必要がある場合は、コンテナの起動時に `--entrypoint=/bin/bash` などの引数を追加します。コンテナに入ったら、Load Generator 環境を設定して、Load Generator を起動します。その後、コンテナをバックグラウンドで実行しながら、CTRL+p と CTRL+q を使用してホストに切り替えることができます。コンテナに再度アクセスするには、`docker attach container_id` コマンドを使用します。
- ホスト・ネットワークに直接アクセスするには、`-p <host_port>:54345` の代わりに `--net=host` を使用します。AUT によって大量のネットワーク・アクティビティが生成される場合は、このフラグを使用することをお勧めします。

Linux 環境の設定

本項では、インストールした Load Generator を使用する前に完了する必要がある設定手順について説明します。

Load Generator のインストール後にセットアップ・プロセスを完了するには、次の手順で行います。

1. 適切な環境変数を設定する。

[「環境変数の設定」\(43ページ\)](#)を参照してください。



注: `source` コマンドを使用して Load Generator をインストールした場合、セットアップ・ウィザードによって適切な環境変数が自動的に設定されるため、この手順を実行する必要はありません。

2. Load Generator のインストールを検証する。

[「Linux インストールの検証」\(44ページ\)](#)を参照してください。

3. Load Generator を開始する。

```
.cd /opt/OT/OT_LoadGenerator/;source env.csh;cd bin;./m_daemon_
setup -install
```

4. Controller が Load Generator にアクセスできるかどうかを確認する。

詳細については、[「Controller の接続の確認」\(46ページ\)](#)を参照してください。

環境変数の設定



注: 本項目の説明は、`source`(または「dot」) コマンドを使用せずに Load Generator セットアップ・ウィザードを実行した場合のみに適用されます。これらのコマンドを使用した場合、以下の手順を実行する必要はありません。

Load Generator を実行できるようにするには、次の環境変数を定義する必要があります。

- **M_LROOT:** Linux Load Generator のルート・フォルダの場所。
- **PATH:** Linux Load Generator の `bin` ディレクトリの場所。
- **PRODUCT_DIR:** Linux Load Generator のルート・フォルダの場所。

Load Generator セットアップ・ウィザードは、これらの環境変数に関連する次のタスクを実行します。

- 環境変数の定義をシステム全体の起動スクリプトに追加します。
セットアップ中に変数の定義が正しく設定されなかった場合は、可能な解決策について「[Linux Load Generator のインストールに関するトラブルシューティング](#)」(52ページ)を参照してください。

- **source** コマンドを使用してセットアップ・ウィザードが実行されている場合、現在のシェル・セッションに対して環境変数を設定します。

本項目では、**source** コマンドを使用せずにセットアップ・ウィザードを実行した場合の現在のシェル・セッションに対する環境変数の設定方法について説明します。

環境変数が設定されているかどうかを判別するには、**verify_generator**(「[verify_generator の実行](#)」(45ページ)を参照)を実行するか、次のコマンドを使用します。

```
echo $M_LROOT
```

Load Generator のルート・フォルダの名前が返される場合、環境設定が現在のシェルに対して正しく設定されています。Load Generator のルート・フォルダの名前が返されない場合は、次のように変数を手動で設定する必要があります。

現在のシェル・セッションに対して環境変数を手動で設定するには(**source** コマンドを使用せずにセットアップ・ウィザードを実行した場合)、次のコマンドのいずれかを実行します。

- Bash ユーザの場合

```
source <Load Generator のルート>/env.sh
```

- C シェル・ユーザの場合

```
source <Load Generator のルート>/env.csh
```

Linux インストールの検証

Load Generator インストールには、Linux マシンでの Load Generator セットアップを確認するセットアップ検証ユーティリティ **verify_generator** が含まれています。この検証ユーティリ

ティは、環境変数と起動スクリプト (`/etc/csh.cshrc`, `${HOME}/.cshrc` または `/etc/profile`, `${HOME}/.profile`) を検査して、正しく設定されていることを検証します。

Load Generator のインストールが終了したら、Load Generator を起動する前に **verify_generator** ユーティリティを実行することを強くお勧めします。**verify_generator** ユーティリティの実行方法の詳細については、「[verify_generator の実行](#)」(45ページ)を参照してください。

verify_generator ユーティリティは次のことを確認します。

- 必須のソフトウェアがすべてインストールされていること。(この確認は、64ビットのインストールに対してのみ実行されます。)
- 少なくとも128のファイル記述子があること。
- `.rhosts` 権限が次のように正しく定義されていること。 `-rw-r--r--`
- `rsh` を使用してホストに接続できること。接続できない場合は、`.rhosts` 内のホスト名を確認する。
- **M_LROOT** が定義されていること。
- `.cshrc` または `.profile` が正しい **M_LROOT** を定義していること。
- `/etc/csh.cshrc`, `${HOME}/.cshrc` または `/etc/profile`, `${HOME}/.profile` が正しい **M_LROOT** を定義していること。
- `.cshrc` または `.profile` がホーム・ディレクトリに存在すること。
- 現在のユーザが `.cshrc` または `.profile` の所有者であること。
- Linux Load Generator が **\$M_LROOT** にインストールされていること。
- 実行可能ファイルに実行可能なアクセス許可があること。
- `PATH` に **\$M_LROOT/bin** および `/usr/bin` が含まれていること。

verify_generator の実行

Load Generator のインストールが終了したら、Load Generator を起動する前に **verify_generator** ユーティリティを実行することをお勧めします。**verify_generator** ユーティリティによる検証内容の詳細については、「[Linux インストールの検証](#)」(44ページ)を参照してください。

注:

- このコマンドを実行するには、root ユーザではなく「通常」のユーザでなければなりません。
- **verify_generator** ユーティリティを実行する前に、マシン上に DISPLAY 環境変数を設定していることを確認します。

verify_generator を実行するには、次の手順を実行します。

1. <Load Generator のルート> /bin フォルダから、次のコマンドを実行します。

```
./verify_generator
```

例:

```
/opt/OT/OT_LoadGenerator/bin/verify_generator
```

検査に関する詳細情報を入手する場合は、-v オプションを次のように使用します。

```
./verify_generator -v
```

2. 結果を確認します。
 - 設定が正しければ、**verify_generator** によって **OK** が返されます。
 - 設定が正しくなければ **verify_generator** によって **Failed** が返され、設定の訂正方法が示されます。

Controller の接続の確認

Controller が **rsh**(リモート・シェル) を使用して Load Generator にリモートで接続する場合、Controller が Load Generator にリモートでアクセスできることを確認する必要があります。

1. Load Generator マシンでユーザのホーム・ディレクトリにある **.rhosts** ファイルを特定します。
2. **.rhosts** ファイルで、Controller がマシンの一覧に含まれていることを確認します。一覧にない場合は、一覧に追加します。

それでも Controller が Load Generator に接続できない場合は、システム管理者にお問い合わせください。

rsh を使用しない Linux Load Generator への接続

Controller は、**rsh** を使用せずに Load Generator に接続するように設定できます。この場合、次のように Load Generator でエージェント・デーモンをアクティブ化する必要があります。

本項では、**rsh** を使用せずに Linux Load Generator に接続する方法について説明します。

1. Linux Load Generator で、**<Load Generator のルート>/bin** から次のコマンドを入力してエージェント・デーモンを実行します。

```
./m_daemon_setup -install
```

これにより **m_agent_daemon** というデーモンが実行され、成功した場合は「**m_agent_daemon <プロセス ID>**」というメッセージが表示されます。

エージェントは、ユーザがログオフしても稼動し続けます。後に示す手順 9 で説明するコマンドを使用するか、マシンを再起動することで、エージェントを停止できます。

注: 一時ディレクトリにある **m_agent_daemon[xxx].log** ログ・ファイルには、インストールが正常に行われた場合でも、通信エラーが記録される場合があります。

2. Controller で、**[シナリオ]>[Load Generators]** を選択します。[Load Generators] ダイアログ・ボックスが開きます。
3. **[追加]** をクリックします。[新規 Load Generator の追加] ダイアログ・ボックスが開きます。
4. **[名前]** ボックスに、Load Generator が稼動しているコンピュータの名前を入力します。
5. **[プラットフォーム]** の一覧で、**[Linux]** を選択します。
6. **[詳細表示]** をクリックします。
7. **[Linux 環境]** タブをクリックし、**[RSH を使用しない]** チェック・ボックスが選択されていることを確認します。
8. 通常どおり接続します。
9. エージェント・デーモンを停止するには、**<LoadRunner Professional のルート>/bin** ディレクトリから次のコマンドを実行します。

```
./m_daemon_setup -remove
```

これにより `m_agent_daemon` が停止され、成功した場合は「`m_agent_daemon is down`」というメッセージが表示されます。

Load Generator のパフォーマンスの改善

このセクションでは、Load Generator のパフォーマンスを向上させるための推奨事項について説明します。カーネルを再構築することで、ファイル記述子、プロセス・エントリ、スワップ領域の容量を増やせます。

注: Linux Load Generator を使用するほとんどのオペレーティング・システムには、十分な数のファイル記述子、プロセス・エントリ、およびスワップ領域が標準で用意されています。

本項の内容:

- [「ファイル記述子を増やす」\(48ページ\)](#)
- [「プロセス・エントリを増やす」\(49ページ\)](#)
- [「スワップ領域を増やす」\(50ページ\)](#)

ファイル記述子を増やす

Load Generator は、次のファイル記述子リソースを使用します。

- 起動サービス用に 14 のファイル記述子
- エージェント用に 20 のファイル記述子
- 各仮想ユーザ・ドライバ用に 30 のファイル記述子。標準設定では 50 仮想ユーザごとに 1 つのドライバがあります。
- 実行中の仮想ユーザ用のファイル記述子。各仮想ユーザには 2 つの記述子が必要です。

たとえば、スレッドとして実行する 100 個の仮想ユーザを実行するのに使用するファイル記述子の数を計算すると、Load Generator は以下を必要とします。

記述子	記述子の目的
14	ランチャー用

20	エージェント用
60	2つのドライバ用(30 x 2。それぞれが50仮想ユーザまで駆動する)
200	100仮想ユーザ用(各仮想ユーザに2つ必要)

合計: 294 のファイル記述子

仮想ユーザをスレッドではなくプロセスとして実行する場合は、仮想ユーザごとに1つのドライバが実行されます。したがって、各仮想ユーザには30のファイル記述子が必要です。

シェルによって、ファイル記述子の増やし方の手順が異なります。

次の例では、記述子の数を最大1024まで増やします。

- sh ユーザと ksh ユーザは、次のように入力します。

```
ulimit -n 1024
```

- csh ユーザは、次のように入力します。

```
limit descriptors 1024
```

ファイル記述子を増やすもう一つの方法を次に示します。この例では、記述子の数を最大8192まで増やします。

1. **/etc/security/limits.conf** ファイルに次の行を追加します。

```
hard nfile 8192
```

2. **/etc/sysctl.conf** ファイルに次の行を追加します。

```
fs.file-max = 8192
```

3. マシンを再起動します。

プロセス・エントリを増やす

各仮想ユーザには空いているプロセス・エントリがいくつか必要です。システムでプロセス・エントリ数を増やすには、カーネルを再構築しなければなりません。

本項では、Linux プラットフォームでのカーネルの再構築の方法について説明します。

1. **/etc/security/limits.conf** ファイルを開きます。
2. limits ファイルでプロセスの最大数を設定します。次のように入力します。

```
hard nproc 8192
```

3. マシンを再起動します。

スワップ領域を増やす

各仮想ユーザには、サイズが 200 KB ~ 4 MB の範囲のスワップ領域が必要です。システム構成に領域を追加する前に、ページング要件を決定することをお勧めします。多くのメモリを必要とするプログラムを実行する環境の場合は、物理メモリの 4 倍のページング領域を確保することをお勧めします。ページング領域を十分に確保しないと、プロセスが強制終了され、ほかのプロセスも起動できなくなることがあります。

Linux マシンからの Load Generator のアンインストール

本項では、Linux マシンから Load Generator をアンインストールする方法について説明します。

本項の内容

- [「Load Generator 12.55 以前のアンインストール」\(51ページ\)](#)
- [「Load Generator 12.56 以降のアンインストール」\(50ページ\)](#)

Load Generator 12.56 以降のアンインストール

Load Generator 12.56 以降のインストールでは、Load Generator はインストール・プロセス中にセットアップ・ウィザードによってアンインストールされ、その後再インストールされます。

また、Load Generator のアンインストールは、手動コマンドを使用するか、またはサイレントで行うこともできます。

注: バージョン 24.1 以降をアンインストールするには、コマンド内の **MF** をすべて **OT** に置き換えます。

Load Generator バージョン 12.56 以降をアンインストールするには、次の手順を実行します。

1. Load Generator をインストールしたユーザでログインしていることを確認します。
2. **m_agent_daemon** プロセスがマシン上で動作していないことを確認します。動作している場合、次のコマンドを使用してそのプロセスを削除します。

```
cd /opt/MF/MF_LoadGenerator/bin;./m_daemon_setup -kill;su -;
```

3. 現在のディレクトリをインストール・ディレクトリに変更します。

```
cd <path_to_installation_folder>/_MF_LoadGenerator_Installation
```

4. スーパー・ユーザに切り替えます。
5. 次の方法で、Load Generator をアンインストールします。

- **手動アンインストール:** 次のコマンドを実行します。

```
sh ./Change_MF_LoadGenerator_Installation
```

- **サイレント・アンインストール:** 次のコマンドを実行します。

```
sh ./Change_MF_LoadGenerator_Installation -i silent
```

Load Generator 12.55 以前のアンインストール

Load Generator 12.55 以前のバージョンは、手動コマンドまたはサイレント・モードでアンインストールできます。

Load Generator バージョン 12.55 以前をアンインストールするには、次の手順を実行します。

1. Load Generator をインストールしたユーザでログインしていることを確認します。
2. **m_agent_daemon** プロセスがマシン上で動作していないことを確認します。動作している場合、次のコマンドを使用してそのプロセスを削除します。

```
cd /opt/HP/HP_LoadGenerator/bin;./m_daemon_setup -kill;su -;
```

3. 現在のディレクトリをインストール・ディレクトリに変更します。

```
cd <path_to_installation_folder>/_HP_LoadGenerator_Installation
```

4. スーパー・ユーザに切り替えます。
5. 次の方法で、Load Generator をアンインストールします。
 - **手動アンインストール:** 次のコマンドを実行します。

```
sh ./Change_HP_LoadGenerator_Installation
```

- **サイレント・アンインストール:** 次のコマンドを実行します。

```
sh ./Change_HP_LoadGenerator_Installation -i silent
```

Linux Load Generator のインストールに関するトラブルシューティング

本項では、Linux Load Generator のセットアップに関するタスクのトラブルシューティングについて説明します。

本項の内容

- [「環境変数がシステム全体の起動スクリプトで正しく設定されなかった」\(52ページ\)](#)
- [「Linux プラットフォームでの Load Generator のインストール時のエラー」\(56ページ\)](#)
- [「Load Generator のアンインストール後に環境変数が設定解除されない」\(56ページ\)](#)
- [「Load Generator で仮想ユーザを実行できない」\(57ページ\)](#)

環境変数がシステム全体の起動スクリプトで正しく設定されなかった

Load Generator を実行できるようにするには、システム全体の起動スクリプトを変更して、特定の環境変数を設定する必要があります。起動スクリプトへの必要な変更は、Load Generator セットアップ・ウィザードによって実行されます。Load Generator のセットアップ時にこれらの起動スクリプトが正しく変更されなかった場合、次に示す方法で起動スクリプトに対する必要な変更を手動で行うことができます。必要な変更は、C シェル・ユーザと Bourne および Korn シェル・ユーザとで多少異なります。

- **C シェル・ユーザの起動スクリプト への手動による変更**

Load Generator のインストール・プロセス中に、ウィザードによって **env.csh** スクリプトが作成されます。このスクリプトには、必要な環境変数を設定するための C シェル・ユーザ用のコマンドが含まれています。サンプルの **env.csh** スクリプトを次に示します。

```
setenv PRODUCT_DIR <Load Generator のインストール・ディレクトリ>
setenv M_LROOT ${PRODUCT_DIR}
  if ( ! $?PATH ) then
    setenv PATH ""
  endif
setenv PATH ${M_LROOT}/bin:${PATH}"
```

/etc/csh.cshrc または **~/.cshrc** 起動スクリプトに次の行を追加して、シェル起動時に **env.csh** スクリプトを実行させます。

```
source <Load Generator のインストール・ディレクトリ>/env.csh
```

例:

```
source /opt/OT/OT_LoadGenerator/env.csh
```

起動スクリプトに対する上記の変更による効果は、セットアップ・ウィザードによって実行される変更と同様です。セットアップ・ウィザードによる **/etc/csh.cshrc** 起動スクリプトへの変更のサンプルを次に示します。

```
# New environment setting added by OT_LoadGenerator on Wed Jan 30
16:20:10 IST 2020 2.

# The unmodified version of this file is saved in
/etc/.login1557000131.

# Do NOT modify these lines; they are used to uninstall.

setenv PRODUCT_DIR "/opt/OT/OT_LoadGenerator"
```

```
# End comments by InstallAnywhere on Wed Jan 30 16:20:10 IST 2020
2.

# New environment setting added by OT_LoadGenerator on Wed Jan 30
16:20:10 IST 2020 5.

# The unmodified version of this file is saved in
/etc/.login1557000131.

# Do NOT modify these lines; they are used to uninstall.

setenv M_LROOT "/opt/OT/OT_LoadGenerator"

# End comments by InstallAnywhere on Wed Jan 30 16:20:10 IST 2020
5.

# New environment setting added by OT_LoadGenerator on Wed Jan 30
16:20:10 IST 2020 8.

# The unmodified version of this file is saved in
/etc/.login1557000131.

# Do NOT modify these lines; they are used to uninstall.

if ( ! $?PATH ) then
setenv PATH ""
endif

setenv PATH "/opt/OT/OT_LoadGenerator/bin:${PATH}"

# End comments by InstallAnywhere on Wed Jan 30 16:20:10 IST 2020
8.
```

- **Bourne および Korn シェル・ユーザの起動スクリプトへの手動による変更**

Load Generator インストール中、セットアップ・ウィザードによって **env.sh** スクリプトが作成されます。このスクリプトには、必要な環境変数を設定するための Bourne シェルおよび Korn シェル・ユーザ用のコマンドが含まれています。

/etc/csh.cshrc または **~/.profile** 起動スクリプトに次の行を追加して、シェル起動時に **env.sh** スクリプトを実行させます。

```
source <Load Generator のインストール・ディレクトリ>/env.sh
```

例:

```
source /opt/OT/OT_LoadGenerator/env.sh
```

起動スクリプトに対する上記の変更による効果は、セットアップ・ウィザードによって実行される変更と同様です。セットアップ・ウィザードによる **/etc/profile** 起動スクリプト への変更のサンプルを次に示します。

```
# New environment setting added by OT_LoadGenerator on Fri Jan 16
11:14:24 IST 2020 1.

# The unmodified version of this file is saved in
/etc/profile1806316421.

# Do NOT modify these lines; they are used to uninstall.
PRODUCT_DIR=/opt/OT/OT_LoadGenerator
export PRODUCT_DIR

# End comments by InstallAnywhere on Fri Jan 16 11:14:24 IST 2020
1.

# New environment setting added by OT_LoadGenerator on Fri Jan 16
11:14:24 IST 2020 4.

# The unmodified version of this file is saved in
/etc/profile1806316421.

# Do NOT modify these lines; they are used to uninstall.
M_LROOT=/opt/OT/OT_LoadGenerator
export M_LROOT

# End comments by InstallAnywhere on Fri Jan 16 11:14:24 IST 2020
4.

# New environment setting added by OT_LoadGenerator on Fri Jan 16
11:14:24 IST 2020 7.

# The unmodified version of this file is saved in
/etc/profile1806316421.

# Do NOT modify these lines; they are used to uninstall.
```

```
PATH="/opt/OT/OT_LoadGenerator/bin:${PATH}"
export PATH

# End comments by InstallAnywhere on Fri Jan 16 11:14:24 IST 2020
7.LoadRunner settings #PATH=${M_LROOT}/bin:$PATH; export PATH
```

Linux プラットフォームでの Load Generator のインストール時のエラー

以前 Load Generator がインストールされていた Linux マシンに **source installer.sh** コマンドを使用して最新の Load Generator バージョンをインストールすると、次のエラー・メッセージが表示される場合があります。

「選択したインスタンスを管理しようとしたときにエラーが発生しました。」

解決策:

1. レジストリ・ファイル **/var/.com.zerog.registry.xml** を開き、属性 **"name"="OT_LoadGenerator"** を持つ要素 **"product"** を特定します。

例:

```
<product name="OT_LoadGenerator" id="77f695c1-1f0c-11b2-883d-
c486a85f6555" version="20.00.0.0" copyright="2019" info_
url="http://www.anyco.com" support_url="http://www.anyco.com"
location="/opt/OT/OT_LoadGenerator" last_modified="2019-11-21 13:12:14">
```

2. **"location"** 属性の値を記録します。
3. **"location"** 属性によって参照されるディレクトリ全体を削除します。
4. レジストリ・ファイル **/var/.com.zerog.registry.xml** を削除します。
5. **source installer.sh** コマンドを再実行します。

Load Generator のアンインストール後に環境変数が設定解除されない

Linux Load Generator をアンインストールしたときに、セットアップ・ウィザードによって Load Generator 環境変数 (M_LROOT, PRODUCT_DIR, および PATH) が現在のシェルに対して設定解除されていない場合があります。環境変数を設定解除するには、現在のシェル

セッションを閉じ、新規のセッションを起動するか、次に示すように変数を手動で設定解除します。

- M_LROOT and PRODUCT_DIR 変数を設定解除するには、次の手順で行います。
 - [bash シェル] **unset** コマンドを使用します。
 - [csh シェル] **unsetenv** コマンドを使用します。
- PATH 変数を更新して、Load Generator バイナリ・ディレクトリを除外するには、次のように入力します。
 - [bash シェル] `PATH=<必要なパス一覧>; export PATH`
 - [csh シェル] `setenv PATH <必要なパス一覧>`

Load Generator で仮想ユーザを実行できない

Load Generator で仮想ユーザを実行できない場合に、具体的なエラーが報告されず、仮想ユーザ・プロトコルによって Load Generator 側にサードパーティ・アプリケーションまたはクライアントが必要になる場合は、そのアプリケーションが使用するダイナミック・ライブラリを検証します。この検証により、見つからない共有オブジェクトがあるかどうかを確認できます。見つからない共有オブジェクトが存在する場合、必須パッケージが不足しているか、環境変数の問題が発生している可能性があります。

アプリケーションで使用するダイナミック・ライブラリを調べるには、次のように入力します。

```
ldd application_name
```

たとえば、`ldd mdrv` と入力すると、**mdrv** 実行可能ファイルのすべての依存関係が見つかるかどうかを判別できます。見つからない依存関係がある場合は、「[verify_generator の実行](#)」(45ページ)の説明に従って **verify_generator** を実行します。

注: クライアント・インストールを必要とするプロトコル(Oracle など)の仮想ユーザを実行している場合は、クライアント・ライブラリのパスがダイナミック・ライブラリのパス環境変数(LD_LIBRARY_PATH または SHLIB_PATH)に含まれていることを確認します。

LoadRunner Data Hub と Web Connector のインストール

LoadRunner Data Hub は、Kafka テクノロジに基づく通信チャネルで、LoadRunner コンポーネント間のより堅牢な通信を可能にします。このチャネルを使用するには、Data Hub をインストールします。さらに、LoadRunner コンポーネントが Web 経由で通信する場合（たとえば、Load Generator がファイアウォールの背後にあり、プロキシ経由で通信する場合）、Data Hub Web Connector をインストールする必要があります。

Data Hub と Web Connector は Linux にインストールする必要があります。これらは、Linux および Windows マシンで実行されている LoadRunner コンポーネントと通信できます。

Data Hub と Web Connector を（必要に応じて）インストールしたら、それらを使用するように LoadRunner コンポーネントを設定する必要があります。詳細については、LoadRunner Professional ヘルプセンターの「[Data Hub と Web Connector のセットアップ](#)」を参照してください。

本章の内容

- [Data Hub のインストール](#) 58
- [Web Connector のインストール](#) 62

Data Hub のインストール

Data Hub をインストールするには、次の主要な手順を実行します。

1. 前提条件

- [サポート・マトリックス](#)で詳しく説明されている通りに、システムが要件を満たしていることを確認します。
- Data Hub では、最新バージョンの Java JRE がインストールされている必要があります。

2. インストール・パッケージをダウンロードする

[AppDelivery Marketplace](#) から Data Hub インストール・パッケージをダウンロードします。

3. セットアップを実行する

- **ウィザード・インストール:**Data Hub セットアップ・ウィザードを使用してインストールを実行するには、[「セットアップ・ウィザードによる Data Hub のインストール」\(59ページ\)](#)を参照してください。
- **サイレント・インストール:**サイレント・インストールの実行方法の詳細については、[「Data Hub のサイレント・インストール」\(59ページ\)](#)を参照してください。

4. インストールを検証する

[「Data Hub のインストールの検証」\(62ページ\)](#)を参照してください。

セットアップ・ウィザードによる Data Hub のインストール

本項では、セットアップ・ウィザードを使用して LoadRunner Data Hub をインストールする方法について説明します。

セットアップ・ウィザードを使用して Data Hub をインストールするには、次の手順を実行します。

1. スーパー・ユーザに切り替えます。
2. Data Hub インストール・パッケージを解凍し、**.bin** ファイルの場所を開きます。
3. セットアップ・ウィザードを起動します。[sh および bash シェル] の場合は、`./inst64.bin`と入力します。
4. 画面の指示に従って Data Hub をインストールします。

標準設定のインストール・パスは次のとおりです。**`/opt/OT/OT_LoadRunner_Data_Hub/_OT_LoadRunner_Data_Hub_Installation`**

インストール中に、CA および TLS(SSL) 証明書をインストールするオプションがあります。これらの証明書の詳細と要件については、[「LRP での TLS\(SSL\) 証明書の使用」\(8ページ\)](#)を参照してください。

5. スーパー・ユーザを終了するか、別のユーザに切り替えます。

Data Hub のサイレント・インストール

本項では、Data Hub のサイレント・インストールを行う方法について説明します。

本項の内容

- [「サイレント・インストールの実行」\(60ページ\)](#)
- [「サイレント・インストールのオプション」\(61ページ\)](#)

サイレント・インストールの実行

次の手順を実行することで、Data Hub のサイレント・インストールを行うことができます。

Data Hub のサイレント・インストールを行うには、次の手順を実行します。

1. スーパー・ユーザに切り替えます。
2. Data Hub インストール・パッケージを解凍します。
3. 次のテンプレートを使用して、**installer.properties** という名前のファイルを作成し、必要に応じてパラメータを変更します。

ヒント: **installer.properties** ファイルは、セットアップ・ウィザードの実行時に生成され、その後のサイレント・インストールに再度使用できます。

```
#インストール・フォルダを選択
#-----
USER_INSTALL_DIR=/opt/OT/OT_LoadRunner_Data_Hub

#リンクの場所を選択
#-----
USER_SHORTCUTS=/usr/local/bin/OT/OT_LoadRunner_Data_Hub

#インストール
#-----
-fileOverwrite_/opt/OT/OT_LoadRunner_Data_Hub/_OT_LoadRunner_
Data_Hub_Installation/Change_OT_LoadRunner_Data_Hub_
Installation.lax=Yes
-fileOverwrite_/opt/OT/OT_LoadRunner_Data_
Hub/zookeeper.service=Yes
-fileOverwrite_/opt/OT/OT_LoadRunner_Data_Hub/kafka.service=Yes
-fileOverwrite_/opt/OT/OT_LoadRunner_Data_Hub/run_after_
install.sh=Yes
```

```
#証明書インストール
#-----
INSTALL_CERTIFICATE=\"\", \"No\"
```

4. **installer.properties** ファイルをインストーラ・ファイルと同じ場所に保存します。
5. 次の構文を使用してインストール・コマンドを実行します。

```
./inst64.bin [-f<installer.properties ファイルのパス> | -options]
```

このコマンドにオプションを追加することで、インストールを変更できます。詳細については、「[サイレント・インストールのオプション](#)」(61ページ)を参照してください。

注:

- **installer.properties** ファイルをインストーラと同じディレクトリに保存した場合、`-f` オプションを使用して別の有効なプロパティ・ファイルを指定しない限り、他のコマンド・ライン・オプションはすべてオーバーライドされます。
- `installer.properties` ファイルのパスは、絶対パスでも、インストーラが存在するディレクトリからの相対パスでもかまいません。

たとえば、次のコマンドは、インストーラと同じ場所に保存されている **installer.properties** ファイルを使用して、Data Hub のサイレント・インストールを行います。



例: `./inst64.bin -i silent`

サイレント・インストールのオプション

サイレント・インストール・コマンドには、次のオプションを追加できます。

オプション	説明
<code>-i [swing console silent]</code>	インストーラのユーザ・インタフェース・モードを指定します。
<code>-D<名前>=<値></code>	インストーラのプロパティを指定します。
<code>-r<応答ファイルの生成先のパス></code>	応答ファイルを生成します。

修復とアンインストールのオプション

Data Hub を修復またはアンインストールするには、次のオプションを使用します。

オプション	説明
-repair	インストールを修復します。
-uninstall	Data Hub をアンインストールします。

Data Hub のインストールの検証

Data Hub が正常にインストールされたことを検証するには、次のコマンドを実行して、zookeeper サービスと kafka サービスのステータスを確認します。これらのサービスが実行されている場合、インストールは成功です。

```
service zookeeper status
```

```
service kafka status
```

Web Connector のインストール

標準のインストール・プログラムを使用して Web Connector をインストールするには、次の主要な手順を実行します。

1. システム要件の確認

Web Connector をインストールする前に、[サポート・マトリックス](#)で詳しく説明されている通りに、システムが要件を満たしていることを確認します。

2. インストール・パッケージをダウンロードする

Web Connector インストール・パッケージは、Data Hub インストール・パッケージに含まれています。Data Hub インストール・パッケージをまだダウンロードしていない場合は、[AppDelivery Marketplace](#) からダウンロードしてください。

3. セットアップを実行する

- **ウィザード・インストール:** Web Connector セットアップ・ウィザードを使用してインストールを実行するには、「[セットアップ・ウィザードによる Web Connector のインストール \(63ページ\)](#)」を参照してください。
- **サイレント・インストール:** サイレント・インストールの実行方法の詳細については、「[Web Connector のサイレント・インストール \(63ページ\)](#)」を参照してください。

4. インストールの検証

[「Web Connector のインストールの検証」\(66ページ\)](#)を参照してください。

セットアップ・ウィザードによる Web Connector のインストール

本項では、セットアップ・ウィザードを使用して Web Connector をインストールする方法について説明します。

セットアップ・ウィザードを使用して Web Connector をインストールするには、次の手順を実行します。

1. スーパー・ユーザに切り替えます。
2. Web Connector インストール・パッケージを解凍し、.bin ファイルの場所を開きます。
3. セットアップ・ウィザードを起動します。[sh および bash シェル] の場合は、`「./inst64.bin」`と入力します。
4. 画面の指示に従って Web Connector をインストールします。

標準設定のインストール・パスは次のとおりです。**`/opt/OT/OT_LoadRunner_Data_Hub_Web_Connector/_OT_LoadRunner_Data_Hub_Web_Connector_Installation`**

インストール中に、CA および TLS (SSL) 証明書をインストールするオプションがあります。これらの証明書の詳細と要件については、[「LRP での TLS \(SSL\) 証明書の使用」\(8ページ\)](#)を参照してください。

5. スーパー・ユーザを終了するか、別のユーザに切り替えます。

Web Connector のサイレント・インストール

本項では、Web Connector のサイレント・インストールを行う方法について説明します。

本項の内容

- [「サイレント・インストールの実行」\(64ページ\)](#)
- [「サイレント・インストールのオプション」\(65ページ\)](#)

サイレント・インストールの実行

次の手順を実行することで、Web Connector のサイレント・インストールを行うことができます。

Web Connector のサイレント・インストールを行うには、次の手順を実行します。

1. スーパー・ユーザに切り替えます。
2. Web Connector インストール・パッケージを解凍します。
3. 次のテンプレートを使用して、**installer.properties** という名前のファイルを作成し、必要に応じてパラメータを変更します。ここにあるテンプレートでは、標準設定の LoadRunner 証明書を使用して Web Connector をインストールします。

ヒント: **installer.properties** ファイルは、セットアップ・ウィザードの実行時に生成され、その後のサイレント・インストールに再度使用できます。

```
#インストール・フォルダを選択
#-----
USER_INSTALL_DIR=/opt/OT/OT_LoadRunner_Data_Hub_Web_Connector
#リンクの場所を選択
#-----
USER_SHORTCUTS=/usr/local/bin/OT/OT_LoadRunner_Data_Hub_Web_Connector
#インストール
#-----
-fileOverwrite_/opt/OT/OT_LoadRunner_Data_Hub_Web_Connector/_OT_LoadRunner_Data_Hub_Web_Connector_Installation/Change_OT_LoadRunner_Data_Hub_Web_Connector_Installation.lax=Yes
-fileOverwrite_/opt/OT/OT_LoadRunner_Data_Hub_Web_Connector/webconnector.service=Yes
-fileOverwrite_/opt/OT/OT_LoadRunner_Data_Hub_Web_Connector/run_after_install.sh=Yes
-fileOverwrite_/opt/OT/OT_LoadRunner_Data_Hub_Web_Connector/webconnector.sh=Yes
#証明書のインストール
#-----
```



```
INSTALL_CERTIFICATE=\\", \\No\"
```

4. **installer.properties** ファイルをインストーラ・ファイルと同じ場所に保存します。
5. 次の構文を使用してインストール・コマンドを実行します。

```
./inst64.bin [-f<installer.properties ファイルのパス> | -options]
```

このコマンドにオプションを追加することで、インストールを変更できます。詳細については、「[サイレント・インストールのオプション](#)」(65ページ)を参照してください。

注:

- **installer.properties** ファイルをインストーラと同じディレクトリに保存した場合、**-f** オプションを使用して別の有効なプロパティ・ファイルを指定しない限り、他のコマンド・ライン・オプションはすべてオーバーライドされます。
- **installer.properties** ファイルのパスは、絶対パスでも、インストーラが存在するディレクトリからの相対パスでもかまいません。

たとえば、次のコマンドは、インストーラと同じ場所に保存されている **installer.properties** ファイルを使用して、Web Connector のサイレント・インストールを行います。



例: `./inst64.bin -i silent`

サイレント・インストールのオプション

サイレント・インストール・コマンドには、次のオプションを追加できます。

オプション	説明
<code>-i [swing console silent]</code>	インストーラのユーザ・インタフェース・モードを指定します。
<code>-D<名前>=<値></code>	インストーラのプロパティを指定します。
<code>-r<応答ファイルの生成先のパス></code>	応答ファイルを生成します。

Web Connector を修復またはアンインストールするには、次のオプションを使用します。

オプション	説明
<code>-repair</code>	インストールを修復します。

オプション	説明
-uninstall	Web Connector をアンインストールします。

Web Connector のインストールの検証

Web Connector が正常にインストールされたことを検証するには、次のコマンドを実行して、webconnector サービスのステータスを確認します。このサービスが実行されている場合、インストールは成功です。

```
service webconnector status
```