

BigFix パッチ
Rocky Linux - ユーザーズ・ガイド



Special notice

Before using this information and the product it supports, read the information in [Notices \(on page Ixiv\)](#).

Edition notice

This edition applies to BigFix version 11 and to all subsequent releases and modifications until otherwise indicated in new editions.

目次

Special notice.....	ii
Edition notice.....	iii
第 1 章. 概要.....	6
本更新リリースの新機能.....	6
サポートされるプラットフォームおよび更新.....	10
サポートされている Rocky Linux リポジトリ.....	10
サイトのサブスクリプション.....	11
第 2 章. ダウンロード・プラグインの使用.....	12
「ダウンロード・プラグインの管理」ダッシュボードの概要.....	13
Rocky Linux ダウンロード・プラグインの登録.....	15
Rocky Linux ダウンロード・プラグインの基本設定の構成.....	17
Rocky Linux ダウンロード・プラグインの詳細設定の構成.....	19
Rocky Linux ダウンロード・プラグインの登録解除.....	21
Rocky Linux ダウンロード・プラグインのアップグレード.....	22
Rocky Linux ダウンロード・プラグインの拡張.....	22
第 3 章. ダウンロード・キャッシュの使用.....	28
Rocky Linux ダウンロード・キャッシュの使用について.....	28
隔離された環境での Rocky Linux ダウンロード・キャッシュの使用.....	34
sha1 フォルダーでのパッケージのキャッシング.....	34
ローカル・キャッシング・フォルダーでのパッケージのキャッシング.....	36
第 4 章. BigFix Patch for Rocky Linux™ の使用.....	37
Fixlet を使用したパッチの適用.....	37
置き換え.....	39
モジュラー型 Fixlet を使用したパッチ.....	39
第 5 章. 複数パッケージのベースラインのインストール.....	40
ベースラインでの複数パッケージのインストール.....	41
第 6 章. カスタム・リポジトリの管理.....	45
「Rocky Linux カスタム・リポジトリ管理」ダッシュボード.....	45
リポジトリの追加.....	46
リポジトリへの Rocky Linux エンドポイントの登録.....	47
リポジトリからの Rocky Linux エンドポイントの登録解除.....	47

リポジトリの削除.....	48
リポジトリのインポート.....	48
DNF のパッケージ更新の確認.....	49
カスタム・リポジトリからのパッケージのインストール.....	49
付録 A. サポート.....	51
付録 B. トラブルシューティング.....	52
付録 C. よくある質問.....	56
Notices.....	lxiv

第1章. 概要

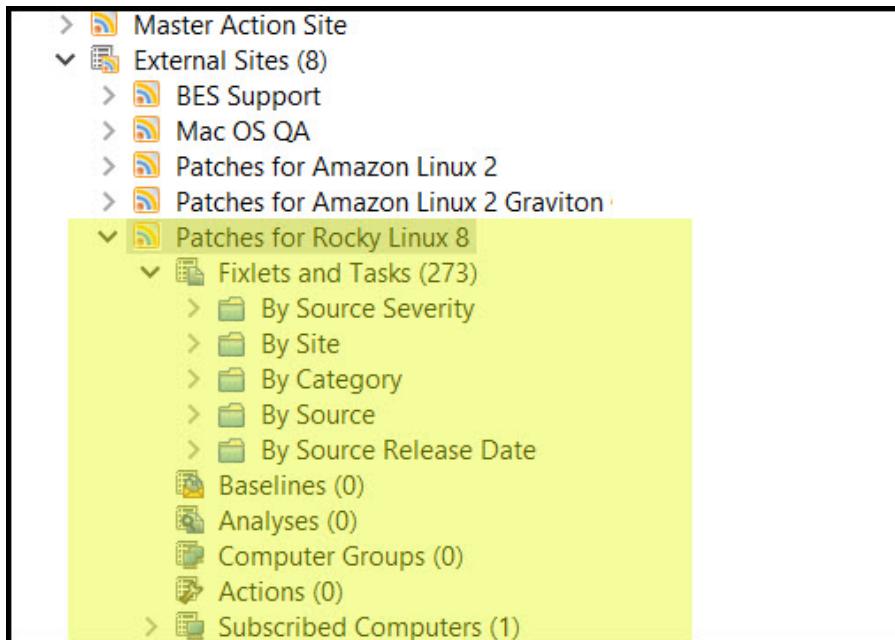
BigFix Patch for Rocky Linux を使用すると、Linux™ クライアントを常に最新の更新およびサービス・パックが適用された状態にすることができます。

BigFix は、新規のパッチまたは更新が入手できるようになるたびに Fixlet® をリリースします。Fixlet により、企業内でそのパッチまたは更新を必要とするすべてのコンピューターが識別され、修正が適用されます。BigFix コンソールのオペレーターは、キーを数回押すだけで、関係するすべてのコンピューターにパッチを適用し、ネットワーク全体の適用の進行状況を視覚化することができます。BigFix エージェントは、オペレーティング・システムのバージョン、ファイルのバージョン、システムの言語、および他の関連する要因をチェックして、パッチが必要かどうかおよびいつ必要となるかを判別します。

BigFix は、リリース前に各 Fixlet® を研究所でテストし、多くの場合に問題を検出します。問題には、Fixlet® に追加のメモを添付するという方法で対処します。通常、そのメモにより、コンソール・オペレーターは問題を回避することができ、パッチの適用プロセスに付加価値を与えるものとなっています。BigFix は、ユーザー・フィードバックもメモに反映して最新の情報を提供します。

Fixlet により、大量の更新およびパッチを比較的簡単に管理することができ、どのようなスケジュールの下でも、自動化され、対象が絞り込まれた適用作業を実行できます。ネットワーク帯域幅を最適化するために大容量のダウンロードを段階的に実行でき、インベントリーまたは監査の制御のために、適用プロセス全体をモニター、グラフ化、および記録できます。

図1. 「パッチ管理」ナビゲーション・ツリー



本更新リリースの新機能

本リリースの BigFix Patch for Rocky Linux には、パッチ適用のさまざまな機能拡張が含まれています。

Rocky Linux コンテンツのサポート

Rocky Linux™ 9 (x86_64) コンテンツが、「Rocky Linux 9 向けパッチ」という新規サイトに公開されています。

このリリースでは、パッケージ・マネージャーのネイティブのコマンド行インターフェースである DNF がサポートされます。DNF を使用すると、Rocky Linux システムにパッチを適用できます。DNF は DNF 上のラッパーであり、これにより、依存関係の問題が削減され、パフォーマンスが向上し、セキュリティー・パッチをより確実にインストールできます。

Rocky Linux ダウンロード・プラグインの機能強化

Rocky Linux ダウンロード・プラグインは、Rocky Linux ダウンロード・キャッシャーの `download_dir` (`plugin.ini` ファイルの `localCache` と呼ばれる設定) によってキャッシュされるパッケージを使用し、同時にインターネットからパッケージを取得できるようになりました。

以前は、Rocky Linux ダウンロード・プラグインは、以下のいずれかの環境にある BigFix server で使用されていました。

- BigFix server は完全に隔離され、Rocky Linux ダウンロード・キャッシャーを使用する。または
- BigFix server はインターネットにアクセスでき、Rocky Linux ダウンロード・キャッシャーを使用しない。

この機能拡張により、パッケージをオフラインでキャッシングして、ダウンロード時間を節約することができます。

詳しくは、「[Rocky Linux ダウンロード・プラグインの詳細設定の構成 \(\(ページ\) 19\)](#)」を参照してください。

Rocky Linux ダウンロード・キャッシャーの機能強化

Rocky Linux ダウンロード・キャッシャーには、次の機能強化が含まれています。

パッケージ sha1 ダウンロード・サポート

Rocky Linux ダウンロード・キャッシャー は、`--sha1_download_dir` を使用することで、RPM 形式ではなく sha1 ファイルとしてパッケージをダウンロードできるようになりました。

これまで、ダウンロード・キャッシャーで「`buildRepo -key Rocky Linux-8-x64`」を使用する と、Rocky Linux リポジトリ「Rocky Linux-8-x64」の構造はオフラインでミラーリングされていました。この方法だと、パッケージが複数のリポジトリにある場合、パッケージが重複してしまう可能性があります。

`--sha1_download_dir` を使用することにより、すべてのパッケージが sha1 のファイル名を持つファイルとして、すべてのリポジトリ (キー) から単一のフラット・ディレクトリーにダウンロードされます。

リポジトリ・アクセス検査

BigFix でサポートされる Rocky Linux 基本リポジトリおよびサブリポジトリにアクセスできるかどうかを検証するための新規コマンドが導入されました。`check-baserepos` および `check-allrepos`

ストレージ・スペース所要量のチェック

`builRepo` コマンドを使用するときにストレージ・スペース所要量を計算して調べるための新規コマンド「`check-storagereq`」が導入されました。

このコマンドは、`--sha1_download_dir` オプションを使用する場合と使用しない場合の、リポジトリ・メタデータおよびパッケージのダウンロードに必要なスペースを出力します。

スペース節約のベンチマーク

`--sha1_download_dir` オプションを使用することで、スペース節約のベンチマークが設定されています。

`--sha1_download_dir` オプションを使用することで、同じ Rocky Linux バージョンの複数のリポジトリをキャッシュする際のストレージ・サイズ、ダウンロード・サイズ、および時間が大幅に削減されます。これは、同じ Rocky Linux バージョン (たとえば、`rockylinux-8.3-x64`、`rockylinux-8.4-x64`、`rockylinux-8.5-x64`) のリポジトリ間で、多くのパッケージが重複しているためです。Rocky Linux バージョン (たとえば、`rockylinux-8.4-x64`、`rockylinux-8.5-x64`) ごとにリポジトリを 1 つだけキャッシュする場合、スペースは節約されません。

詳しくは、以下のトピックを参照してください。

- [Rocky Linux ダウンロード・キャッシュの使用について \(\(ページ\) 28\)](#)
- [隔離された環境での Rocky Linux ダウンロード・キャッシュの使用 \(\(ページ\) 34\)](#)
- [sha1 フォルダーでのパッケージのキャッシング \(\(ページ\) 34\)](#)
- [ローカル・キャッシング・フォルダーでのパッケージのキャッシング \(\(ページ\) 36\)](#)

表 1. Rocky Linux の機能拡張または機能

機能拡張	説明	リソース
複数パッケージのベースラインのインストール	BigFix Patch には、ベースラインに含まれる複数パッケージの更新のインストールを単一のタスクに結合できるソリューションが用意されています。これにより、ベースラインの実行時間を効果的に短縮できます。	複数パッケージのベースラインのインストール ((ページ) 40)
ロギングとエラー処理の強化	エラー・ログとデバッグ出力が設計し直され、エラー報告がより分かりやすくなりました。これにより、問題のトラブルシューティングやデバッグにかかる時間が短縮されます。	Rocky Linux ダウンロード・プラグインの詳細設定の構成 ((ページ) 19)
Rocky Linux 8 コンテンツのサポート	この機能強化は、Rocky Linux 8 プラグイン向けパッチ・サイトで利用できます。	Rocky Linux 8 向けパッチのサイトは、BaseOS と AppStream リポジト
	Rocky Linux™ BaseOS 8 (x86_64) コンテンツが、「CentOS 8 向けパッチ」という新規サイトに公開されています。	

表 1. Rocky Linux の機能拡張または機能 (続く)

機能拡張	説明	リソース
機能拡張	<p>このリリースでは、パッケージ・マネージャーのネイティブのコマンド行インターフェースであるDNFがサポートされます。DNFを使用すると、Rocky Linux 8 システムにパッチを適用できます。DNFはDNF上のラッパーであり、これにより、依存関係の問題が削減され、パフォーマンスが向上し、セキュリティー・パッチをより確実にインストールできます。</p>	<p>リーのみをサポートしています。詳しくは、「サポートされている Rocky Linux リポジトリ」を参照してください。</p>
アナウンス	<p>Rocky Linux 8 will not make announcements on its official site as in earlier versions. BigFix Patch team retrieves the package information from http://dl.rockylinux.org for generating the content based on the changes mentioned in BaseOS and AppStream change list.</p> <p>Red Hat API's are used to get all the dependent packages and errata information.</p>	
置き換え	<p>Rocky Linux 8 repositories metadata now contains information related to latest available packages, hence all the superseded fixlets are expected to fail with an error <code>No matching Package found</code>.</p> <p>Once the fixlet is superseded, the fixlet is made non-relevant by default. The value(client) of <code>_BESClient_Rocky Linux_EnableSupersededEval</code> is removed for superseded fixlets and relevance for superseded fixlets will be set to <code>false <Relevance>False</Relevance></code>. Please deploy only the latest available patches.</p> <p>In order to deploy superseded fixlet using custom site, the relevant repository metadata and packages must be precently cached.</p>	



注: ベースラインの Fixlet を置き換えるとエラーが発生してインストールが失敗するため、インストールを続行する前にすべてのベースラインを同期する必要があります。最新の使用可能なパッケージがベースラインとともにデプロイされます。

サポートされるプラットフォームおよび更新

BigFix Patch for Rocky Linux Rocky Linux 8 プラットフォームで、Rocky Linux セキュリティー・アドバイザリー、修正のアドバイザリー、改善に関するアドバイザリー用の Fixlet が提供されます。

表 2. BigFix Patch for Rocky Linux でサポートされるプラットフォームおよびパッチ

以下の表に、サポートされるプラットフォームと、パッチ適用の Fixlet が含まれる対応サイトをリストします。

サポートされるプラットフォーム	Fixlet サイト名	更新のタイプ
Rocky Linux 8 (x86_64)	Patches for Rocky Linux 8	サポートされるパッケージを含むリポジトリのリストを確認するには、 サポートされている Rocky Linux リポジトリ ((ページ) 10) を参照。
Rocky Linux 9 (x86_64)	Patches for Rocky Linux 9	サポートされるパッケージを含むリポジトリのリストを確認するには、 サポートされている Rocky Linux リポジトリ ((ページ) 10) を参照。



注:

- 置き換えられて 1 年以上になる Fixlet コンテンツは、外部サイトから削除され、別のサイトにアーカイブされます。置き換えられたコンテンツにアクセスする必要がある場合は、HCL の技術営業担当員にお問い合わせください。

サポートされている Rocky Linux リポジトリ

BigFix Patch for Rocky Linux いくつかの Rocky Linux リポジトリのパッケージをサポートしています。

次の表に、各 Rocky Linux バージョンでサポートされるパッケージを含むリポジトリのリストを示します。

表 3. サポートされている Rocky Linux リポジトリ

Rocky Linux バージョン	リポジトリ
Rocky Linux-8-x64	BaseOS: http://dl.rockylinux.org/pub/rocky/8/BaseOS/x86_64/os AppStream: http://dl.rockylinux.org/pub/rocky/8/AppStream/x86_64/os
Rocky Linux-8.3-x64	BaseOS: http://dl.rockylinux.org/vault/rocky/8.3/BaseOS/x86_64/os

表 3. サポートされている Rocky Linux リポジトリ (続く)

Rocky Linux バージョン	リポジトリ
Rocky Linux-8.4-x64	AppStream: http://dl.rockylinux.org/vault/rocky/8.3/AppStream/x86_64/os BaseOS: http://dl.rockylinux.org/vault/rocky/8.4/BaseOS/x86_64/os
Rocky Linux-8.5-x64	AppStream: http://dl.rockylinux.org/vault/rocky/8.4/AppStream/x86_64/os BaseOS: http://dl.rockylinux.org/vault/rocky/8.5/BaseOS/x86_64/os
Rocky Linux-8.6-x64	AppStream: http://dl.rockylinux.org/vault/rocky/8.5/AppStream/x86_64/os BaseOS: http://dl.rockylinux.org/vault/rocky/8.6/BaseOS/x86_64/os
Rocky Linux-9-x64	AppStream: https://dl.rockylinux.org/vault/rocky/9/AppStream/x86_64/os/ BaseOS: https://dl.rockylinux.org/vault/rocky/9/BaseOS/x86_64/os/
Rocky Linux-9.1-x64	AppStream: https://dl.rockylinux.org/vault/rocky/9.1/AppStream/x86_64/os/ BaseOS: https://dl.rockylinux.org/vault/rocky/9.1/BaseOS/x86_64/os/
Rocky Linux-9.2-x64	AppStream: https://dl.rockylinux.org/vault/rocky/9.2/AppStream/x86_64/os/ BaseOS: https://dl.rockylinux.org/vault/rocky/9.2/BaseOS/x86_64/os/

サイトのサブスクリプション

サイトとは、ユーザー、HCL、またはベンダーにより内部的に作成される Fixlet メッセージの集合です。

実装環境内のシステムにパッチを適用するために、サイトにサブスクライブして Fixlet メッセージにアクセスします。

サイトのサブスクリプションを追加するには、ベンダーまたは HCL からマストヘッド・ファイルを入手するか、またはライセンス・ダッシュボードを使用します。Fixlet サイトへのサブスクライブ方法について詳しくは、『BigFix インストール・ガイド』を参照してください。

サイトについて詳しくは、「BigFix コンソール・オペレーター・ガイド」を参照してください。

第2章. ダウンロード・プラグインの使用

ダウンロード・プラグイン、RockyLinux プラグインは、パッチのベンダーから関連パッケージを直接ダウンロードする実行可能プログラムです。Fixlet は、内部プロトコルを使用してダウンロード・プラグインと通信することで、ファイルをダウンロードします。これらの Fixlet は、ベンダーが作成する更新を基盤としています。

Fixlet が前述のプロトコルを使用できるようにするには、BigFix server でダウンロード・プラグインを登録します。該当するプラグインを登録するには、「ダウンロード・プラグインの管理 (Manage Download Plug-ins)」ダッシュボードを使用します。



注:

- ・ダウンロード・プラグインでは、基本認証のみがサポートされます。
- ・null エラーを回避するために、BigFix server と BigFix client は必ず同じバージョンにしてください。

表 4. RockyLinux ダウンロード・プラグイン

ダウンロード・プラグイン名	適用可能サイト
RockyLinux プラグイン	Patches for Rocky Linux 8
RockyLinux プラグイン	Patches for Rocky Linux 9

RockyLinux プラグイン は、ベンダーの Web サイトから BigFix server にパッチを直接ダウンロードしキャッシュします。これにより、パッケージ依存関係の解決およびリポジトリに対するサポートの正確性と信頼性が向上します。



注: RockyLinux プラグイン は、BigFix 管理ツールの「**SHA-256 ダウンロードが必要**」オプションが有効になっている場合は機能しません。このオプションが有効になっていると、すべてのダウンロードの検証で、SHA-256 アルゴリズムのみが使用されます。しかし、リポジトリ内には、プラグインで使用されるパッケージの SHA-256 値が含まれていない、ベンダーからの特定のリポジトリ・メタデータが存在します。

パッチを正常にデプロイするために、「**SHA-256 ダウンロードが必要**」オプションを無効にすることを検討してください。パッケージの GPG シグニチャーを使用して別の層の検査および検証が実行されるため、セキュリティーおよびパッケージの整合性が低下することはありません。ダウンロード・オプションについて詳しくは、BigFix Platform インストール・ガイド (https://help.hcl-software.com/bigfix/10.0/platform/Platform/Installation/c_security_settings.html) を参照してください。

ダウンロード・プラグインは拡張性および堅牢性が非常に高く、以下のようなことを可能にします。

- ・ユーザーによる拡張が可能なリポジトリ・リストを通じて、使用可能なリポジトリをカスタマイズする。
- ・インストールおよび依存関係解決の対象を、出荷時にサポートされているリポジトリだけでなく、すべてのリポジトリに簡単な操作で拡張できる。
- ・お客様やサービス・チームが簡単に機能を拡張できる。

これにより、大量のパッケージをダウンロードする際のパフォーマンスも向上するため、結果としてパッチ適用のターンアラウンド・タイムを短縮できます。

改訂された Rocky Linux プラグインは、以下の URL にアクセスします。

- ・<https://dl.rockylinux.org/pub/rocky>
- ・<https://dl.rockylinux.org/vault/rocky/>

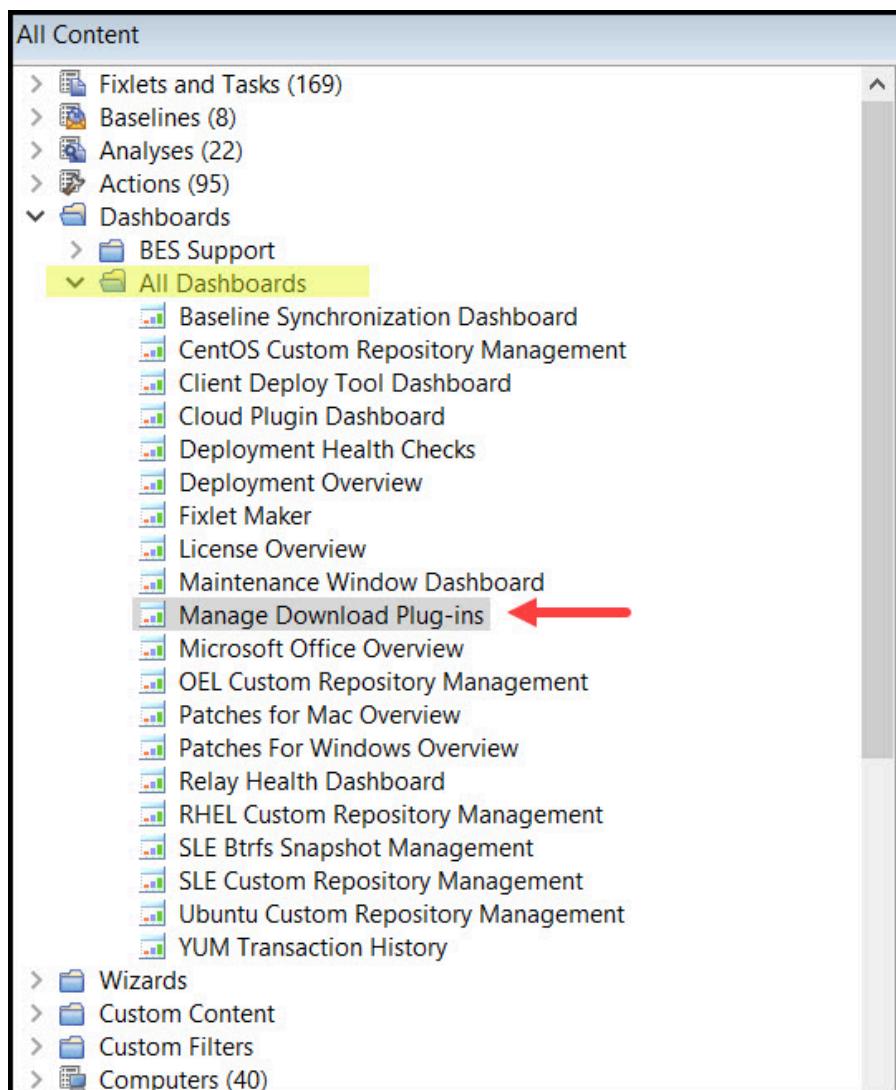
「ダウンロード・プラグインの管理」ダッシュボードの概要

「ダウンロード・プラグインの管理」ダッシュボードを使用して、デプロイメント内のダウンロード・プラグインの監視と管理を行います。

「ダウンロード・プラグインの管理」ダッシュボードを使用すると、さまざまなパッチ・ベンダーのダウンロード・プラグインの登録、登録解除、構成、およびアップグレードを実行できます。

「パッチ・サポート (Patching Support)」サイトにサブスクライブして、ダッシュボードにアクセスする必要があります。「ダウンロード・プラグインの管理」ダッシュボードを表示するには、「すべてのコンテンツ」ドメイン > 「すべてのダッシュボード (All Dashboards)」> 「ダウンロード・プラグインの管理」に進みます。

図 2. 「パッチ管理」ナビゲーション・ツリー



ダッシュボードには、デプロイメント内のすべてのサーバーとリレー (Windows のみ) が表示されます。サーバーまたはリレーを選択すると、そのコンピューターのすべてのプラグインが表示されます。ダッシュボードには、統合された 1 つのビューに、各プラグインのバージョンと状況も表示されます。

図 3. 「ダウンロード・プラグインの管理」ダッシュボード

プラグインは、以下のいずれかの状態にあります。

- ・インストールされていません
- ・新規バージョンが使用可能
- ・最新
- ・サポートされません

このダッシュボードには、ライブ・キーワード検索機能が用意されています。サーバー、リレー、およびプラグインの命名規則に基づいて検索できます。



注: ダウンロード・プラグインを BigFix リレーにインストールする場合、ダウンロードの問題を回避するためにダウンロード・プラグインを BigFix server にもインストールする必要があります。

Rocky Linux ダウンロード・プラグインの登録

「ダウンロード・プラグインの管理 (Manage Download Plug-ins)」ダッシュボードを使用して、「RockyLinux プラグイン向けパッチ (Patches for RockyLinux Plugin)」サイトから Rocky Linux パッチをインストールするための Rocky Linux ダウンロード・プラグインを登録します。

以下のタスクを実行する必要があります。

- Ensure that the BigFix server and the BigFix client are on the same version to avoid a null error.
- 「パッチ・サポート (Patching Support)」サイトにサブスクライブして、「ダウンロード・プラグインの管理」ダッシュボードにアクセスします。

- ・「**BES サポート**」サイトから使用できる「**クライアントの暗号化分析 (Encryption Analysis for Clients)**」分析をアクティブにします。
- ・「**パッチ・サポート (Patching Support)**」サイトから使用できる「**ダウンロード・プラグインのバージョン (Download Plug-in Versions)**」分析をアクティブにします。
- ・エンドポイントを暗号化する場合、「**BES サポート**」サイトから使用できる「**クライアントの暗号化の有効化 (Enable Encryption for Clients)**」Fixlet を適用します。

ダウンロード・プラグインを、そのプラグインがインストールされていないコンピューターに登録すると、プラグインは自動的にインストールされ、構成ファイルが作成されます。

ダウンロード・プラグインがすでにコンピューターにインストールされている場合、構成ファイルは上書きされます。

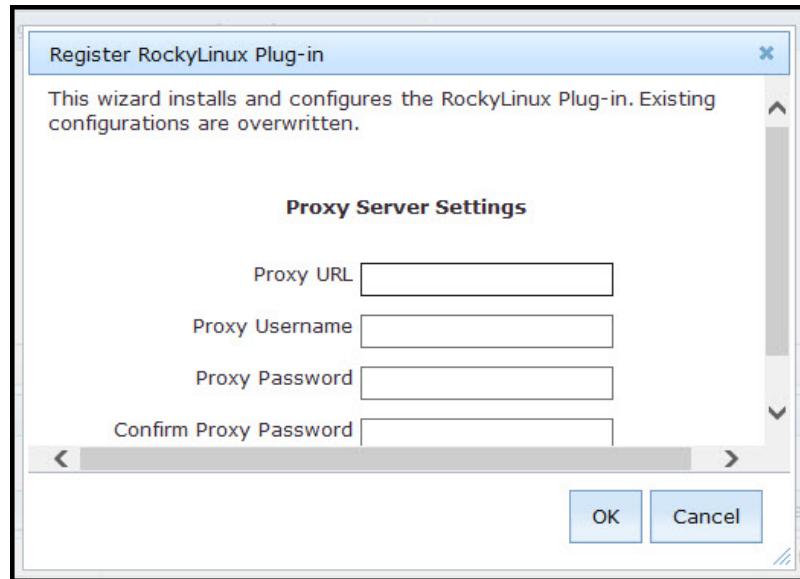
1. 「パッチ管理」ドメインから、「すべてのパッチの管理」>「ダッシュボード」>「ダウンロード・プラグインの管理」ダッシュボードをクリックします。
2. 「サーバーとリレー」テーブルから、ダウンロード・プラグインを登録するサーバーを選択します。

! **重要:** 常にダウンロード・プラグインを BigFix server に登録する必要があります。

3. 「プラグイン」テーブルから、「**RockyLinux プラグイン**」を選択します。
4. 「登録」をクリックします。

「RockyLinux プラグインの登録 (Register RockyLinux Plug-in)」ウィザードが表示されます。

図 4. 「Rocky Linux ダウンロード・プラグインの登録」ウィザード



5. プロキシー・サーバーを経由してダウンロードする必要がある場合は、プロキシー・パラメーターを入力してください。



注: 基本認証のみがサポートされています。

プロキシー URL

プロキシー・サーバーの URL。プロトコルとホスト名が含まれた整形式の URL である必要があります。この URL は通常、プロキシー・サーバーの IP アドレスまたは DNS 名とそのポートを、コロンで区切ったものです。例: `http://192.168.100.10:8080`。

プロキシー・ユーザー名

プロキシー・サーバーで認証が必要な場合のプロキシー・ユーザー名。

プロキシー・パスワード

プロキシー・サーバーで認証が必要な場合のプロキシー・パスワード。

「プロキシー・パスワードの確認」

確認用のプロキシー・パスワード。

6. 「OK」をクリックします。
「アクションの実行」ダイアログが表示されます。
7. ターゲット・コンピューターを選択します。
8. 「OK」をクリックします。

Rocky Linux ダウンロード・プラグインが正常に登録されました。

Rocky Linux ダウンロード・プラグインの基本設定の構成

「ダウンロード・プラグインの管理」ダッシュボードを使用して、Rocky Linux プラグインのプロキシー設定を構成します。



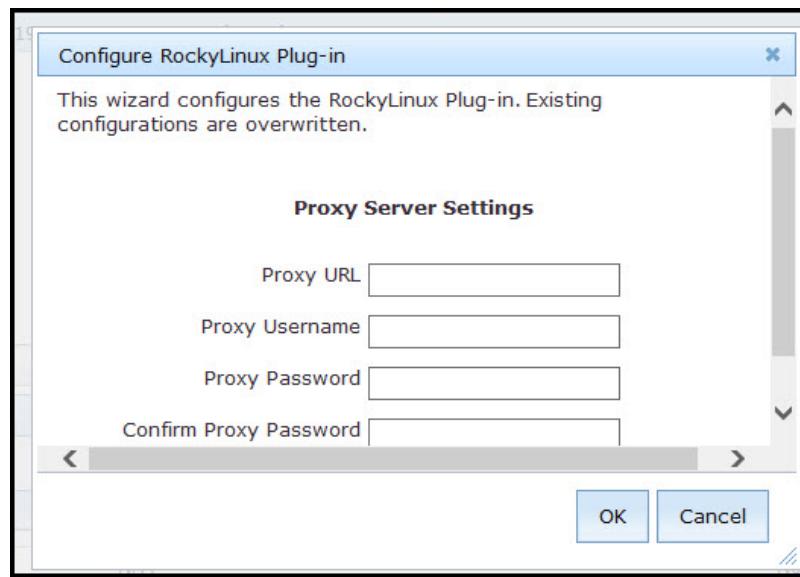
注: Ensure that the BigFix server and the BigFix client are on the same version to avoid a null error.

このタスクの適用範囲は、BigFix console から行う基本的な RockyLinux プラグインの構成に限られます。RockyLinux プラグインの詳細設定を構成するには、`plugin.ini` ファイルを編集する必要があります。この実行方法について詳しくは、[Rocky Linux ダウンロード・プラグインの詳細設定の構成 \(ページ 19\)](#)を参照してください。

ダウンロード・プラグイン用の既存の構成内容をメモしておくことをお勧めします。ダウンロード・プラグインを構成すると既存の構成が上書きされます。

1. 「パッチ管理」ドメインから、「すべてのパッチの管理」>「ダッシュボード」>「ダウンロード・プラグインの管理」ダッシュボードをクリックします。
 2. 「サーバーとリレー」テーブルから、ダウンロード・プラグインを構成するサーバーを選択します。
 3. 「プラグイン」テーブルから、「RockyLinux プラグイン」を選択します。
 4. 「構成」をクリックします。
- 「RockyLinux プラグインの構成 (Configure RockyLinux Plug-in)」ウィザードが表示されます。

図 5. 「Rocky Linux ダウンロード・プラグインの構成」 ウィザード



5. プロキシー・サーバーを経由してダウンロードする必要がある場合は、プロキシー・パラメーターを入力してください。



注: 基本認証のみがサポートされています。

プロキシー URL

プロキシー・サーバーの URL。プロトコルとホスト名が含まれた整形式の URL である必要があります。この URL は通常、プロキシー・サーバーの IP アドレスまたは DNS 名とそのポートを、コロンで区切ったものです。例: `http://192.168.100.10:8080`。

プロキシー・ユーザー名

プロキシー・サーバーで認証が必要な場合のプロキシー・ユーザー名。

プロキシー・パスワード

プロキシー・サーバーで認証が必要な場合のプロキシー・パスワード。

「プロキシー・パスワードの確認」

確認用のプロキシー・パスワード。

6. 「OK」をクリックします。
- 「アクションの実行」ダイアログが表示されます。
7. ターゲット・コンピューターを選択します。
8. 「OK」をクリックします。

Rocky Linux ダウンロード・プラグインが正常に構成されました。

Rocky Linux ダウンロード・プラグインの詳細設定の構成

詳細な構成を行うには、`plugin.ini` という Rocky Linux ダウンロード・プラグインの構成ファイルを手動で編集します。

`plugin.ini` ファイルは、「ダウンロード・プラグインの管理 (Manage Download Plug-in)」 ダッシュボードからダウンロード・プラグインが登録されたときに、自動的に作成されます。このファイルには、ロギングやキャッシングに関する設定の他、リポジトリ・リスト・ファイルを拡張するためのカスタム構成も含まれています。

Linux システムでは、このファイルはダウンロード・プラグインが格納されているルート・ディレクトリーのツリー内にあります。たとえば、`/var/opt/BESServer/DownloadPlugins/RockyLinuxProtocol` です。

Windows システムでは、このファイルは BigFix server のインストール・ディレクトリー内にあります。例えば、`%PROGRAM_FILES%\BigFix Enterprise\BES Server\DownloadPlugins\RockyLinuxProtocol` です。

図 6. Rocky Linux ダウンロード・プラグイン構成ファイルの例

```
[Logger]
file = logs/RockyLinuxPlugin.log
level = INFO

[UA]
proxy =
proxyUser =
proxyPass =

primaryRepoListFile = C:\Program Files (x86)\BigFix Enterprise\BES Server\
                     GatherDBData\gather\Patching Support\CurrentSiteData\DLRockyLinuxRepoList.json

extendedRepoListFile =
onlyUseExtendedRepoListFile = no

localCache =
localCacheOnly = no
```



注: `plugin.ini` は、大括弧で示される複数のセクションに分かれています。オプションは必ず正しいセクションで指定するようにしてください。オプションを別のセクションに移動すると、エラーが発生することがあります。

パスを指定する必要があるオプション `file`、`primaryRepoListFile`、`extendedRepoListFile`、および `localCache` では、絶対パスと相対パスのどちらでも使用できます。相対パスは、ダウンロード・プラグインの実行可能ディレクトリーを基準とする相対パスです。デフォルトでは、実行可能ファイルは `DownloadPlugins\RockyLinuxProtocol` フォルダーにあります。

ロギング・レベルの設定

ロギング・レベルにより、`RockyLinuxProtocol.log` ファイルに書き込まれる詳細情報の量が決まります。

使用可能なロギング・レベルは以下のとおりです。

ERROR

ダウンロード・プラグインの実行に関連するエラーが出力されます。このようなエラーは、致命的エラーが発生する直前であることを示している場合があります。

WARNING

ダウンロードの失敗と、失敗の理由に関する情報が出力されます。

INFO

ダウンロードの進行状況とダウンロードの成功に関する一般情報が、最小限のトレース情報とともに出力されます。

DEBUG

問題のトラブルシューティングに使用される詳細情報が出力されます。これは、使用可能なレベルの中で最も詳細なレベルです。

ロギング・レベル・オプションは、`[Logger]` ファイルの `plugin.ini` セクションで変更できます。

```
[Logger]
file = logs/RockyLinuxProtocol.log
level = INFO
```

たとえば、ロギングが INFO に設定されている場合は、そのレベルと、そのレベルの上に記載されているすべてのレベルのログが、ロガーによって出力されます。この場合は、INFO、WARNING、および ERROR の各ログが出力されます。



注: ロギング・レベルを DEBUG に設定すると、ログに記録する情報の量が増えるため、パフォーマンスに影響が及ぶ可能性があります。ロギング・レベルを DEBUG に上げるのは、問題を調査するときだけにとどめ、問題の解決後は INFO または WARNING に戻しておいてください。

拡張リポジトリ・リスト・ファイルの追加

Rocky Linux ダウンロード・プラグインは、必要に応じて、BigFix が正式にサポートしていないリポジトリと連動するように構成できます。

そのようなリポジトリをサポートするようにダウンロード・プラグインを構成する方法について詳しくは、[Rocky Linux ダウンロード・プラグインの拡張](#) (ページ 22)を参照してください。

ダウンロード・キャッシュの設定

ダウンロード・キャッシュ・ツールを使用すると、指定した場所にパッケージとリポジトリ・メタデータをダウンロードできます。

ダウンロード・キャッシャーを構成できるシナリオとして、次の 3 つが考えられます。

隔離された環境での Sha1 ダウンロード機能

ダウンロード・キャッシャー・ツールは主に、隔離された環境用に設計されています。このような環境では、セキュアなネットワークが必要するために、ベンダーのサイトから直接ファイルを直接ダウンロードするためにインターネットにアクセスすることができません。

sha1 ダウンロード機能により、BigFix server の sha1 フォルダーにパッケージが直接キャッシュされるので、パフォーマンスが向上します。

詳しくは、[隔離された環境での Rocky Linux ダウンロード・キャッシャーの使用 \(\(ページ\) 34\)](#)を参照してください。

インターネット対応 BigFix サーバーでの Sha1 ダウンロード機能

この方法は、インターネット対応の BigFix server が含まれている環境にパッケージをキャッシュする場合のベスト・プラクティスと考えられます。sha1 ダウンロード機能により、BigFix server の sha1 フォルダーにパッケージが直接キャッシュされるので、パフォーマンスが向上します。

詳しくは、[sha1 フォルダーでのパッケージのキャッシング \(\(ページ\) 34\)](#)を参照してください。

インターネット対応の BigFix server で sha1 ダウンロード機能を使用しない

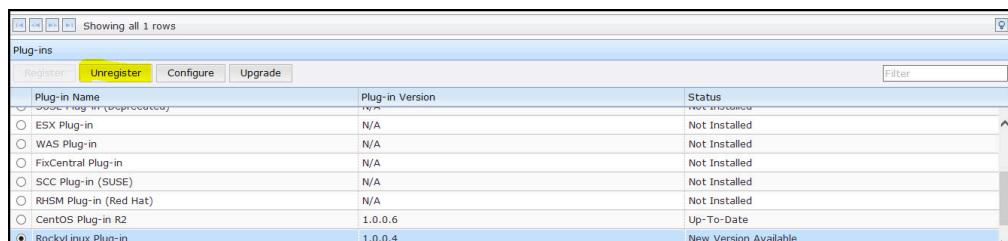
何らかの理由で、sha1 ダウンロード機能による BigFix server の sha1 フォルダーへのパッケージのキャッシングを行わないことにした場合は、ローカル・キャッシングを使用できます。詳しくは、[ローカル・キャッシング・フォルダーでのパッケージのキャッシング \(\(ページ\) 36\)](#)を参照してください。

Rocky Linux ダウンロード・プラグインの登録解除

Rocky Linux ダウンロード・プラグインの登録を解除するには、「ダウンロード・プラグインの管理 (Manage Download Plug-ins)」ダッシュボードを使用します。

1. 「パッチ管理」ドメインから、「すべてのパッチの管理」>「ダッシュボード」>「ダウンロード・プラグインの管理」ダッシュボードをクリックします。
2. 「サーバーとリレー」テーブルから、ダウンロード・プラグインを登録解除するサーバーを選択します。
3. 「プラグイン」テーブルから、「RockyLinux プラグイン」を選択します。
4. 「登録解除 (Unregister)」をクリックします。

図 7. Rocky Linux ダウンロード・プラグインの登録解除



The screenshot shows a table with columns: 'Plug-in Name', 'Plug-in Version', and 'Status'. The 'Status' column includes a 'Filter' button. The table lists several plug-ins, including 'RockyLinux Plug-in' which is currently registered (Status: Up-To-Date). A yellow box highlights the 'Unregister' button in the top navigation bar of the dashboard.

Showing all 1 rows		
Plug-ins	Register	Unregister
Plug-in Name	Plug-in Version	Status
RockyLinux Plug-in (preprocessor)	1.0.0.6	Up-To-Date
ESX Plug-in	N/A	Not Installed
WAS Plug-in	N/A	Not Installed
FixCentral Plug-in	N/A	Not Installed
SCC Plug-in (SUSE)	N/A	Not Installed
RHSM Plug-in (Red Hat)	N/A	Not Installed
CentOS Plug-in R2	1.0.0.6	Up-To-Date
RockyLinux Plug-in	1.0.0.4	New Version Available

「アクションの実行」ダイアログが表示されます。

5. ターゲット・コンピューターを選択します。
6. 「OK」をクリックします。

Rocky Linux ダウンロード・プラグインが正常に登録解除されました。

Rocky Linux ダウンロード・プラグインのアップグレード

Rocky Linux ダウンロード・プラグインをアップグレードするには、「ダウンロード・プラグインの管理 (Manage Download Plug-ins)」ダッシュボードを使用します。

1. 「パッチ管理」ドメインから、「すべてのパッチの管理」>「ダッシュボード」>「ダウンロード・プラグインの管理」ダッシュボードをクリックします。
2. 「サーバーとリレー」テーブルから、ダウンロード・プラグインをアップグレードするサーバーを選択します。
3. 「プラグイン」テーブルから、「RockyLinux プラグイン」を選択します。
4. 「アップグレード」をクリックします。
「アクションの実行」ダイアログが表示されます。
5. ターゲット・コンピューターを選択します。
6. 「OK」をクリックします。



注: ダウンロード・プラグインを再構成する必要があります。



注: 最新バージョンのダウンロード・プラグインは、プロキシー・パスワードとベンダー・パスワードの保存に関するセキュリティを増強するために強化されています。

これで、Rocky Linux ダウンロード・プラグインの最新バージョンがインストールされました。

Rocky Linux ダウンロード・プラグインの拡張

BigFix が正式にサポートしているリポジトリ以外のリポジトリからパッケージをダウンロードしてキャッシュするように、Rocky Linux ダウンロード・プラグインを構成することができます。Rocky Linux プラグインは、サポートされていない Rocky Linux リポジトリのパッケージ依存関係を解決できるため、サポートされていない依存パッケージをインストールすることができます。この拡張ソリューションにより、BigFix を使用して、出荷時にサポートされていないリポジトリからパッケージを適用できます。

操作を始める前に、以下の作業を完了しておく必要があります。

- 適切なライセンスが登録されていること、および RockyLinux サブスクリプションがアクティブであることを確認します。
- BigFix Patch for RockyLinux のサポート情報を確認し、サポートされるリポジトリのリストを検証します。
 - [サポートされるプラットフォームおよび更新 \(\(ページ\) 10\)](#)
 - [サポートされている Rocky Linux リポジトリ \(\(ページ\) 10\)](#)

BigFix で正式にサポートされていないリポジトリを使用するようにプラグインを拡張するには、以下の手順を慎重に実行してください。

1. [拡張リポジトリ・リスト・ファイルを作成する](#) ((ページ) 23)。
2. [Rocky Linux ダウンロード・プラグイン構成ファイルを更新する](#) ((ページ) 25)。
3. [拡張リポジトリのセットアップを検証する](#) ((ページ) 26)。

ステップ 1: 拡張リポジトリ・リスト・ファイルの作成

Rocky Linux プラグインでは、サブスクリプションに組み込まれている基本製品と拡張製品の両方の追加リポジトリが含まれた、拡張リポジトリ・リスト・ファイルを使用できます。

拡張リポジトリ・リスト・ファイルは、以下の基準を満たしている必要があります。

- これは有効な `.json` ファイルでなければなりません。
- Rocky Linux プラグインにアクセス可能である必要があります。管理者または root ユーザーがファイルを読み取ることができることを確認してください。
- 以下のフォーマットが含まれていること。

```
{RockyLinux Plug-in
  "<OS_key>": [
    { "name": "<name>" , "baseurl": "<base_url>" },
    { "name": "<name>" , "baseurl": "<base_url>" }
  ],
  "<OS_key_1>": [
    { "name": "<name>" , "baseurl": "<base_url>" }
  ]
}
```



注: このファイルは、`DLRockyLinuxRepoList.json` ファイルの `repoListFile` で設定される `plugin.ini` ファイルと同じ形式を使用します。

不等号括弧 `<>` で囲まれたプレースホルダーは、実際値に置き換える必要があります。

OS_key

BigFix の Rocky Linux パッチ適用コンテンツでサポートされていないリポジトリを使用するには、`DLRockyLinuxRepoList.json` ファイルにリストされている以下の OS キーを使用します。

- Rocky Linux-8-x64
- Rocky Linux-9-x64

ダウンロード・プラグインの新バージョンがリリースされているため、これは完全なリストではない可能性があります。最新版の完全なリポジトリ・リスト入手するには、以下の操作を実行します。

1. エンドポイントが最新のダウンロード・プラグインに登録されているかどうかを確認します。プラグインのダウンロード管理ダッシュボードに、プラグインが最新であるか、新しいバージョンが利用可能であるかが表示されます。
2. 以下の場所から `DLRockyLinuxRepoList.json` ファイルを表示します。

Windows システムの場合

```
%PROGRAM FILES%\BigFix Enterprise\BES Server\GatherDBData
\gather\Patching Support\CurrentSiteData
```

Linux システムの場合

```
/var/opt/BESServer/gatherDBData/gather/Patching Support/
CurrentSiteData
```



重要: ダウンロード時や依存関係の解決時に問題が起こらないようにするために、各リポジトリに対応する正しい OS キーを使用してください。

`base_url`

基本 URL は、Rocky Linux のミラーおよび Vault リポジトリ・リストから形成されます。

```
http://dl.rockylinux.org/pub/rocky/<Rocky Linux_version>/<repo_name>/<architecture>/<os>
http://dl.Rocky Linux.org/vault/rocky/<Rocky Linux_version>/<repo_name>/<architecture>
```

以下は、指定された形式の `.json` ファイルの例です。

```
{
  "Rocky Linux-8-x64": [
    {"name": "Rocky Linux - BaseOS", "baseurl": "http://dl.Rocky
Linux.org/pub/rocky/8/BaseOS/x86_64/os/"},
    {"name": "Rocky Linux - AppStream", "baseurl": "http://dl.Rocky
Linux.org/pub/rocky/8/AppStream/x86_64/os/"}
  ],
  "Rocky Linux-8.3-x64": [
    {"name": "Rocky Linux 8.3 - BaseOS", "baseurl": "http://dl.Rocky
Linux.org/vault/rocky/8.3/BaseOS/x86_64/os/"},
    {"name": "Rocky Linux 8.3 - AppStream", "baseurl": "http://dl.Rocky
Linux.org/vault/rocky/8.3/AppStream/x86_64/os/"}
  ],
  "Rocky Linux-8.4-x64": [
    {"name": "Rocky Linux 8.4 - BaseOS", "baseurl": "http://dl.Rocky
Linux.org/vault/rocky/8.4/BaseOS/x86_64/os/"},
    {"name": "Rocky Linux 8.4 - AppStream", "baseurl": "http://dl.Rocky
Linux.org/vault/rocky/8.4/AppStream/x86_64/os/"}
  ]
}
```

```

] ,
"Rocky Linux-8.5-x64": [
  {"name": "Rocky Linux 8.5 - BaseOS", "baseurl": "http://dl.Rocky
Linux.org/vault/rocky/8.5/BaseOS/x86_64/os/"} ,
  {"name": "Rocky Linux 8.5 - AppStream", "baseurl": "http://dl.Rocky
Linux.org/vault/rocky/8.5/AppStream/x86_64/os/"} ,
] ,
"Rocky Linux-8.6-x64": [
  {"name": "Rocky Linux 8.6 - BaseOS", "baseurl": "http://dl.Rocky
Linux.org/vault/rocky/8.6/BaseOS/x86_64/os/"} ,
  {"name": "Rocky Linux 8.6 - AppStream", "baseurl": "http://dl.Rocky
Linux.org/vault/rocky/8.6/AppStream/x86_64/os/"} ,
] ,
"Rocky Linux-9-x64": [
  {"name": "Rocky Linux - BaseOS", "baseurl": "https://dl.rockylinux.org/pub/rocky/9/BaseOS/x86_64/os/"} ,
  {"name": "Rocky Linux - AppStream", "baseurl": "https://dl.rockylinux.org/pub/rocky/9/AppStream/x86_64/os/"} ,
] ,
"Rocky Linux-9.1-x64": [
  {"name": "Rocky Linux - BaseOS", "baseurl": "https://dl.rockylinux.org/vault/rocky/9.1/BaseOS/x86_64/os/"} ,
  {"name": "Rocky Linux - AppStream", "baseurl": "https://dl.rockylinux.org/vault/rocky/9.1/AppStream/x86_64/os/"} ,
] ,
"Rocky Linux-9.2-x64": [
  {"name": "Rocky Linux - BaseOS", "baseurl": "https://dl.rockylinux.org/pub/rocky/9.2/BaseOS/x86_64/os/"} ,
  {"name": "Rocky Linux - AppStream", "baseurl": "https://dl.rockylinux.org/pub/rocky/9.2/AppStream/x86_64/os/"} ,
]
}

```

このファイルには「`epel_repos.json`」という名前を付け、このセクション全体の例で使用します。

ステップ 2: Rocky Linux ダウンロード・プラグイン構成ファイルの更新

拡張リポジトリ・リスト・ファイルを使用するように Rocky Linux プラグインを構成します。「ダウンロード・プラグインの管理」ダッシュボードから Rocky Linux プラグインの登録解除や構成を行うと、`plugin.ini` 構成ファイルが上書きされます。構成ファイルで行った変更はすべて失われるため、以前に行った変更を書き留めておいてください。

1. テキスト・エディターを使用して、`plugin.ini` ファイルを以下の場所から開きます。

Windows システムの場合

```
%PROGRAM FILES%\BigFix Enterprise\BES Server\DownloadPlugins
\RockyLinuxProtocol
```

Linux システムの場合

```
/var/opt/BESServer/DownloadPlugins/RockyLinuxProtocol
```

2. **extendedRepoListFile** フィールドに、拡張リポジトリー・リスト・ファイルの絶対パスまたは相対パスを入力します。相対パスに設定する場合は、Rocky Linux プラグインの実行可能プログラムの場所を基準としたパスを指定する必要があります。

例:

```
extendedRepoListFile = ./epel_repos.json
```

この例は、`epel_repos.json` ファイルがダウンロード・プラグイン・バイナリーと同じディレクトリーにあることを示しています。

3. 拡張リポジトリー・リストだけを使用するように Rocky Linux プラグインを設定するには、**onlyUseExtendedRepoListFile** フィールドを `yes` に設定します。

例:

```
onlyUseExtendedRepoListFile= yes
```

両方のリポジトリー・リスト・ファイルを使用するように Rocky Linux プラグインを設定するには、この設定を `no` に構成します。両方のファイルで同じ OS キーが使用されている場合は、リポジトリーが結合されます。

4. ファイルを保存します。

ステップ 3: 拡張リポジトリーのセットアップの検証

Rocky Linux ダウンロード・キャッシャーを使用して、拡張リポジトリーが正しくセットアップされているかどうか、およびそれが使用可能であることを検証します。キャッシャーの詳細については、[Rocky Linux ダウンロード・キャッシャーの使用について \(ページ 28\)](#) を参照してください。

これを行うには、`RockyLinuxDownloadCacher.exe --check-allrepos` コマンドを実行します。

セットアップに問題がない場合は、次のような出力になります。

```
11288 : 2023-06-07 07:28:51 : INFO      : Testing access to: rockylinux-8-x64
11288 : 2023-06-07 07:28:51 : INFO      : RockyLinux-RT
11288 : 2023-06-07 07:28:51 : INFO      : Success!
11288 : 2023-06-07 07:28:51 : INFO      : RockyLinux-Extras
11288 : 2023-06-07 07:28:51 : INFO      : Success!
```

```
11288 : 2023-06-07 07:28:51 : INFO   :
11288 : 2023-06-07 07:28:51 : INFO   : Testing access to: rockylinux-9-x64
11288 : 2023-06-07 07:28:51 : INFO   : RockyLinux-RT
11288 : 2023-06-07 07:28:51 : INFO   : Success!
11288 : 2023-06-07 07:28:51 : INFO   : RockyLinux-Extras
11288 : 2023-06-07 07:28:51 : INFO   : Success!
```

第3章. ダウンロード・キャッシュの使用

ダウンロード・キャッシャーは、必要なファイルをダウンロードしてキャッシュすることを目的として設計された、スタンドアロンのキャッシング用コマンド・ライン・ツールです。事前キャッシュされたファイルは、ダウンロード・プラグインがエンドポイントにパッチを適用するために使用することができます。

ダウンロード・キャッシャーは、隔離された環境を対象に使用することを目的として設計されています。このツールを使用して、Fixlet で必要となる大量のパッケージをダウンロードおよびキャッシュします。ファイルを事前キャッシュすることで、BigFix client にファイルを配布する前にそれらのファイルをインターネットからダウンロードする必要がなくなるため、アクションの実行速度が向上します。



注: BigFix server がインターネットにアクセスできる場合は、ダウンロード・プラグインを使用します。「ダウンロード・プラグインの管理」ダッシュボードからダウンロード・プラグインを登録する必要があります。

ダウンロードしてこのツールにアクセスし、手動で実行することができます。

Rocky Linux ダウンロード・キャッシャーの使用について

Rocky Linux ダウンロード・キャッシャーを使用すると、隔離された環境で Rocky Linux パッチをダウンロードし、キャッシュに入れることができます。このツールは、「[Rocky Linux 8 向けパッチ](#)」サイトおよび「[Rocky Linux 9 向けパッチ](#)」サイトをサポートします。

Rocky Linux ダウンロード・キャッシャーは、「<https://dl.rockylinux.org/pub/rocky>」および「<https://dl.rockylinux.org/vault/rocky/>」にアクセスして、更新用のパッケージをダウンロードします。ダウンロード・エラーを回避するため、リストされているホストでのトラフィックを許可するようにホワイトリスト、ファイアウォール、またはプロキシーを構成してください。

Rocky Linux ダウンロード・キャッシャーは、Windows システムまたは Linux システム上で実行できます。要件については、「[BigFix 10.0 - System Requirements](#)」を参照してください。

最新の Rocky Linux ダウンロード・キャッシャーは、以下の BigFix サポート・サイトから入手できます。

- Windows システムを使用する場合は、<http://software.bigfix.com/download/bes/util/RockyLinuxDownloadCacher.exe> でこのツールをダウンロードしてください。
- Linux システムを使用する場合は、<http://software.bigfix.com/download/bes/util/RockyLinuxDownloadCacher-linux.tar.gz> でこのツールをダウンロードしてください。このツールは、x86-64 (64 ビット) システムでサポートされます。



注: このツールを正しく使用するには、以下のパッケージとその依存関係を必ずインストールしてください。



- GLIBC バージョン 2.2.5 以降
- GLIBC バージョン 2.3 以降
- GNU/Linux カーネルのバージョン 2.6.31 以降

説明のために、このセクションでは、Windows での Rocky Linux ダウンロード・キャッシャーの実行手順を示します。ただし、Rocky Linux ダウンロード・キャッシャーを実行するためのパラメーターとサブコマンドは、Windows システムでも Linux システムでも同じです。

ツール `RockyLinuxDownloadCacher.exe` により、追加の操作を実行することができます。このツールをコマンド・プロンプトから実行するには、以下のコマンドを使用します。

```
RockyLinuxDownloadCacher.exe [-h] [parameters...] {subcommand} [subparameters...]
```

各コマンドの説明。

-h

コマンドを実行せずに、そのコマンドのヘルプ・メッセージを表示するよう指定します。

parameters

ダウンロード・キャッシャーを構成するために使用するオプションのパラメーターを指定します。

--proxyServer

使用するプロキシー・サーバーの URL を指定します。これは、プロトコルとホスト名を含む整形形式の URL でなければなりません。この URL は通常、プロキシー・サーバーの IP アドレスまたは DNS 名とそのポートを、コロンで区切ったものです。例:
`http://192.168.100.10:8080`

--proxyUser

プロキシー・サーバーで認証が必要な場合は、プロキシー・ユーザー名を指定します。

--proxyPass

プロキシー・サーバーで認証が必要な場合は、プロキシーのパスワードを指定します。

基本認証のみがサポートされています。

--download_dir

リポジトリ・メタデータ・ファイルがキャッシュされるディレクトリーを指定します。

このパラメーターが定義されていない場合は、ダウンロード・キャッシャーの実行可能ディレクトリーを基準とした相対ディレクトリーにファイルがダウンロードされます。

キャッシュされたファイルを使用するように Rocky Linux ダウンロード・プラグインを構成できます。これは、`localCache` ファイルに `plugin.ini` を設定することで行います。

--sha1_download_dir

パッケージを sha1 のファイル名で単一のフラット・ディレクトリーにキャッシュする場合のディレクトリーを指定します。キャッシャーは、すべてのリポジトリ (キー) からのすべてのパッケージを、指定されたディレクトリーにファイルとしてダウンロードします。

パッケージのみが `sha1_download_dir` に保管されます。各リポジトリ・メタデータは、`download_dir` に保管され、Rocky Linux リポジトリのディレクトリー構造が維持されます。

`--sha1_download_dir` サブコマンドで `check-storagered` を使用することで、スペース節約のベンチマークが設定されます。`--sha1_download_dir` を使用することで、同じ Rocky Linux バージョンの複数のリポジトリをキャッシュする際のストレージ・サイズ、ダウンロード・サイズ、および時間が大幅に削減されます。これは、同じ Rocky Linux バージョン (たとえば、rockylinux-8.3-x64、rockylinux-8.4-x64、rockylinux-8.5-x64) のリポジトリ間で、多くのパッケージが重複しているためです。Rocky Linux バージョン (たとえば、rockylinux-8.4-x64、rockylinux-8.5-x64) ごとにリポジトリを 1 つだけキャッシュする場合、スペースは節約されません。



注: このパラメーターを使用する場合は、BigFix の sha1 ファイル・フォルダーのキャッシュ制限を考慮してください。

--redownload

RPM ファイルを再ダウンロードして、ダウンロード・ディレクトリー内の既存の RPM ファイルを上書きすることを示すフラグを指定します。

このパラメーターが定義されていない場合は、RPM ファイルが再ダウンロードされません。ただし、メタデータはデフォルトでダウンロードされ上書きされます。

--verifyExistingPkgChecksum

「buildRepo」、「downloadPkg」、または「downloadbypatchid」の各サブコマンドを使用してパッケージのダウンロードを試みる際に、既存の RPM ファイルにチェックサム・チェックを強制するフラグを指定します。



注: チェックサムは、デフォルトで「オフ」に設定されています。

--loglevel

ログ・レベルを指定します。「DEBUG」、「INFO」、「WARNING」、または「ERROR」の中から選択できます。デフォルトでは、この値は「INFO」に設定されています。

INFO

ダウンロードの進行状況とダウンロードの成功に関する一般情報が、最小限のトレース情報とともにに出力されます。

WARNING

ダウンロードの失敗と、失敗の理由に関する情報が出力されます。

ERROR

ダウンロード・プラグインの実行に関連するエラーが出力されます。このようなエラーは、致命的エラーが発生する直前であることを示している場合があります。

DEBUG

問題のトラブルシューティングに使用される詳細情報が出力されます。これは、使用可能なレベルの中で最も詳細なレベルです。

--help

コマンドを実行せずに、そのコマンドの完全な説明とヘルプを表示するよう指定します。

subcommand subparameter

ダウンロード・キャッシャーを実行するために使用するサブコマンドとサブパラメーターを指定します。



注: サブコマンドとサブパラメーターの名前では、大/小文字が区別されます。

subparameter は以下のように *subcommand* によって異なります。

check-baserepos

BigFix でサポートされる Rocky Linux 基本リポジトリにアクセスできるかどうかを検査します。結果はコマンド・プロンプトに表示され、`<cacher directory>\logs\RockyLinuxDownloadCacher.log` ファイルに格納されます。

check-allrepos

BigFix でサポートされる Rocky Linux 基本リポジトリおよびサブリポジトリにアクセスできるかどうかを検査します。結果はコマンド・プロンプトに表示され、`<cacher directory>\logs\RockyLinuxDownloadCacher.log` ファイルに格納されます。

check-storagereq

`builRepo` コマンドで `--sha1_download_dir` オプションを指定する場合と指定しない場合のストレージ・スペース所要量を調べます。結果はコマンド・プロンプトに表示され、`<cacher directory>\logs\RockyLinuxDownloadCacher.log` ファイルに格納されます。

showKeys

サポートされるリポジトリの OS キーのリストを `<cacher_directory>\logs\RockyLinuxDownloadCacher.log` ファイルに出力します。OS キーは、単一 Rocky Linux リポジトリの Rocky Linux オペレーティング・システムのバージョンおよびアーキテクチャーを示します。

このサブコマンドを実行するための構文は以下のとおりです。

```
RockyLinuxDownloadCacher.exe --download_dir <download_dir>
[parameters] showsKeys
```

例: `RockyLinuxDownloadCacher.exe --download_dir C:\downloads showKeys`

buildRepo

指定された OS キーに基づいて、ローカルのミラー・リポジトリを作成し、すべての関連ファイルをダウンロードします。

このサブコマンドを実行するための構文は以下のとおりです。

```
RockyLinuxDownloadCacher.exe --download_dir <download_dir>
--sha1_download_dir <sha1_download_dir> [parameters]
buildRepo --key <OS_key1,OS_key2,...>
```

例: `RockyLinuxDownloadCacher.exe --download_dir C:\downloads --sha1_download_dir C:\sha1_downloads buildRepo --key rockylinux-8-x64`

各コマンドの説明。

--key OS_key1,OS_key2,...

Rocky Linux オペレーティング・システムのバージョンおよびアーキテクチャーを指定します。項目はコンマで区切る必要があります、スペースを含めることはできません。以下のフォーマットを使用する必要があります。

```
<product>-<version_number>-<architecture>
```

例えば、`--key rockylinux-8-x64` です。

downloadMetadataOnly

指定された OS キーのメタデータをダウンロードします。

このサブコマンドを実行するための構文は以下のとおりです。

```
RockyLinuxDownloadCacher.exe --download_dir <download_dir>
[parameters] downloadMetadataOnly --key <OS_key1,OS_key2,...>
```

例:

```
RockyLinuxDownloadCacher.exe --download_dir C:\downloads downloadMetadataOnly
--key rockylinux-8-x64
```

各コマンドの説明。

--key OS_key1,OS_key2,...

Rocky Linux オペレーティング・システムのバージョンおよびアーキテクチャーを指定します。項目はコンマで区切る必要があります、スペースを含めることはできません。以下のフォーマットを使用する必要があります。

`<product>-<version_number>-<architecture>`

例えば、`--key rockylinux-8-x64` です。

downloadPkg

指定された OS キーのリストされた RPM ファイルをダウンロードします。



注: ダウンロードするパッケージに依存関係がある場合は、代わりに `buildrepo` を使用して依存関係の問題を回避することをお勧めします。

このサブコマンドを実行するための構文は以下のとおりです。

```
RockyLinuxDownloadCacher.exe --download_dir <download_dir>
[parameters] downloadPkg --key <OS_key1,OS_key2...>
--pkg <pkg1,pkg2,...>
```

例:

```
RockyLinuxDownloadCacher.exe --download_dir C:\temp --redownload downloadPkg
--key rockylinux-8-x64 --pkg python-qrcode-core-5.0.1-1.el7.noarch.rpm
```

各コマンドの説明。

--key OS_key1,OS_key2,...

Rocky Linux オペレーティング・システムのバージョンおよびアーキテクチャーを指定します。項目はコンマで区切る必要があります、スペースを含めることはできません。以下のフォーマットを使用する必要があります。

`<product>-<version_number>-<architecture>`

例えば、`--key python-qrcode-core-5.0.1-1.el7.noarch.rpm` です。

--pkg pkg1,pkg2,...

パッケージ名を示します。

各項目はコンマで区切る必要があります、スペースを含めることはできません。例えば、`--pkg`

`liblcms1-1.17-77.12.1.x86_64.rpm,liblcms1-32bit-1.17-77.12.1.x86_64.rpm`
です。

隔離された環境での Rocky Linux ダウンロード・キャッシャーの使用

buildRepo サブコマンドを使用して 1 つのリポジトリのすべてのパッチを特定のディレクトリーにダウンロードすることにより、隔離された環境用の Rocky Linux ダウンロード・キャッシャーを使用できます。

- BigFix でサポートされる Rocky Linux 基本リポジトリとサブリポジトリにアクセスできることを確認します。確認するには、`check-allrepos` サブコマンドを実行します。
- リポジトリ・メタデータおよびパッケージをダウンロードするための十分なスペースがあることを確認します。必要なストレージ・スペースを調べるには、サブコマンド `check-storagereq` を実行します。
- 以下の手順を実行して、BigFix server の sha1 フォルダー・サイズの制限を大きくします。
 - BigFix console から、コンピューターを右クリックして、「**コンピューター設定の編集**」を選択します。
 - _BESGather_Download_CacheLimitMB** のサイズを増やします。

推奨されるサイズは、現行の BigFix server の sha1 フォルダー・サイズに `sha1_download_dir` のサイズを加えた値です。

`sha1_download_dir` のサイズを事前に決定できない場合、推奨される `sha1_download_dir` のサイズは、リポジトリごとに最小で 20GB です。最小サイズの 20GB は、時間の経過とともに増加される可能性があることに注意してください。

- `RockyLinuxDownloadCacher.exe` ファイルで `buildRepo` サブコマンドを使用して、リポジトリのすべてのファイルを指定のディレクトリーにダウンロードします。例:

```
RockyLinuxDownloadCacher.exe --download_dir C:\downloads
--sha1_download_dir C:\sha1_downloads
buildRepo --key Rocky Linux-8-x64, Rocky Linux-8.3-x64
```

この例では、ダウンロード・キャッシャー・ツールは、「Rocky Linux-8-x64」リポジトリについて、パッケージを `sha1_download_dir` にダウンロードし、リポジトリ・メタデータを `download_dir` にダウンロードします。

- `download_dir` をエラー・ギャップされた BigFix server に転送し、`sha1_download_dir` の sha1 ファイルを BigFix server の sha1 フォルダーに転送します。
- Rocky Linux ダウンロード・キャッシャーの構成ファイル `plugin.ini` を開きます。`plugin.ini` ファイルを以下の設定で構成し、BigFix server がリポジトリ・メタデータおよび必要なパッケージをオンラインでなくローカル・キャッシュから取得するように設定します。

```
localCache = <download_directory_specified_for_the--download_dir_parameter>
localCacheOnly = yes
```

sha1 フォルダーでのパッケージのキャッシング

Rocky Linux ダウンロード・キャッシャーを使用して、インターネット対応の BigFix server の sha1 フォルダーに直接パッケージをキャッシュして、Fixlet 適用中のパフォーマンスを向上させることができます。

- BigFix でサポートされる Rocky Linux 基本リポジトリとサブリポジトリにアクセスできることを確認します。確認するには、`check-allrepos` サブコマンドを実行します。
- リポジトリ・メタデータおよびパッケージをダウンロードするための十分なスペースがあることを確認します。必要なストレージ・スペースを調べるには、サブコマンド `check-storagereq` を実行します。
- 以下の手順を実行して、BigFix server の sha1 フォルダー・サイズの制限を大きくします。
 - BigFix console から、コンピューターを右クリックして、「コンピューター設定の編集」を選択します。
 - `_BESGather_Download_CacheLimitMB` のサイズを増やします。

推奨されるサイズは、現行の BigFix server の sha1 フォルダー・サイズに `sha1_download_dir` のサイズを加えた値です。

`sha1_download_dir` のサイズを事前に決定できない場合、推奨される `sha1_download_dir` のサイズは、リポジトリごとに最小で 20GB です。最小サイズの 20GB は、時間の経過とともに増加される可能性があることに注意してください。

このタスクにより、BigFix server はインターネットを活用して、リポジトリ・メタデータやパッケージなど、必要なファイルが Fixlet 適用時に必ず使用できるようにします。

- `buildRepo` ファイルで `RockyLinuxDownloadCacher.exe` サブコマンドを使用して、リポジトリから BigFix server の sha1 フォルダーにパッケージをダウンロードします。例:

```
RockyLinuxDownloadCacher.exe --download_dir C:\downloads --sha1_download_dir
C:\Program Files (x86)\BigFix Enterprise\BES Server\wwwrootbes\bfmirror\downloads\sha1
buildRepo --key rockylinux-8-x64
```

この例では、ダウンロード・キャッシャー・ツールは、BigFix server の sha1 フォルダーにパッケージを直接ダウンロードします。

- Rocky Linux ダウンロード・キャッシャーの構成ファイル `plugin.ini` を開きます。`plugin.ini` ファイルに以下の設定を構成して、BigFix server がその sha1 フォルダーからパッケージを取得するように設定します。

```
localCache =
localCacheOnly = no
```

BigFix server がインターネットに対応していて、パッケージが BigFix server の sha1 フォルダーに直接保存されている場合は、`localCache` 設定に値を指定しないでください。

BigFix server はまず、必要なファイルをインターネットから取得する前に、それらがダウンロード・プラグインのキャッシュ・フォルダーおよび sha1 フォルダーにあるかどうかを確認します。ダウンロード・プラグインのキャッシュ・フォルダー内のリポジトリ・メタデータが期限切れの場合は、新しいリポジトリ・メタデータがオンラインでダウンロードされます。必要なパッケージが BigFix server の sha1 フォルダーに存在しない場合、パッケージはインターネットからダウンロードされます。

ローカル・キャッシュ・フォルダーでのパッケージのキャッシング

Rocky Linux ダウンロード・キャッシャーを使用して、インターネット対応の BigFix server にパッケージをキャッシングできます。BigFix server がリポジトリ・メタデータおよびパッケージをキャッシング・フォルダーとインターネットの両方から使用するように構成できます。

フォルダー・サイズが増大する場合に備えて、BigFix server の sha1 フォルダーではなく別のフォルダーにパッケージを保管します。BigFix server は最新のダウンロードを保管するだけなので、BigFix server の sha1 フォルダー・サイズの制限が小さすぎると、保管されたパッケージは新しいファイルによって置き換えられます。

パッケージをローカル・キャッシュに保管することにより、Rocky Linux ダウンロード・プラグインはパッケージをインターネットから取得することなく使用できます。

BigFix server がローカル・キャッシュからのパッケージを要求する場合、パッケージは BigFix server の sha1 フォルダーにもキャッシングされるため、追加のスペースが必要になる可能性があります。

1. `buildRepo` ファイルで `RockyLinuxDownloadCacher.exe` サブコマンドを使用して、リポジトリから BigFix server の sha1 フォルダーにパッケージをダウンロードします。例:

```
RockyLinuxDownloadCacher.exe --download_dir C:\downloads
buildRepo --key rockylinux-8-x64
```

この例で、ダウンロード・キャッシャー・ツールは、指定されたダウンロード・フォルダーにパッケージをダウンロードします。

2. Rocky Linux ダウンロード・キャッシャーの構成ファイル `plugin.ini` を開きます。`plugin.ini` ファイルに以下の設定を構成します。

```
localCache = <location of the transferred download_dir>
localCacheOnly = no
```

この設定により、BigFix server は最初にダウンロード・プラグインのキャッシング・フォルダー内のリポジトリ・メタデータを確認します。これが期限切れでなければ、BigFix server はそのメタデータを使用します。それ以外の場合、BigFix server はインターネットからリポジトリ・メタデータを取得します。パッケージの場合、BigFix server は最初に、パッケージが sha1 フォルダーに存在するかどうか確認し、次に `localCache` を確認してから、インターネットに進みます。

第4章. BigFix Patch for Rocky Linux™ の使用

「Linux™ RPM パッチ」サイトおよび各種の「Rocky Linux 向けパッチ」サイトにある Fixlet を使用して、デプロイメント環境にパッチを適用します。

Rocky Linux Enterprise に使用可能な Fixlet サイトについては、[サポートされるプラットフォームおよび更新（ページ）10](#)を参照してください。

パッチ・コンテンツのキャッシングは、エラー・ギャップ環境またはカスタム・リポジトリを使用している場合を除いて、ダウンロード・プラグイン経由で行われなければなりません。詳しくは、以下のトピックを参照してください。

- [ダウンロード・プラグインの使用（ページ）12](#)
- [ダウンロード・キャッシャーの使用（ページ）28](#)

Fixlet を使用したパッチの適用

Rocky Linux Fixlet サイトから入手可能な Fixlet を使用して、Rocky Linux パッチをデプロイメントに適用できます。

- 適切な Fixlet サイトをサブスクライブします。
- 「パッチ・サポート」サイトまたは Fixlet サイトから、必要な分析（「エンドポイントの依存解決 - 適用結果」分析など）をアクティブ化して、パッチの適用結果を表示します。
- エンドポイントに GPG キーをインストールし、有効化します。 「[RPM-GPG-KEY-Rocky Linux-release のインポート](#)」タスクを使用すると、必要なキーをインポートできます。
- クライアントの `/var` ディレクトリーについて、適切な構成を設定します。
 - `/var` ディレクトリーが `noexec` オプションに設定された場合は、パッチ適用の実行可能ファイルを実行するために別のディレクトリーを指定してください。そうしないと、パッチ適用は失敗します。 「[_BESClient_LinuxPatch_executable_directory のパスの設定 \(Set the path for _BESClient_LinuxPatch_executable_directory\)](#)」Fixlet を実行し、代替ディレクトリ名を指定することができます。
 - `/var` ディレクトリーのディスク・スペースが限られている場合は、パッチ・メタデータをキャッシングできるスペースがより多いディレクトリーを指定してください。 「[_BESClient_LinuxPatch_metadata_directory のパスの設定 \(Set the path for _BESClient_LinuxPatch_metadata_directory\)](#)」Fixlet を実行し、代替ディレクトリ名を指定することができます。



注: `_BESClient_LinuxPatch_executable_directory` と `_BESClient_LinuxPatch_metadata_directory` 両方の設定で、指定したディレクトリー・パスは有効な絶対パス名でなければなりません。使用できるのは英数字、スラッシュ、および下線のみです。

- リポジトリを使用していない場合は、適切な Rocky Linux ダウンロード・プラグインを登録します。ダウンロード・プラグインについて詳しくは、[ダウンロード・プラグインの使用（ページ）12](#)を参照してください。

Rocky Linux パッチ Fixlet® を実行する場合、パッチの適用前にテスト実行を適用することもできます。「適用の結果」分析を表示して、依存関係が正常に解決したかどうか、およびインストールが成功するかどうかを判断できます。

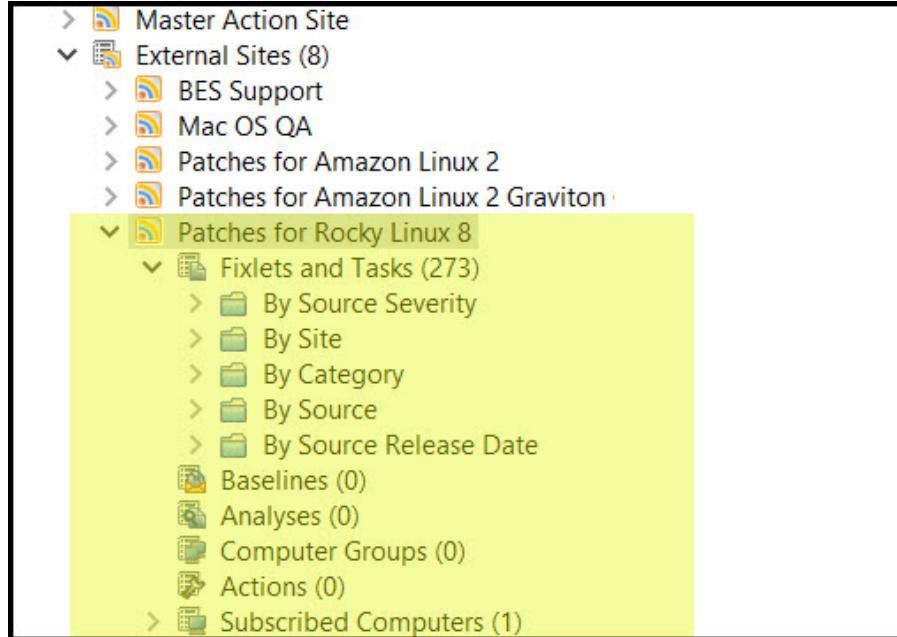


注: テスト実行機能を使用すると、テストに失敗した場合でも、アクションは「修正済み」と報告します。

kernel Fixlet には、すべての kernel パッケージをアップグレードまたはインストールするオプションが用意されています。アップグレード・オプションを使用すると、既存の kernel パッケージがそれより後のバージョンで置き換えられます。インストール・オプションでは、以前のバージョンの次に新しい kernel パッケージがインストールされます。kernel 更新のデフォルトの動作は、パッケージの横並びでのインストールです。また、各 kernel 更新 Fixlet® には、これらの各オプションをテストする機能が用意されています。

1. 「パッチ管理」ドメインから、「OS ベンダー」>「Rocky Linux」をクリックし、ドメイン・ノードを使用してパッチ・コンテンツに移動します。

図 8. 「パッチ管理」ナビゲーション・ツリー



2. リスト・パネルに表示されたコンテンツの中から、適用する Fixlet を選択します。作業域で Fixlet が開きます。
3. Fixlet の詳細を確認するには、ウィンドウの上部にあるタブをクリックします。
4. 「アクションの実行」をクリックして、Fixlet を適用します。



注: パッケージ・ソースとなるリポジトリを使用している場合は、Fixlet アクションを実行する前に、「カスタム・リポジトリ・サポートの有効化 - RockyLinux」タスクを実行します。このタスクにより、YUM を使用してエンドポイント上の構成済みリポジトリから必要なダウンロードを実行するためのアクションが構成されます。カスタム・リポジトリ・サポートを有効にする

 と、Fixlet は Bigfix インフラストラクチャーによるメタデータおよびパッケージのダウンロードを停止し、必要なファイルを YUM でダウンロードできるようにします。

「アクション」ボックスで、以下の該当するリンクをクリックすることもできます。

- デプロイメント・プロセスを開始できます。
 - パッチを適用する前にテスト実行を適用できます。「**エンドポイントの依存解決 - 適用結果**」分析を表示して、依存関係が正常に解決したかどうか、およびインストールが成功するかどうかを判断します。
 - 特定の Fixlet の Red Hat セキュリティー情報を表示することができます。パッチ・ページを表示するには、「**こちらをクリックして、パッチ ページを表示**」アクションを選択します。
5. 「アクションの実行」ダイアログでは、追加のパラメーターを設定することができます。
- 「アクションの実行」ダイアログを使用したパラメーターの設定について詳しくは、『[BigFix コンソール・オペレーター・ガイド](#)』を参照してください。
6. 「OK」をクリックします。

置き換え

置き換えについて詳しくは、「Windows 以外での置き換え ((ページ))」を参照してください。

モジュラー型 Fixlet を使用したパッチ

このトピックでは、Rocky Linux でモジュラー Fixlet をデプロイする方法について説明します。

モジュラー型 Fixlet は、Rocky Linux 8 と Rocky Linux 9 で使用できます。

Rocky Linux 用モジュラー型 Fixlet は、RHEL と同じように適用できます。Rocky Linux モジュラーの適用については、「モジュラー型 Fixlet を使用したパッチ ((ページ))」を参照してください。

第5章. 複数パッケージのベースラインのインストール

BigFix Patch には、ベースラインに含まれる複数パッケージの更新のインストールを单一のタスクに結合するためのソリューションが用意されており、それによってベースラインの実行時間を短縮できます。

ベースラインを使用すると、複数の Fixlet をグループにまとめて、任意の組み合わせのターゲット・コンピューターにすぐに適用することが可能になります。これは、特定のアクション・グループをネットワーク全体に適用するための強力な方法です。ただし、ベースライン内の各 Fixlet は、ベースラインの実行時に、個別の DNF 更新トランザクションを作成します。単一のベースラインで膨大な数の DNF 呼び出しが行われる可能性があり、その場合はすべてのトランザクションが完了するまでに時間がかかるため、パフォーマンスに深刻な影響が及ぶことがあります。

「複数パッケージのベースラインのインストール (multiple-package baseline installation)」ソリューションを使用すると、依存関係の解決やパッケージのインストールが Fixlet ごとに個別に行われることに伴うパフォーマンスの低下に対処できます。このソリューションでは、ベースラインの先頭でこの機能を有効化して、関連パッケージを单一の DNF 呼び出しからインストールするためのインストール・タスクを付加する必要があります。

パッケージをインストールするのではなく、リストに追加するよう Fixlet に指示するフラグを設定するには、「複数パッケージのベースラインのインストール機能を有効化 - Rocky Linux 8」タスクを使用します。このフラグは、ベースラインの完了後にクリアされます。複数のパッケージを単一のコマンドからインストールできるようにするには、該当するタスクをベースラインの先頭に追加する必要があります。



注:

- Rocky Linux Vault に保管されているパッケージが含まれる Fixlet (Fixlet タイトルにラベル (`(vault x.y)`) で示される) は、複数パッケージのベースラインのインストール機能を使用して適用することはできません。このような Fixlet は、個別に適用するか、または通常のベースラインで適用する必要があります。
- 複数パッケージのベースラインのインストール機能では、「アクションの実行」の事前キャッシュ・オプション 「すべての制約を満たす前のダウンロードの開始 (Start download before all constraints are met)」 はサポートされません。

複数パッケージのインストール・タスクは、それぞれの Rocky Linux オペレーティング・システムのバージョンおよびアーキテクチャーに対して有効化されます。依存関係の解決を行い、パッケージをダウンロードして、それらをエンドポイントにインストールするには、該当するインストール・タスクをベースラインの最後に追加する必要があります。



重要: 「複数パッケージのベースラインのインストール機能を有効化」タスクおよび「複数パッケージのベースラインのインストール」タスクは、同じベースライン内に存在する必要があります。

Patches for Rocky Linux 8 サイトから利用可能:

複数パッケージのベースラインのインストール - Rocky Linux 8 - x86_64

ベースライン全体の依存関係の解決とパッケージのインストールを単一のインスタンスで実行するには、これらのタスクがベースラインの最後に実行される必要があります。

インストールの予行演習を行うことで、パッケージに対する変更をプレビューして、意図しないパッケージの更新によって発生する可能性のある依存関係の破損を防ぐこともできます。テスト・アクションは、`/var/opt/BESClient/EDRDeployData` にある以下のファイルに結果を出力します。

PkgToInstallList.txt file

このファイルには、依存関係チェック後にインストールされるパッケージが記載されます。

PkgToRemoveList.txt file

このファイルには、ターゲット Rocky Linux エンドポイントから削除されるパッケージが記載されます。

BigFix Patch には、インストールを促進するために、以下のコンテンツも用意されています。

複数パッケージのベースラインのインストール用に Rocky Linux 8 パッケージ・リスト・ファイルを削除

ターゲットの Rocky Linux 8 コンピューターからパッケージ・リスト・ファイルを削除します。

パッケージの依存関係の解決に関連して発生する可能性のある問題を防ぐために、このタスクをベースラインの開始時に追加します。

複数パッケージのインストール機能の使用について詳しくは、[ベースラインでの複数パッケージのインストール \(\(ページ\) 41\)](#)を参照してください。

ベースラインでの複数パッケージのインストール

複数パッケージのベースラインのインストール機能を使用すると、複数の固有パッケージを含む Fixlet を 1 つのベースラインから適用するときに、時間を短縮することができます。

- クライアント・ダウンロード事前キャッシュ・サイズを 2GB 以上に設定します。`_BESClient_Download_PreCacheStageDiskLimitMB` 設定を使用して、事前キャッシュの推奨サイズを設定してください。
- クライアント・ダウンロード・キャッシュ・サイズを 2GB 以上に設定します。`_BESClient_Download_DownloadsCacheLimitMB` 設定を使用して、キャッシュの推奨サイズを設定してください。
- 各エンドポイントの空きディスク・スペースが 4GB 以上あることを確認します。
- エンドポイントに GPG キーをインストールし、有効化します。Fixlet サイトから入手可能な「**RPM-GPG-KEY-rockylinux-release のインポート (Import RPM-GPG-KEY-rockylinux-release)**」タスク (ID# 301) を使用すると、必要なキーをインポートできます。

単一の DNF 呼び出しを実行することによってベースラインのすべての Fixlet のパッケージをインストールまたは更新するには、その機能を有効化するタスクと、適切な複数パッケージのベースラインのインストール・タスクを、ベースラインに追加する必要があります。使用可能なタスクのリストについては、[複数パッケージのベースラインのインストール \(\(ページ\) 40\)](#)を参照してください。



注:

- Rocky Linux 8 の場合、ベースラインの Fixlet を置き換えるとエラーが発生してインストールが失敗するため、インストールを続行する前にすべてのベースラインを同期する必要があります。最新の使用可能なパッケージのみがベースラインとともにデプロイされます。
- 依存関係が破損したパッケージは、可能な限りスキップされます。ただし、パッケージをスキップできず、インストールがキャンセルされる場合があります。既知のケースを以下にいくつか示します。
 - Rocky Linux の依存関係に関する問題が含まれるパッケージ。
 - 依存関係エラーのあるパッケージはインストール中に発生します。これは通常、以下のエラー・メッセージによって示されます。 `File conflicts happen when two packages attempt to install files with the same name but different contents.`
- タイトルに `(vault x.y)` が含まれる Fixlet は、複数パッケージのインストールのベースライン内では動作しません。
- 複数パッケージのベースラインのインストール機能では、「アクションの実行」の事前キャッシュ・オプション「すべての制約を満たす前のダウンロードの開始 (Start download before all constraints are met)」はサポートされません。



重要: この機能が動作するためには、「複数パッケージのベースラインのインストール機能を有効化」タスクおよび「複数パッケージのベースラインのインストール」タスクを、同じベースライン内に追加する必要があります。さらに、タスクと Fixlet の順序がベースライン・インストールにとって重要であるため、手順を正しい順序で実行するように注意してください。

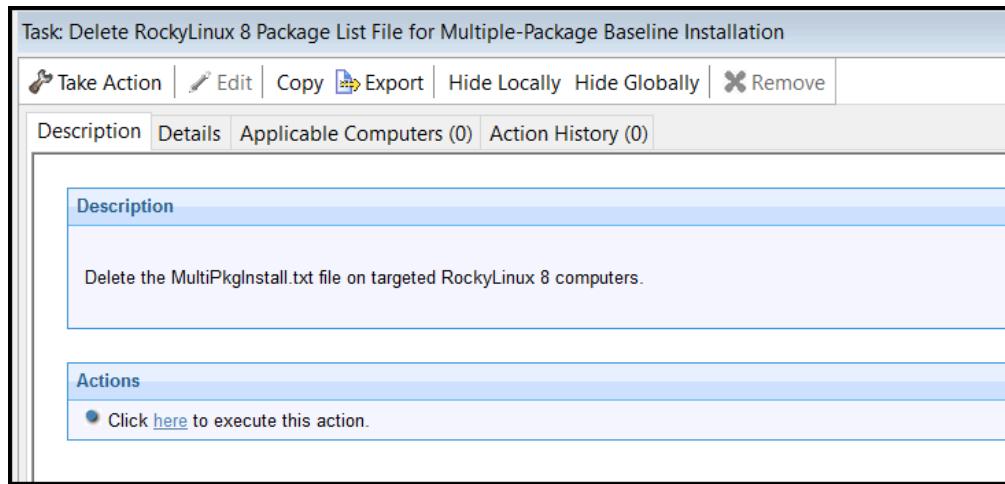
1. ベースラインを作成します。

Fixlet サイトの Fixlet を強調表示して、コンテキスト・メニューから「新しいベースラインに追加」を選択します。「ツール」メニューから「ベースラインの新規作成」を選択することもできます。

2. オプション: Fixlet を追加する前に「複数パッケージのベースラインのインストール用に Rocky Linux 8 パッケージ・リスト・ファイルを削除」タスクを追加します。

「このコンポーネントが該当する場合はベースラインも適用可能なコンピューターに該当する (Baseline will be relevant on applicable computers where this component is relevant)」オプションが選択されていないことを確認します。

図 9. ベースライン・コンポーネントのオプション



3. 該当する「複数パッケージのベースラインのインストール機能を有効化 (Enable the Multiple-Package Baseline Installation feature)」タスクを追加します。
「このコンポーネントが該当する場合はベースラインも適用可能なコンピューターに該当する (Baseline will be relevant on applicable computers where this component is relevant)」オプションが選択されていないことを確認します。
 4. 目的のパッチ Fixlet をベースラインに追加します。
すべての Fixlet について、「このコンポーネントが関連する適用可能なコンピューター上では、ベースラインが関連します」オプションが選択されていることを確認します。
-  **注:** 同一パッケージの異なるバージョンに影響する複数の Fixlet をベースラインに追加すると、インストール・タスクは古い方のバージョンのパッケージをスキップし、最新バージョンのみインストールします。
5. 該当する「複数パッケージのベースラインのインストール (Multiple-Package Baseline Installation)」タスクを、ベースラインの最後に追加します。このタスクにより、以下のいずれかのアクションを適用できます。
 - 発生する可能性のある問題を確認するために、パッケージを実際にインストールすることなくインストールのプレビューを実行する。
 - すべての RPM を单一の DNF トランザクションでインストールする。
 「このコンポーネントが該当する場合はベースラインも適用可能なコンピューターに該当する (Baseline will be relevant on applicable computers where this component is relevant)」オプションが選択されていないことを確認します。

ベースラインを実行する前に、以下の要件を満たしておく必要があります。

- エンドポイントで登録されているリポジトリに、対象パッケージと必要なすべての依存パッケージが格納されている必要があります。
- 同じ Fixlet を個別に適用する前に、複数パッケージのインストール方式を使用している Fixlet が、すべての DNF トランザクションを完了し、エンドポイントのステータスを更新するための十分な時間を空けてください。
- 同一エンドポイントに対して、同一サイトから複数のベースラインを実行しないでください。
- 技術情報に記載されたベースラインのベスト・プラクティスに従ってください。 https://hclpnpssupport.service-now.com/csm?id=kb_article&sys_id=d288c2021b098c9477761fc58d4bcfdf



注: ベースラインを適用すると、すべてのパッチ Fixlet の最初のサブアクション・ステータスに、Fixlet が失敗したことが示されます。これは予想された動作です。ベースラインでパッケージをダウンロードおよびインストールするプロセスは、Fixlet アクション・レベルで実行されるのではなく、「**複数パッケージのベースラインのインストール (Multiple-Package Baseline Installation)**」タスクで実行されます。ベースラインが完了すると、Fixlet のベースライン・サブアクション・ステータスに、各パッチ・インストールの最終状態が反映されます。

第6章. カスタム・リポジトリの管理

Rocky Linux 向けのパッチを管理するために、カスタム・リポジトリをセットアップできます。このソリューションでは、適用環境全体で複数のリポジトリが許可されます。

カスタム・リポジトリのサポートにより、Rocky Linux ネイティブ・ツール・サイトの Fixlet で DNF を使用してカスタム・リポジトリから直接パッケージをダウンロードできます。dl.rockylinux.org/pub と dl.rockylinux.org/vault でパッケージ・ソースにアクセスする必要はありません。カスタム・リポジトリ・アーキテクチャーでは、帯域幅スロットリングはサポートされません。

カスタム・リポジトリを使用すると、ご使用の適用環境のエンドポイントに適用できる内容を柔軟に制御できます。たとえば、カスタム・リポジトリ内でホストしているカスタム・ソフトウェアを適用し、その後、「パッチ・サポート」サイトから「DNF を使用したパッケージのインストール」タスクを使用して、ソフトウェアをエンドポイントにインストールすることができます。詳しくは、「[カスタム・リポジトリからのパッケージのインストール \(ページ 49\)](#)」を参照してください。

「Rocky Linux カスタム・リポジトリ管理」ダッシュボードを使用すると、カスタム・リポジトリ・ソリューションを簡単に統合できます。ただし、このダッシュボードでは、物理リポジトリは追加されません。このアクションは別個に行う必要があります。リポジトリの作成方法については、https://docs.rockylinux.org/gemstones/setup_local_repo を参照してください。



重要: ダウンロード・エラーを回避するために、カスタム・リポジトリを使用して Rocky Linux ダウンロード・プラグインを登録する必要があります。「ダウンロード・プラグインの管理 (Manage Download Plugins)」ダッシュボードでダウンロード・プラグインを登録できます。

「Rocky Linux カスタム・リポジトリ管理」ダッシュボード

「Rocky Linux カスタム・リポジトリ管理」ダッシュボードを使用すると、既存のリポジトリ・ソリューションを BigFix パッチ管理ソリューションと簡単に統合できます。

「Rocky Linux カスタム・リポジトリ管理」ダッシュボードを使用すると、Fixlet は、標準の BigFix ダウンロード・インフラストラクチャーを使用する代わりに、DNF を使用してダウンロードを実行することができます。このダッシュボードでは、エンドポイントにパッケージをインストールする際に、リポジトリを登録して DNF コマンドを使用することもできます。

このダッシュボードにアクセスするには、「パッチ・サポート」サイトをサブスクライブします。「パッチ管理ドメイン」から、「すべてのパッチの管理」 > 「ダッシュボード」 > 「Rocky Linux カスタム・リポジトリ管理」ダッシュボードをクリックします。「リポジトリ構成 - Rocky Linux」分析をアクティブにして、ダッシュボード上でコンテンツを表示します。



重要: このダッシュボードを使用する前に、必要なメタデータとヘッダーを使用してカスタム・リポジトリを事前構成する必要があります。

「Rocky Linux カスタム・リポジトリ管理」ダッシュボードを使用して、パッチ管理用の以下のアクションを実行します。

- リポジトリへのエンドポイントの登録と登録解除
- リポジトリ・ダッシュボード・リストでのローカル・リポジトリの追加、削除、およびインポート



注: 「Rocky Linux カスタム・リポジトリ管理」ダッシュボードでは、物理的なリポジトリ・サーバーの作成はサポートされません。このリポジトリは、個別に作成する必要があります。リポジトリの作成方法については、以下のリソースを参照してください。https://docs.rockylinux.org/gemstones/setup_local_repo/.



重要: ダウンロード・エラーを回避するために、カスタム・リポジトリを使用して Rocky Linux ダウンロード・プラグインを登録する必要があります。「ダウンロード・プラグインの管理 (Manage Download Plugins)」ダッシュボードでダウンロード・プラグインを登録できます。

リポジトリの追加

リポジトリを登録してエンドポイントに接続できるようにするために、「Rocky Linux カスタム・リポジトリ管理」ダッシュボードを使用して、リポジトリをリポジトリ・リストに追加します。

「リポジトリ構成 - Rocky Linux」分析をアクティブ化して、ダッシュボードにエンドポイントおよびリポジトリの情報を取り込みます。

- 「Rocky Linux カスタム・リポジトリ管理」ダッシュボードで「リポジトリ」タブをクリックします。
- 「追加」をクリックします。
- 「新規リポジトリの追加 (Add a New Repository)」ダイアログで、以下のフィールドに値を入力します。
 - リポジトリ名
 - リポジトリ URL



注: リポジトリ設定がリポジトリ・サーバー構成と一致することを確認する。

- 「保存」をクリックします。

追加したリポジトリをエンドポイントに接続するには、[リポジトリへの Rocky Linux エンドポイントの登録 \(\(ページ\) 47\)](#)を参照してください。

エンドポイントの既知のすべての既存リポジトリをこのダッシュボードのリポジトリ・リストに追加する場合は、インポート機能を使用します。詳しくは、「[リポジトリのインポート \(\(ページ\) 48\)](#)」を参照してください。

リポジトリへの Rocky Linux エンドポイントの登録

「Rocky Linux カスタム・リポジトリ管理」ダッシュボードを使用すると、リポジトリをエンドポイントにパッチを適用するときのパッケージ・ソースとして追加できます。

- リポジトリ設定がリポジトリ・サーバー構成と一致することを確認する。
- 「リポジトリ構成 - Rocky Linux」分析をまだアクティブ化していない場合はアクティブ化します。

- 「Rocky Linux カスタム・リポジトリ管理 (Rocky Linux Custom Repository Management)」ダッシュボードで「エンドポイント」タブをクリックします。
- 最初のテーブルから、リポジトリに登録するエンドポイントを選択します。選択したエンドポイントのリポジトリが2番目のテーブルにリストされます。



注: 名前が「未指定」になっているリポジトリは、ダッシュボードのリポジトリ・リストにリストされません。

- 「新規リポジトリの登録 (Register a new repository)」をクリックします。
- 「新規リポジトリの登録 (Register a New Repository)」ダイアログで、リポジトリを選択して、「次へ」をクリックします。
- 後続のウィンドウに、エンドポイントの登録先リポジトリの名前と URL が表示されます。
- オプション: 「追加フィールド (Additional Fields)」**に詳細な構成情報を追加できます。
ベンダー・サイトの単なるミラーではないカスタム・リポジトリを持つユーザーは、「追加フィールド (Additional Fields)」に gpgcheck=0 を追加する必要があります。gpg シグニチャー・ファイルが除外されると、rpm ファイルに対して認証チェックが行われず、インストールが失敗する可能性があります。
- 「保存」をクリックします。この情報は、DNF 構成ファイルに保存されます。
- 「アクションの実行」ダイアログで、コンピューターを選択し、「OK」をクリックしてアクションを適用します。

選択されたエンドポイントのパッケージ・ソースとしてリポジトリが正常に割り当てされました。

パッチ適用時にこのリポジトリを使用するには、「カスタム・リポジトリ・サポートの有効化 - Rocky Linux」タスクを実行します。

リポジトリからの Rocky Linux エンドポイントの登録解除

「Rocky Linux カスタム・リポジトリ管理」ダッシュボードを使用して、関連性がなくなったリポジトリからエンドポイントを登録解除します。

リポジトリを登録解除すると、DNF 構成ファイルは削除されず、無効化のみが行われます。また、「Rocky Linux カスタム・リポジトリ管理」ダッシュボードは、選択したコンピューターからシステム ID ファイルを削除します。

1. 「Rocky Linux カスタム・リポジトリ管理 (Rocky Linux Custom Repository Management)」ダッシュボードで「エンドポイント」タブをクリックします。
2. リポジトリを登録解除するエンドポイントを選択します。
3. 「新規リポジトリの登録解除 (Unregister a new repository)」をクリックします。
4. 「新規リポジトリの登録解除 (Unregister a New Repository)」ダイアログで、リポジトリを選択して「保存」をクリックします。
5. 「アクションの実行」ダイアログで、コンピューターを選択し、「OK」をクリックしてアクションを適用します。

リポジトリの削除

ダッシュボードのリポジトリ・リストの管理を容易にするために、適用環境にもはや存在していないリポジトリを削除します。

1. 「Rocky Linux カスタム・リポジトリ管理」ダッシュボードで「リポジトリ」タブをクリックします。
2. 削除するリポジトリを選択し、「削除」をクリックします。削除を確認するダイアログが表示されます。
3. 「はい」をクリックして確認し、選択したリポジトリの削除を進めます。

選択したリポジトリがリストから削除されます。

リポジトリのインポート

「Rocky Linux カスタム・リポジトリ管理」ダッシュボードのインポート機能を使用して、エンドポイントの既知のすべての既存リポジトリをこのダッシュボードのリポジトリ・リストに追加します。

「リポジトリ構成 - Rocky Linux」分析をアクティブ化して、ダッシュボードにエンドポイントおよびリポジトリの情報を取り込みます。

既存のリポジトリをインポートする場合は、`.repo` ファイルで以下のエントリーが以下の順序で入力されているようにしてください。

```
name=
baseurl=
enabled=
gpgcheck=
```

1. 「Rocky Linux カスタム・リポジトリ管理」ダッシュボードで「リポジトリ」タブをクリックします。
2. 「インポート」をクリックします。
3. 「既存のリポジトリのインポート (Import Existing Repositories)」ダイアログで、ダッシュボードのリポジトリ・リストに追加するリポジトリを選択します。
4. リポジトリの名前を入力します。
5. 「保存」をクリックします。

これでリポジトリがインポートされ、ダッシュボードのリポジトリ・リストに追加されました。

DNF のパッケージ更新の確認

「DNF: 使用可能なパッケージ更新の確認」タスクを使用して、Rocky Linux エンドポイントに、インストール用の適用可能な DNF パッケージ更新を識別して判別します。

「DNF: 使用可能なパッケージ更新の確認」タスクは、`dnf check-update` コマンドを使用して、エンドポイントにインストールされているパッケージに使用可能な更新を判別します。これは、「パッチ・サポート」サイトで使用可能なタスクです。

このタスクを実行する前に、DNF ログ分析 (ID #28) がアクティブ化されていることを確認します。エンドポイントに適用可能な DNF パッケージ更新の結果の詳細は、分析内の「DNF check-update 出力 (YUM check-update output)」という名前の新規の列にリストされます。

このタスクは適用環境内のリポジトリを使用するため、このタスクの実行時に DNF パッケージがリポジトリ内で使用可能であることを確認します。

カスタム・リポジトリからのパッケージのインストール

BigFix には、カスタム・リポジトリに登録されている Rocky Linux のエンドポイントで、パッケージのインストールと更新を簡単に実行できるタスクが用意されています。

- 「パッチ・サポート」サイトをサブスクライブして、「DNF を使用したパッケージのインストール」という名前のインストール・タスクにアクセスします。
- 「Rocky Linux カスタム・リポジトリ管理」ダッシュボードで、カスタム・リポジトリを構成します。詳しくは、「[Rocky Linux カスタム・リポジトリ管理](#) ダッシュボード (ページ 45)」を参照してください。
- 「ダウンロード・プラグインの管理 (Manage Download Plug-ins)」ダッシュボードで Rocky Linux ダウンロード・プラグインを登録します。ダウンロード・プラグインによって生成されたデータは使用されない場合がありますが、ダウンロード・エラーを回避するためにプラグインを登録することが重要です。
- 構成したリポジトリが最新の状態になっていて、必要なパッケージとメタデータが含まれていることを確認します。

エンドポイントでパッケージのインストールや更新を行うには、「DNF を使用したパッケージのインストール」タスクを使用します。

インストール用に選択したパッケージを指定するには、パッケージ名または Common Vulnerabilities and Exposures (CVE) ID 番号を使用します。

カスタム・リポジトリ内の有効な新しいバージョンを使用して、エンドポイントにインストールされているすべてのパッケージを更新することもできます。

使用可能な各アクションの DNF コマンドを以下に示します。

`dnf install <package_name1> <package_name2>`

特定の名前のパッケージを更新またはインストールします。複数のパッケージの更新やインストールを行うことができます。パッケージ名を区切るために、スペースを使用してください。

dnf update

エンドポイントのすべてのインストール済みパッケージを更新します。

dnf patch --cve=<cve_number>

特定の CVE ID 番号を持つパッケージを更新します。CVE ID 番号が指定されておらず、単一の CVE 参照だけが許可されている場合、このコマンドは失敗します。

dnf patch --date=<YYYY-MM-DD>

指定した日付までに発行されたすべてのパッチをインストールします。

`dnf install` コマンドと `dnf update` コマンドの場合のみ、コマンド・オプションが追加のフラグとしてサポートされます。詳しい使用方法については、`dnf` の man ページを参照してください。

このタスクには、インストール用のパッケージをテストするためのアクションも用意されています。テストを実行するためにパッケージをエンドポイントにインストールする必要はありません。

1. 「パッチ管理」ドメインから、「すべてのパッチの管理」 > 「Fixlet とタスク」をクリックします。
2. 「DNF を使用したパッケージのインストール」タスクを選択して、エンドポイントにカスタム・パッケージをインストールします。
3. 「タスク」ペインで説明を確認し、「アクション」ボックスの指示に従ってアクションを適用します。
4. 選択したアクションに応じて必要な情報を入力し、「OK」をクリックします。



注: エンドポイントにインストールされているすべてのパッケージを更新するには、パッケージをインストールするためのアクションを選択します。ただし、パッケージ名は指定しないでください。

5. 「アクションの実行」ペインで、パッケージのインストール先となるエンドポイント、または更新するパッケージが存在するエンドポイントを選択します。
6. 「OK」をクリックします。

付録 A. サポート

この製品について詳しくは、以下のリソースを参照してください。

- [BigFix サポート・ポータル](#)
- [BigFix Developer](#)
- [YouTube の BigFix プレイリスト](#)
- [YouTube の BigFix Tech Advisors チャネル](#)
- [BigFix フォーラム](#)

付録 B. トラブルシューティング

Rocky Linux エンドポイントへのパッチ適用時に問題が発生した場合は、ログ・ファイルを確認して、発生した問題およびエラーの修正方法を判別します。

ログ・ファイル

トラブルシューティングの精度を上げるために、エラー報告とエラー処理をより明確に示すことで、ロギングを強化しています。

RockyLinuxPlugin.log

Rocky Linux ダウンロード・プラグインの実行に関連するダウンロードの結果が記録されます。情報量はロギング・レベルによって異なります。

このログの場所は以下のとおりです。

- Windows システムの場合: `%PROGRAM FILES%\BigFix Enterprise\BES Server\DownloadPlugins\RockyLinuxProtocol`
- Linux システムの場合: `/var/opt/BESServer/DownloadPlugins/RockyLinuxProtocol`

以下のログ・ファイルが、ディレクトリー `/var/opt/BESClient/EDRDeployData` のクライアント・フォルダーにあります。

EDR_DeploymentResults.txt

EDR 適用の結果と DNF 出力が記録されます。

register-repo.log

「Rocky Linux カスタム・リポジトリ管理」ダッシュボードのリポジトリ登録アクションの実行結果が記録されます。

unregister-repo.log

「Rocky Linux カスタム・リポジトリ管理」ダッシュボードのリポジトリ登録解除アクションの実行結果が記録されます。

dnf_history.log

「DNF トランザクション履歴」ダッシュボードからの実行結果が記録されます。

その他の有用なログ・ファイル:

dnf.log

これは DNF がデフォルトで `/var/log/dnf.log` に生成する正式なログです。DNF 関連のすべての操作およびトランザクションが記録されます。

ダウンロード・プラグインのロギング・レベル

ロギング・レベルは、RockyLinux ダウンロード・プラグインがログ・ファイルに書き込む詳細情報の量を決定します。%PROGRAM_FILES%\BigFix Enterprise\BES Server\DownloadPlugins\RockyLinuxProtocol\plugin.ini ファイルでロギング・レベルを設定します。



注: ロギング・レベルの値では、大/小文字が区別されます。

以下のロギング・レベルは、ログに記録される情報量が少ない順にリストされています。

ERROR

ダウンロード・プラグインの実行に関連するエラーが出力されます。このようなエラーは、致命的エラーが発生する直前であることを示している場合があります。

WARNING

ダウンロードの失敗と、失敗の理由に関する情報が出力されます。

INFO

ダウンロードの進行状況とダウンロードの成功に関する一般情報が、最小限のトレース情報とともに出力されます。

DEBUG

問題のトラブルシューティングに使用される詳細情報が出力されます。これは、使用可能なレベルの中で最も詳細なレベルです。



注: ロギング・レベルを DEBUG に設定すると、ログに記録する情報の量が増えるため、パフォーマンスに影響が及ぶ可能性があります。ロギング・レベルを DEBUG に上げるのは、問題を調査するときだけにとどめる必要があります。

クライアントのデバッグ・ログを有効にする

「パッチ・サポート」サイトから、「Linux パッチ適用のデバッグ・ログを有効化するようにクライアント設定を構成する (Configure the client setting to enable the Debug Log for Linux Patching)」Fixlet (ID #57) を使用します。

プリフェッチ・プラグインのエラー

アクション・スクリプトの `execute prefetch plug-in` が含まれる行で失敗した Fixlet でアクションを実行した場合、同じプリフェッチ・プラグインに対するすべてのアクション・スクリプトからのそれ以降の呼び出しが、そのエンドポイントで失敗する可能性があります。スクリプトがブラックリストに登録された可能性があり、プリフェッチ・プラグインがエラーになります。

確認するには、クライアント・ログを調べます。プリフェッチ・プラグインを実行する Fixlet アクションに対して、以下のメッセージのいずれかが見つかります。

```
execute prefetch plug-in' didn't complete within 300 seconds. Black listing plug-ins
matching the sha1 hash of 'name of 'bash' until agent is restarted.
```

```
Execute prefetch plug-in attempting to reuse plug-in which took too long earlier.
```

この問題を解決するには、以下のアクションを実行します。

1. BigFix client を再始動して、ブラックリストをクリアします。
2. `_BESClient_ActionManager_PrefetchPlugInTimeoutSeconds` クライアント構成設定に、パッチで依存関係をインストールして解決するための十分な時間を設定します。このクライアント設定は、クライアントがスクリプトをブラックリストに登録する前に待機する時間を示します。パッチ・サポート・サイトで使用できる「**リファレンス・プラグインのタイムアウトの変更 (Change Timeout for Prefetch Plugins)**」タスクを使用して、この設定を 30 分 (1800 秒) に設定できます。



注: `_BESClient_ActionManager_PrefetchPlugInTimeoutSeconds` 設定は、エンドポイントおよびインストール中の Fixlet によって異なります。最適な値を求めるには、最も遅いエンドポイントで設定を 3,000 秒などの高い値に設定し、大きな Fixlet を実行して所要時間を確認します。その時間に 2 を乗算した値を使用できます。あるいは、推奨値ではうまくいかない場合は、クライアント設定を 600 秒に設定し、適宜調整していきます。

/var を noexec としてマウントした場合のエラー

使用可能なすべての Fixlet は、デフォルトでは、エンドポイント上のパーティションである `/var` ディレクトリーから直接実行される実行可能ファイルを使用します。`/var` が `noexec` オプションを使用して設定されている場合、Rocky Linux ダウンロード・プラグイン・ソリューションの使用の有無にかかわらず、Fixlet は動作しません。そのため、以下の手順を実行して、`/var` ディレクトリーが `noexec` オプションを使用して設定されないようにする必要があります。

1. クライアント・ログを確認して、リファレンス・プラグインから `exit code 126` が返されているかどうかを調べます。例:
2. root ユーザーとして `mount` を実行し、現在使用されているマウント・オプションを確認します。

```
[root@host ~]# mount
/dev/mapper/vg_data-lv_root on / type ext4 (rw)
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
devpts on /dev/pts type devpts (rw,gid=5,mode=620)
tmpfs on /dev/shm type tmpfs (rw)
/dev/sdal on /boot type ext4 (rw,nodev)
/dev/mapper/vg_data-lv_var on /var type ext4 (rw,noexec,nosuid,nodev)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
```

`/var` が `noexec` に設定されている場合は、以下のいずれかのアクションを実行する必要があります。

- `noexec` マウント・オプションを解除する。
- `/var/opt/BESClient` を `noexec` でない別のパーティションに移動し、そこへのシンボリック・リンクを元の場所に作成する。
- 「`_BESClient_LinuxPatch_executable_directory` のパスの設定 (Set the path for `_BESClient_LinuxPatch_executable_directory`)」 Fixlet を実行して、パッチ適用の実行可能ファイルを実行するための代替ディレクトリーを指定します。ディレクトリー・パスは、有効な絶対パス名でなければなりません。使用できるのは英数字、スラッシュ、および下線のみです。

GPG キーがない

適用時の問題を回避するためには、パッチ Fixlet を適用する前に Fixlet サイトから入手可能な 「**RPM-GPG-KEY-rockylinux-release のインポート (Import RPM-GPG-KEY-centos-release)**」 タスク (ID# 301) を適用します。

リポジトリ・メタデータが大きすぎる

パッチ適用で使用されるリポジトリ・メタデータは、ベンダーによって提供されるものであり、サイズが大きい場合があります。`/var` ディレクトリーにメタデータを保管するための十分なスペースがない場合は、そのエンドポイント上でメタデータを保管できるだけの十分なスペースを持つ代替ディレクトリーを設定します。 「`_BESClient_LinuxPatch_metadata_directory` のパスの設定 (Set the path for `_BESClient_LinuxPatch_metadata_directory`)」 を使用すると、リポジトリ・メタデータが作成されるディレクトリーを設定できます。

BigFix Patch ダウンロード・プラグインの構成時に Null エラーが発生する

BigFix server の BigFix server および BigFix client が同じバージョンでない場合、Null エラーが発生する可能性があります。このエラーが発生するのは、BigFix server 8.x バージョンと 9.x バージョンが異なる方法で暗号化を処理するためです。BigFix server 上のクライアントのバージョンを使用して BigFix server のバージョンが判別され、BigFix server と BigFix server 上のクライアントのバージョンが同じであると想定されます。

ダウンロード・プラグインを構成する際、Null エラーを回避するために BigFix server の BigFix server と BigFix client のバージョンが一致していることを確認してください。少なくとも、バージョンは 8.x や 9.x など、同じメジャー・バージョン・レベルになければなりません。

付録 C. よくある質問

BigFix Patch for Rocky Linux をより深く理解するために、以下の質問と回答をお読みください。

「ダウンロード・プラグインの管理」ダッシュボードはデータを反映していません。どうすればよいでしょう。

この問題のトラブルシューティングのために実施できる手順を以下に示します。

- ・最新の「パッチ・サポート」サイトを収集します。
- ・「パッチ・サポート」サイトから使用できる「ダウンロード・プラグインのバージョン」分析をアクティブにします。
- ・BigFix console キャッシュをクリアします。

置き換えられるパッチとは何ですか？

置き換えられる Fixlet とは、古いパッケージが含まれた Fixlet のことです。Fixlet® が置き換えられるごとに、新しいバージョンのパッケージが含まれた新規 Fixlet® が存在するようになります。新規 Fixlet® の ID は、置き換えられた Fixlet® の説明で確認できます。

適用ログはエンドポイントのどこにありますか？

ログは、`EDRDeployData` にあるクライアント・フォルダー内の `/var/opt/BESClient/EDRDeployData` という名前のフォルダーにあります。

アクションがダウンロードの失敗として報告されるのはなぜですか？

ダウンロード・プラグインが最新バージョンに更新されており、正しい資格情報で登録されていることを確認してください。

アクションがインストール失敗のレポートを返した場合はどうすればよいですか？

競合がベンダー提供のパッケージによって発生したのかどうかを調べてください。インストールを実行するには、それらの競合を削除する必要があります。

`/var` ディレクトリーが `noexec` としてマウントされているエンドポイントの Fixlet にパッチを適用できません。どうすればよいでしょう。

回避策については、[よくある質問（ページ 56）](#)を参照してください。

ダウンロード・プラグインが正しく登録されているかどうかは、どのように確認すればよいですか？

ダウンロード・プラグインが正しく登録されているかどうかを確認するには、アクション・タスクと共に、Fixlet を実行します。パッチのダウンロードが成功したことを確認します。成功していない場合は、ダウンロード・プラグインを登録解除してから再登録することが必要な場合があります。

ダウンロード・プラグインを登録する方法を教えてください。ダウンロード・プラグインの登録タスクまたは「ダウンロード・プラグインの管理」ダッシュボードのどちらを使用すればよいですか？

ダウンロード・プラグインを登録するには、「Patching Support」サイトの「ダウンロード・プラグインの管理」ダッシュボードを使用する必要があります。既存のダウンロード・プラグインの登録タスクは使用しないでください。プラグインの登録について詳しくは、[『よくある質問（ページ 56）』](#)を参照してください。



注: ダウンロード・プラグインの登録解除、構成、およびアップグレードを行う際にも、「ダウンロード・プラグインの管理」ダッシュボードを使用する必要があります。既存のダウンロード・プラグインの登録解除および編集タスクは使用しないでください。ダッシュボードについて詳しくは、[BigFix Knowledge Center](#) の『「ダウンロード・プラグインの管理」ダッシュボード』のトピックを参照してください。

どのバージョンの DNF ネイティブ・ツールを使用する必要がありますか?

Rocky Linux ネイティブ・ツール用パッチ・サイトでは、バージョン 3.2.19-18 以降が必要です。

アクションが失敗し、ログには DNF 固有のエラーが表示されています。失敗したアクションはどのようにトラブルシューティングすればよいですか?

DNF および関連するエラーについて詳しくは、<http://yum.baseurl.org> の YUM の資料および Red Hat Customer Portal の DNF 関連の記事を参照してください。

以前に構成したリポジトリを再び構成できますか?

はい、以前に構成したことのあるリポジトリを再び構成できます。

ログによって、リポジトリに対する通常の DNF プロセスを使用しているかどうか分かりますか?

はい、ログには、リポジトリ内での通常の DNF プロセスが使用されているかどうかが記録されます。

リポジトリの登録とリポジトリのインポートの違いは何ですか?

ダッシュボードの「リポジトリ」リストに含まれていない既存のリポジトリがある場合は、インポート機能を使用します。「リポジトリ」リストにリポジトリがすでにあります場合でも、そのリポジトリをエンドポイントにリンクする必要がある場合は、登録機能を使用します。

リポジトリにパッケージが含まれていないとどうなりますか?

パッケージが見つからない場合、Fixlet は失敗します。DNF 出力のログが記録される `EDR_DeploymentResult.txt` からトラブルシューティングを実行できます。

「タスク: YUM を使用したパッケージのインストール (Task: Install packages by using YUM)」を使用して DNF を使用して複数のパッケージをインストールできますか?

はい、このタスクで複数のパッケージをインストールできます。各 rpm 名をスペースで区切ってください。

「DNF トランザクション履歴」ダッシュボードがサポートする DNF のバージョンは何ですか?

最小でも 3.2.28 の DNF バージョンが必要です。

「DNF トランザクション履歴」ダッシュボードがサポートする Rocky Linux のバージョンは何ですか?

このダッシュボードは、Rocky Linux バージョン 8 をサポートします。

「DNF トランザクション履歴」ダッシュボードの「ロールバック」と「元に戻す」の違いは何ですか?

ロールバック・コマンドは、指定したトランザクションの時点までのトランザクションをすべて取り消します。元に戻すコマンドは、選択したトランザクションのみを元に戻します。

DNF トランザクション履歴ログと DNF ログ分析の違いは何ですか?

Patch Management for Rocky Linux は、「DNF トランザクション履歴」ダッシュボードで実行されるアクションの結果を記録する DNF トランザクション履歴ログを生成します。ログは、`/var/opt/BESClient/EDRDeployData/DNF_history.log` にあります。

DNF ログは、DNF がデフォルトで `/var/log/dnf.log` に生成する正式なログです。デフォルトの場所を変更するには、`/etc/dnf.conf` のログ・ファイルの設定を変更します。

「DNF トランザクション履歴」ダッシュボードに表示されるのは `dnf -update all` だけですか？

`dnf -update all` だけでなく、ダッシュボードの「コマンド・ライン」列に、以下のインストール・コマンドなどのさまざまなトランザクションが表示されます。

- `install bzip2`
- `install net-tools`
- `install vim enhance`
- `install wget`

DNF コマンドについて詳しくは、Red Hat Enterprise Linux の Web サイト (<https://access.redhat.com/>) を参照してください。

BigFix client が DNF リポジトリを使用するように構成されている場合、「DNF トランザクション履歴」ダッシュボードはどのように動作しますか？

BigFix client が DNF リポジトリを使用するように構成している場合でも、「DNF トランザクション履歴」ダッシュボードが適用環境に悪影響を及ぼすことはありません。ローカルの DNF リポジトリを既に使用している場合は、ロールバック用のパッケージを指定する方が簡単な場合があります。

クライアント・ログに、Fixlet が正常に完了しないプリフェッチ・プラグイン・エラーが表示されています。エラーの原因は何でしょうか。どうすればよいでしょう。

エンドポイントで実行されていた ActionScript がブラックリストに登録され、プリフェッチ・プラグインの問題が発生した可能性があります。

この問題を解決するには、BigFix client を再始動してブラックリストをクリアしてください。スクリプトがブラックリストに登録されないように、`_BESClient_ActionManager_PrefetchPlugInTimeoutSeconds` クライアント構成設定に、パッチを処理するための十分な時間を設定してください。詳しくは、「[よくある質問 \(ページ 56\)](#)」を参照してください。

Rocky Linux ダウンロード・プラグインのログ・レベルを設定することはできますか？

必要な情報のレベルに応じたログ・メッセージを生成するように、ダウンロード・プラグインを設定することができます。

ロギング・レベルは、Rocky Linux プラグインがログ・ファイルに書き込む詳細情報の量を決定します。`%PROGRAM_FILES%\BigFix Enterprise\BES Server\DownloadPlugins\RockyLinuxProtocol\plugin.ini` ファイルでロギング・レベルを設定します。



注: ロギング・レベルの値では、大/小文字が区別されます。

以下のロギング・レベルは、ログに記録される情報量が少ない順にリストされています。

ERROR

ダウンロード・プラグインの実行に関連するエラーが出力されます。このようなエラーは、致命的エラーが発生する直前であることを示している場合があります。

WARNING

ダウンロードの失敗と、失敗の理由に関する情報が出力されます。

INFO

ダウンロードの進行状況とダウンロードの成功に関する一般情報が、最小限のトレース情報とともにに出力されます。

DEBUG

問題のトラブルシューティングに使用される詳細情報が出力されます。これは、使用可能なレベルの中で最も詳細なレベルです。



注: ロギング・レベルを DEBUG に設定すると、ログに記録する情報の量が増えるため、パフォーマンスに影響が及ぶ可能性があります。ロギング・レベルを DEBUG に上げるのは、問題を調査するときだけにとどめる必要があります。

Rocky Linux ダウンロード・プラグインのログはどこにありますか? 使用できるログ・レベルにはどのようなものがありますか?

ロギングは `plugin.ini` ファイルによって制御されます。このファイルは、ダウンロード・プラグインの実行可能プログラムと同じ場所にあります。デフォルトでは、Windows システム上の `%PROGRAM FILES%\BigFix Enterprise\BES Server\DownloadPlugins\RockyLinuxProtocol` にあります。Linux システムの場合は、`/var/opt/BESServer/DownloadPlugins/RockyLinuxProtocol` にあります。ログ・ファイルは 1 日ごとに循環して使用されます。つまり、新規ログ・ファイルが作成されると、古いログ・ファイルの名前がその作成開始日に変更されます。

新バージョンのダウンロード・プラグインがインストールされると、Rocky Linux ダウンロード・プラグインの構成ファイル (`plugin.ini`) は上書きされますか?

いいえ。構成ファイルが上書きされることはありません。構成ファイルが上書きされるのは、ダウンロード・プラグインが再構成されたときだけです。

Rocky Linux ダウンロード・プラグインの登録後に、プロキシーを再構成することはできますか?

はい。「ダウンロード・プラグインの管理」ダッシュボードからダウンロード・プラグインを構成することで、プロキシー設定を更新できます。

DLRockyLinuxRepoList.json ファイルを編集してリポジトリを追加するとどうなりますか?

BigFix が「パッチ・サポート (Patching Support)」サイトを更新すると、このファイルが上書きされるため、行った変更はこのときに削除されます。

拡張製品のサブスクリプションがあります。各製品に割り当てられたリポジトリにアクセスするように Rocky Linux ダウンロード・プラグインを構成できますか？

はい、できます。詳しくは、「[Rocky Linux ダウンロード・プラグインの拡張（（ページ） 22）](#)」を参照してください。

拡張リポジトリ・リスト・ファイルはどこに保存すればよいですか？

ダウンロード・プラグインがアクセスできる場所であれば、どこに保管しても構いません。BigFix server がその場所に対するアクセス権を保持していることを確認する必要があります。

Rocky Linux プラグインにアップグレードしたため、パッケージをインストールできません。すべてのタスクの結果、次のような行が表示されます。「Failed add prefetch item {concatenation ";" of lines of file (parameter "EDR_PkgRequest")}」。何が問題なのでしょうか？

BigFix 拡張セキュリティー・オプション `-requireSHA256Downloads` または BigFix 管理ツールの「**SHA-256 ダウンロードが必要**」オプションが有効になっている可能性があります。このオプションは、すべてのダウンロード検証で SHA-256 アルゴリズムのみを使用するように構成します。Rocky Linux ダウンロード・プラグインは、プラグインで使用される、リポジトリ内のパッケージについての SHA-256 値が含まれていない特定の Rocky Linux リポジトリ・メタデータが原因で失敗する場合があります。

パッチを正常にデプロイするために、「**SHA-256 ダウンロードが必要**」オプションを無効にすることを検討してください。パッケージの GPG シグニチャーを使用して別の層の検査および検証が実行されるため、セキュリティーおよびパッケージの整合性が低下することはありません。ダウンロード・オプションについて詳しくは、BigFix Platform インストール・ガイド (https://help.hcl-software.com/bigfix/9.5/platform/Platform/Installation/c_security_settings.html) を参照してください。

拡張リポジトリ・リストだけを使用するように Rocky Linux ダウンロード・プラグインを構成できますか？

はい、`onlyUseExtendedRepoListFile` の `plugin.ini` フラグを「`yes`」に設定します。

ベースライン内の複数の Fixlet を適用するときに、破損した依存関係をスキップして、残りのパッケージのインストールを続行することはできますか？

「**複数パッケージのベースラインのインストール (Multiple-Package Baseline Installation)**」タスクは、依存関係が破損したパッケージを可能な限りスキップします。ただし、Rocky Linux の依存関係に関する問題が含まれるパッケージはスキップできません。パッケージがスキップされないもう 1 つのシナリオは、インストール中に依存関係エラーが発生した場合です。これは、次のエラー・メッセージで示されます。`File conflicts happen when two packages attempt to install files with the same name but different contents.` このような場合は、インストールが取り消され、エンドポイントにパッチがインストールされません。

複数パッケージのベースラインのインストール方式を使用したときに障害が発生する原因として考えられるものは何ですか？

Fixlet のインストールに失敗した理由は以下の可能性があります。

- ・カスタム・サイトに、「複数パッケージのベースラインのインストール (multiple-package baseline installation)」タスクを同時に実行するベースラインが複数存在していた。
- ・複数の Fixlet で、同一パッケージの複数のバージョンを更新するよう要求された。
- ・複数の Fixlet で、同一のパッケージ依存関係を更新するよう要求された。
- ・ベースラインが複数パッケージのインストール方式を実行した直後に Fixlet が適用された。複数パッケージのインストールで、すべての DNF トランザクションを完了し、エンドポイントのステータスを更新するための十分な時間が取られなかった。
- ・DNF ツールにパッチが適用されていないのに、ベースラインに Rocky Linux リリース・パッケージが含まれている。
- ・「**複数パッケージのベースラインのインストール機能を有効化 (Enable the Multiple-Package Baseline Installation feature)**」タスクは、残りのコンテンツと同じベースライン内にある必要があります。このタスクは、パッチ Fixlet および複数パッケージのインストール・タスクより前に追加する必要があります。
- ・複数パッケージのベースラインのインストール機能は、有効化タスクとインストール・タスクの両方が同じベースライン内に存在する場合にのみ動作します。詳しくは、「[ベースラインでの複数パッケージのインストール \(ページ 41\)](#)」を参照してください。
- ・以下のクリーンアップ・タスクのいずれかの後に、「**複数パッケージのベースラインのインストール機能を有効にする**」タスクを追加する必要があります。複数パッケージのベースラインのインストール用に Rocky Linux 8 パッケージ・リスト・ファイルを削除 (Delete Rocky Linux 8 Package List File for Multiple-Package Baseline Installation) またはトラブルシューティング: Rocky Linux 8 パッチ・デプロイメント・ログ - クリーンアップ (Rocky Linux 8 Patching Deployment Logs - Cleanup)
- ・正しい Rocky Linux ディストリビューション、オペレーティング・システム・バージョン、サービス・パック・レベル、およびアーキテクチャーを使用するインストール・タスクをベースラインの最後に追加する必要があります。

エンドポイントが隔離された環境にあります。これらのエンドポイントにパッチを適用するには、BigFix をどのように構成すればよいですか？

隔離された環境の場合は、エンドポイントにパッチを適用するために必要なパッケージをホストするサポート対象 Rocky Linux リポジトリに対して、必ずミラーリングを行ってください。これを行うには、Rocky Linux ダウンロード・キャッシュを使用して、BigFix server からアクセスできる場所にローカル・リポジトリを作成します。この場所は、ローカル・キャッシュと呼ばれます。



注: 依存関係の解決時に問題が起こらないように、ローカル・キャッシュには必要なリポジトリがすべて含まれている必要があります

適用時にファイルをダウンロードする際にローカル・キャッシュを使用するには、ダウンロード・プラグインの構成ファイル `plugin.ini` を構成します。以下の手順に従ってください。

1. ローカル・キャッシュの構成 `localCache` を、ダウンロード・キャッシャー・ツールを使用してダウンロードしたファイルの場所に設定します。



注: この場所は、BigFix server からアクセス可能でなければなりません。

2. ダウンロード・キャッシュからのファイルのみをダウンロードし、ベンダーのサイトからはダウンロードしないよう `localCacheOnly` フラグを `yes` に設定します。

「複数パッケージのベースラインのインストール (Multiple-Package Baseline Installation)」タスクを使用してベースラインを実行しました。そのベースラインで失敗した Fixlet のリストを確認するには、どうすればよいですか？

「アクション情報の表示」ダイアログを使用すると、ベースライン適用の全体的な進行状況をモニターや、各サブアクションの詳細なステータスを確認することができます。

このダイアログにアクセスするには、以下の手順を実行します。

1. ナビゲーション・ツリーで「アクション」アイコンをクリックします。
2. アクション・リスト・パネルでアクションを選択します。
3. 作業域で「コンピューター」タブを選択します。
4. リストの任意のコンピューターを右クリックします。
5. コンテキスト・メニューから「アクション情報の表示」を選択するか、「編集」メニューから「アクション情報の表示」を選択します。

失敗した Fixlet に関する詳細情報を調べるには、ターゲット・エンドポイントの `/var/opt/BESClient/_BESData/_Global/Logs` にあるクライアント・ログを確認してください。

複数パッケージのインストール・タスクを使用して実行したベースラインが正常に完了しましたが、まだ関連するものとして表示されるのはなぜですか？

依存関係が破損したパッケージを、Fixlet コンポーネントがインストールできなかったことが原因と考えられます。 「複数パッケージのベースラインのインストール (Multiple-Package Baseline Installation)」タスクは、デフォルトでは、依存関係に問題がないパッケージがターゲット・エンドポイントに正常に適用されるように、破損した依存関係を無視します。

サポートされる Rocky Linux リポジトリにアクセスできるかどうかを確認する方法はありますか？

コマンド・ラインから `--check-baserepos` コマンドと `--check-allrepos` コマンドを使用して、サポート対象の Rocky Linux リポジトリにアクセスできるかどうかを確認できます。これらのコマンドについて詳しくは、[Rocky Linux ダウンロード・キャッシャーの使用について \(\(ページ\) 28\)](#)を参照してください。

BigFix がサポートする Rocky Linux リポジトリ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。

Rocky Linux ダウンロード・キャッシャー・ツールを使用してリポジトリ・メタデータおよびパッケージをキャッシングするときに必要なスペースを調べる方法はありますか？

ストレージ・スペース所要量を調べるには、`check-storagereq` サブコマンドを使用します。

Rocky Linux ダウンロード・キャッシャー・ツールを使用してリポジトリ・メタデータおよびパッケージをキャッシングするときに、スペースを節約するにはどうすればよいですか？

`--sha1_download_dir` を使用することで、スペース節約のベンチマークが設定されます。

`--sha1_download_dir` を使用することで、同じ Rocky Linux バージョンの複数のリポジトリをキャッシュする際のストレージ・サイズ、ダウンロード・サイズ、および時間が大幅に削減されます。これは、同じ Rocky Linux バージョン（たとえば、rockylinux-8.3-x64、rockylinux-8.4、rockylinux-8.5）のリポジトリ間で、多くのパッケージが重複しているためです。Rocky Linux バージョン（たとえば、rockylinux-8.4、rockylinux-8.5）ごとにリポジトリを 1 つだけキャッシュする場合、スペースは節約されません。

出力例を以下に示します。

```
Caching rockylinux-8.3-x64, rockylinux-8.4, rockylinux-8.5
(with --sha1_download_dir):
Total Repo Metadata and Packages will take up 28 GB of space
instead of 37 GB (23% space saved)
```

EDR ログに、何もインストールできないため、最新のカーネルを使用しているかどうか確認するよう警告するメッセージが表示され、Fixlet のインストールが失敗しました。この場合の対処方法を教えてください。

このメッセージは、カーネル・パッケージをデプロイする Fixlet の場合にのみ表示されます。エンドポイントにターゲット・カーネル・パッケージがインストールされていない場合、またはエンドポイントのアクティブ・カーネルがターゲット・カーネル・パッケージより低いバージョンである場合、カーネル Fixlet が関連状態になります。エンドポイントに最新のカーネルがインストールされているもののアクティブに使用していない場合も、カーネルの脆弱性の対象と見なされます。

To remediate the issue, restart the endpoint and ensure it is using the latest kernel available.

Notices

This information was developed for products and services offered in the US.

HCL may not offer the products, services, or features discussed in this document in other countries. Consult your local HCL representative for information on the products and services currently available in your area. Any reference to an HCL product, program, or service is not intended to state or imply that only that HCL product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any HCL intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-HCL product, program, or service.

HCL may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

*HCL
330 Potrero Ave.
Sunnyvale, CA 94085
USA*

Attention: Office of the General Counsel

For license inquiries regarding double-byte character set (DBCS) information, contact the HCL Intellectual Property Department in your country or send inquiries, in writing, to:

*HCL
330 Potrero Ave.
Sunnyvale, CA 94085
USA*

Attention: Office of the General Counsel

HCL TECHNOLOGIES LTD. PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. HCL may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-HCL websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this HCL product and use of those websites is at your own risk.

HCL may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

*HCL
330 Potrero Ave.
Sunnyvale, CA 94085
USA
Attention: Office of the General Counsel*

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by HCL under terms of the HCL Customer Agreement, HCL International Program License Agreement or any equivalent agreement between us.

The performance data discussed herein is presented as derived under specific operating conditions. Actual results may vary.

Information concerning non-HCL products was obtained from the suppliers of those products, their published announcements or other publicly available sources. HCL has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-HCL products. Questions on the capabilities of non-HCL products should be addressed to the suppliers of those products.

Statements regarding HCL's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to HCL, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. HCL, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS," without warranty of any kind. HCL shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work must include a copyright notice as follows:

© (your company name) (year).

Portions of this code are derived from HCL Ltd. Sample Programs.

Trademarks

HCL Technologies Ltd. and HCL Technologies Ltd. logo, and hcl.com are trademarks or registered trademarks of HCL Technologies Ltd., registered in many jurisdictions worldwide.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other product and service names might be trademarks of HCL or other companies.

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions.

Applicability

These terms and conditions are in addition to any terms of use for the HCL website.

Personal use

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of HCL.

Commercial use

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of HCL.

Rights

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

HCL reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by HCL, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

HCL MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.