BigFix
構成ガイド

Special notice

Before using this information and the product it supports, read the information in Notices (on page 497).

Edition notice

This edition applies to BigFix version 10 and to all subsequent releases and modifications until otherwise indicated in new editions.

第 1 章. 概要

このガイドでは、インストール後に環境で実行できる追加の構成手順について説明します。

BigFix 10 プラットフォームの新機能

BigFix 10 プラットフォームは、新機能と拡張機能を提供します。

パッチ 8

BigFix エージェントのサポートを追加

Rocky Linux 8 x86 64 ビットで稼働する BigFix エージェントのサポートが追加されました。

ライブラリーのアップグレード

- libcurl ライブラリーは、バージョン 7.84.0 にアップグレードされました。
- libssh2 ライブラリーは、バージョン 1.10 にアップグレードされました。

パッチ 7

ネットワークに基づく直接ダウンロードの有効化

特定のサブネットに接続されている BigFix クライアントにのみ直接ダウンロードを許可できるようになりました。

詳しくは、ダウンロードの管理 (（ページ） 393)を参照してください。

リレー・スイッチ後のダウンロード再開

リレー・スイッチで進行中のダウンロードを中断できるようになりました。

詳しくは、ダウンロードの管理 (（ページ） 393)を参照してください。

強化されたサイト Rest API によるサイト表示名と NMO 権限の表示

BigFix Platform 10.0.7 ではサイト Rest API が強化され、BigFix コンソールでのサイト表示名で構成される新しいエレメントを返すようになりました。サイト Rest API で、指定されたサイトに対する要求者の権限も表示できるようになりました。

詳しくは、『サイト』を参照してください。

VMware Cloud プラグインを使用した VM カスタム属性の取得

BigFix Platform 10.0.7 以降、VMware プラグインは、現在取得可能なプロパティーに加えて VM カスタム属性も取得できるようになりました。この情報は、BigFix コンソールと WebUI に表示されます。

詳しくは、クラウド分析データ (（ページ） 278)を参照してください。

クライアント認証

最新の業界標準に準拠するため、BigFix エージェントのクライアント証明書の期限は 13 カ月に短縮されます。

詳しくは、クライアント認証 (（ページ） 473)を参照してください。

Web レポートの再認証

Web レポートのセキュリティー強化のため、特定のページに変更を加える場合、現在の資格情報による再認証が必要となります。

詳しくは、再認証の実行 (（ページ）)を参照してください。

BigFix リレーのサポートを追加

Ubuntu 22.04 LTS x86 64 ビットで稼働する BigFix リレーのサポートが追加されました。

BigFix エージェントのサポートを追加

以下で稼働する BigFix エージェントのサポートが追加されました。

- Power 10 で稼働する AIX 7.2
- Power 9/10 で稼働する AIX 7.3
- Debian 11 x86 64 ビット
- MacOS 13 ARM/x86 64 ビット
- Power 10 で稼働する Red Hat Enterprise Linux 8
- Red Hat Enterprise Linux 9 x86 64 ビット
- Power 10 で稼働する SUSE Linux Enterprise 15
- Ubuntu 22.04 LTS x86 64 ビット

Active Directory 2016/2019 のサポートを追加

Active Directory 2016/2019 のサポートが追加され、フォレスト機能レベルの Windows Server 2016 と、Windows でのみ稼働する BigFix サーバーの エンタープライズ証明局が追加されました。

詳しくは、Windows サーバーと Active Directory との統合 (（ページ） 40)を参照してください。

ライブラリーのアップグレード

- libcurl ライブラリーは、バージョン 7.83.1 にアップグレードされました。

パッチ 6

BigFix エージェントのサポートを追加

Raspberry Pi 4 の Raspberry Pi OS 11 で稼働する BigFix エージェントのサポートが追加されました。

プラグイン・ポータルのパフォーマンス向上による RunAction 実行時間の短縮

プラグイン・ポータルがクラウドおよびモバイル・デバイス用の BigFix に完全対応するようになり、効率が大幅にアップしました。プラグインあたりのメモリー所要量が 89% 削減され、Run Actions 実行時間が 18% 改善されました。

ライブラリーのアップグレード

- OpenSSL ライブラリーがバージョン 1.0.2zd にアップグレードされました。
- zlib ライブラリーがバージョン 1.2.12 にアップグレードされました。
- jQuery ライブラリーがバージョン 3.6.0 にアップグレードされました。
- jQuery UI ライブラリーがバージョン 1.13.1 にアップグレードされました。

パッチ 5

プラグイン・ポータルのカスタム・インストール・パスの指定

Windows でプラグイン・ポータルをインストールする際にカスタム・インストール・パスを指定できるようになりました。

詳しくは、プラグイン・ポータル (（ページ） 155)を参照してください。

AWS プラグインでのスキャン済みリージョンの制限

AWS プラグインのインストール時に、許可されたリージョンを指定できるようになりました。

詳しくは『AWS リージョンの制限によるデバイス検出範囲の設定』を参照してください。

BigFix サーバーおよび BigFix コンソールのサポートの追加

Windows Server 2022 で稼働する BigFix サーバーおよび BigFix コンソールのサポートが追加されました。

BigFix リレーのサポートを追加

Tiny Core 12 で稼働する BigFix リレーのサポートが追加されました。

ライブラリーのアップグレード

- libcurl ライブラリーは、バージョン 7.79.1 にアップグレードされました。
- OpenSSL ライブラリーは、バージョン 1.0.2zb にアップグレードされました。

パッチ 4

AWS IAM ロールのサポート

AWS IAM ロールでクラウド・インスタンスの検出と管理ができるようになりました。IAM ユーザーまたは IAM ロールのアクセス権を活用できるため、AWS 資格情報の管理方法の幅が広がりました。

詳しくは、クラウド・プラグインのインストール (（ページ） 213)を参照してください。

相関エンドポイントをターゲットしたアクションの効率化

BigFix エージェントとプラグイン・ポータルによってエンドポイントで取得されたプロパティーに基づいたコンピューター・グループを作成できるようになりました。例えば、クラウド・インスタンスに関連付けられたプロパティーに基づいてクラウド・エンドポイントのグループを作成し、このグループを BigFix エージェントが実行するアクションのターゲットとして使用できます。

詳しくは、サーバー・ベースのコンピューター・グループの作成 (（ページ）)を参照してください。

特定のサブネット上の PeerNest UDP メッセージの制限によるネットワーク・トラフィック削減

PeerNest 機能を使用する場合、同じサブネットに接続されているエンドポイントによって交換される PeerNest UDP メッセージに関連付けられたネットワーク・トラフィックを削減できるようになりました。これは、VPN インフラストラクチャーで多数の BigFix クライアントを実行している場合に便利です。

詳しくは、PeerNest を使用した作業 (（ページ） 307)を参照してください。

ActionScript での MS-PowerShell の活用

BigFix アクション・スクリプト、UNIX シェル・スクリプト、AppleScript に追加して、MS-PowerShell for Action Scripts も使用できるようになりました。
詳しくは、以下のセクションを参照してください。

- 「編集」の「アクション」タブ (（ページ）)
- 「アクション・スクリプト」タブ (（ページ）)
- 「実行前アクション・スクリプト」タブ (（ページ）)
- 「実行後アクション・スクリプト」タブ (（ページ）)

CDT UI 改善による BigFix エージェントのデプロイメントの効率化

クライアント適用ツール (CDT) のユーザー・インターフェースが改善され、ユーザーは複数のクライアント設定と資格情報を使用して効率的にデータを入力できるようになりました。これにより、資格情報が異なる複数のターゲットが存在するケースや、複数のカスタム・クライアント設定を指定する必要があるケースで、BigFix エージェントのデプロイメントが高速化されます。

詳しくは、コンソールからのクライアントの適用 (（ページ）)を参照してください。

ライセンス情報の表示の改善

BigFixの「ライセンスの概要」ダッシュボードが改善され、BigFix デプロイメントに関連付けられたライセンス情報を確認しやすくなりました。さまざまな使用権のステータスや、エンドポイントがサブスクライブしている BigFix の機能についてもわかりやすく表示されるようになりました。

詳しくは、「ライセンスの概要」ダッシュボード (（ページ）)を参照してください。

プラグイン・ポータル・インスタンス 1 件あたりのエンドポイントが 5 倍

BigFix 10.0.4 では、プラグイン・ポータルの管理機能がインスタンスあたりエンドポイント 10,000 件から 50,000 件に増えました。これにより、多数のクラウドまたは MCM エンドポイントを管理する必要があるケースでの総所有コストが削減されます。

詳しくは、プラグイン・ポータル (（ページ） 155)を参照してください。

BigFix コンソールのサポートが追加されました。

Windows 11 21H2 で稼働する BigFix コンソールのサポートが追加されました。

BigFix リレーのサポートを追加

以下で稼働する BigFix エージェントのサポートが追加されました。

- Tiny Core 11
- Windows Server 2022
- Windows 11 21H2

BigFix エージェントのサポートを追加

以下で稼働する BigFix エージェントのサポートが追加されました。

- Windows Server 2022
- Windows 11 21H2
- Windows 11 22H2
- MacOS 12 ARM/x86 64 ビット

セキュリティーの脆弱性およびライブラリーのアップグレード

- libcurl ライブラリーは、バージョン 7.77.0 にアップグレードされました。
- OpenLDAP ライブラリーは、バージョン 2.4.58 にアップグレードされました。
- SQlite ライブラリーは、バージョン 3.35.5 にアップグレードされました。

パッチ 3

BigFix リレー/コンソール/エージェントのサポートの追加

Windows 10 バージョン 22H2 で稼働する BigFix リレー/コンソール/エージェントのサポートが追加されました。

BigFix リレー/コンソール/エージェントのサポートの追加

Windows 10 バージョン 21H2 で稼働する BigFix リレー/コンソール/エージェントのサポートが追加されました。

BigFix リレー/コンソール/エージェントのサポートの追加

Windows 10 バージョン 21H1 で稼働する BigFix リレー/コンソール/エージェントのサポートが追加されました。

BigFix エージェントのサポートを追加

MacOS 11 ARM64 で稼働する BigFix エージェントのサポートが追加されました。

セキュリティーの脆弱性およびライブラリーのアップグレード

- SQLite ライブラリーは、バージョン 3.34.1 にアップグレードされました。
- OpenLDAP ライブラリーは、バージョン 2.4.56 にアップグレードされました。
- OpenSSL ライブラリーは、バージョン 1.0.2y にアップグレードされました。

オペレーティング・システム・インスペクターにプロパティーを追加

`display version` という名前の新規プロパティーが `operating system` インスペクターに追加されました。このプロパティーは、Windows オペレーティング システムのバージョンと、Windows 10 20H2 以降のバージョンにのみ有効な情報を返します。

パッチ 2

クラウド API を使用して AWS または Azure VM に BigFix エージェントをインストールする

クラウド・プロバイダー・サービスと API を利用して、AWS および Azure 環境に BigFix エージェントをインストールできるようになりました。この機能拡張により、エージェントのデプロイメントを高速化できます。このとき、クライアント適用ツール (CDT) のデプロイと構成、およびターゲット・クラウド・インスタンスの OS アクセス資格情報の提供は必要ありません。

詳しくは、BigFix クラウド・リソースへのエージェントのインストール (（ページ） 285)を参照してください。

MS-SQL 構成のガイド付きチューニングによるパフォーマンスと回復力の向上

インストーラーにより、SQL Server インスタンスの DoP (並列処理の限度) と CTFP (並列処理のコストしきい値) の観点から、準最適な構成がチェックされ、オプションで調整されるようになりました。構成の問題を自動的に解決できない場合、十分なバックグラウンドとガイダンスが提供されます。

詳しくは、SQL Server の並列処理の最適化 (（ページ）
)を参照してください。

Windows でルート・サーバー DB の Docker イメージを利用する

Windows BigFix ルート・サーバーのリモート・データベースとして、Docker 用の MS SQL Server の公式 Ubuntu ベース・イメージを利用できるようになりました。プラットフォーム 10.0.2 では、MS SQL Server 2017 および MS SQL Server 2019 の Docker コンテナーを正式に認定しています。

詳しくは、Detailed system requirements (（ページ）)を参照してください。

ペイロードが大規模な場合の PeerNest 動作の改善

このリリース以降、ピアのキャッシュ・サイズに基づいてファイルをダウンロードするピアを選択することもできます。特定のクライアントのみ、リレーから大きいファイルを直接ダウンロードします。これにより、十分なキャッシュがないクライアントはダウンロードを開始できず、そのため、効率が向上し、ネットワーク帯域幅の使用率が低下します。

詳しくは、ピアツーピア・モード (（ページ）)を参照してください。

クライアントがダウンロード・フェーズで追加の CPU を使用できるようにすることで応答を高速化する

BigFix クライアントに追加の CPU を一時的に使用するように指示することで、ダウンロードしたファイルのハッシュ (sha1/sha256) コードを評価する操作を高速化できるようになりました。これにより、処理を行う CPU が増加するにつれて、ハッシュの評価に必要な時間が短縮されるため、ダウンロード・フェーズの時間を一貫して最適化できます。

詳しくは、設定のリストと詳細な説明 (（ページ）)を参照してください。

BigFix サーバーのサポートを追加

Red Hat Enterprise Linux (RHEL) 8 x86 64 ビットで実行される BigFix サーバーのサポートが追加されました。

BigFix リレーのサポートを追加

Raspberry Pi 4 の Raspbian 10 で実行される BigFix リレーのサポートが追加されました。

BigFix エージェントのサポートを追加

以下で稼働する BigFix エージェントのサポートが追加されました。

- Debian 10 x86 64 ビット。
- MacOS 11 x86 64 ビット。
- Power 8 の Ubuntu 20.04 LTS PPC 64 ビット LE。

新規データベース・レベルのサポートを追加

- DB2 バージョン 11.5.4/11.5.5/11.5.6/11.5.7 Stardard Edition をサポートします。

  注: DB2 11.5.0 を 11.5.4/11.5.5/11.5.6/11.5.7 にアップグレードする前に、BigFix をバージョン 10 パッチ 2 以降にアップグレードしてください。

- Microsoft SQL Server 2019 のサポート。
- docker コンテナーにデプロイされる Microsoft SQL Server 2017 および 2019。

新しい RPM パッケージが必要

パッチ 2 以降、Linux システムのサーバー・コンポーネントに対して unixODBC RPM が前提条件となります (『サーバー要件 (（ページ）　　)』を参照)。

アップグレードされたライブラリー

libcurl ファイルの転送ライブラリーのレベルがバージョン 7.73.0 にアップグレードされました。

パッチ 1

Google Cloud Platform からもクラウド・アセットを検出して報告する

この機能を使用すると、プラグイン・ポータルとプラグイン・テクノロジーを使用して、さまざまなクラウド・プロバイダーにわたってクラウド・アセットの可視性を検出および管理できます。検出されたクラウド・アセットに BigFix クライアントをインストールするには、WebUI または BigFix コンソールを使用します。

詳しくは、BigFix 管理機能の拡張 (（ページ） 194)を参照してください。

監査ログから詳細を取得する

監査ログ・サービスにより、BigFix サーバーへのログインとログアウトに関する詳細と、クライアントがサーバーにアクセスするために使用する IP アドレスに関する情報が提供されるようになりました。

詳しくは、サーバー監査ログ (（ページ） 448)を参照してください。

Forward Secrecy のサポートにより TLS 接続のセキュリティーを強化

デプロイメントのセキュリティーのレベルを向上させるために、鍵交換に一時的な Diffie-Hellman (DHE) と一時的な楕円曲線 Diffie-Hellman (ECDHE) を利用できるようになりました。

詳しくは、DHE/ECDHE 鍵交換方式の使用 (（ページ） 84)を参照してください。

VPN 経由で接続されたクライアントでネットワークへの影響と帯域幅の要件を軽減

サイトの構成可能リストに基づいて、インターネットからペイロードを直接取得するように BigFix クライアントを構成できる

ようになりました。これにより、VPN 経由で接続された BigFix クライアントにサービスを提供する BigFix リレーに関連するネットワークへの影響と帯域幅の要件を軽減できます。

詳細については、「設定のリストと詳細な説明 (（ページ）　　)」で説明されている `_BESClient_Download_DirectRecovery` という名前の構成設定を参照してください。

Web レポートの E メール・サーバーとして Microsoft Office 365 を使用する

以前のバージョンの BigFix プラットフォームでは、Web レポートは、SMTP 経由の基本認証を使用した場合にのみ E メール・サーバーに接続できました。このリリースでは、OAuth 2.0 と資格情報付与フローを備えた Office 365 E メール・サーバーを使用して、レポートの送信をスケジュールできます。

詳しくは、電子メールのセットアップ (（ページ）　　)を参照してください。

BigFix リレーのサポートを追加

Intel で Ubuntu 20.04 LTS で稼働する BigFix リレーのサポートが追加されました。

BigFix エージェントのサポートを追加

以下で稼働する BigFix エージェントのサポートが追加されました。

- Intel の Ubuntu 20.04 LTS。
- Windows 10 Enterprise for Virtual Desktops。

注: Windows 10 Enterprise for Virtual Desktops の場合、関連式「オペレーティング・システムの製

品情報ストリング」は「Server RDSH」を返します。この制限は、パッチ 1 に対してのみ有効です。

その他の機能拡張

- インストーラーが変更され、BigFix の評価版インストールのオプションから SQL Server 2016 SP1 評価版のセットアップが削除されました。

  詳しくは、Windows での評価版インストールの実行 (（ページ）)を参照してください。
- 追加情報とローテーションおよび最大サイズを設定する機能により、PeerNest と BigFix クライアントのデバッグ・ログのサービスが拡張されました。

  詳しくは、設定のリストと詳細な説明 (（ページ）)を参照してください。
- クライアント適用ツール (CDT) のウィザードが改善されました。クラウド・プラグインによって検出されるクライアントのインストール・プロセスが簡素化されました。

  詳しくは、検出されたリソースへの BigFix エージェントのインストール (（ページ） 284)を参照してください。
- 次の外部ライブラリーがアップグレードされました。
  - libcurl ファイルの転送ライブラリーのレベルがバージョン 7.69.1 にアップグレードされました。
  - Codejock ライブラリーは、バージョン 19.2.0 にアップグレードされました。
  - jQuery ライブラリーがバージョン 3.5.1 にアップグレードされました。

バージョン 10

マルチクラウド・サポート

BigFix 10 は、エンドポイントがクラウド内にあるかオンプレミスにあるかに関係なく、すべてのエンドポイントの単一の包括的なビューを提供します。この BigFix の拡張機能は、ネイティブのクラウド API を使用して複数のクラウド・プロバイダーにまたがる管理対象外のサーバーを同時に検出することで、Amazon Web Services、Microsoft Azure、VMware 環境で管理されていないクラウド・ブラインド・スポットを排除します。この機能により、クラウド・デバイスを完全に管理するために、BigFix エージェントを簡単にデプロイしてより深いレベルの可視性と制御を提供することもできます。

詳しくは、BigFix 管理機能の拡張 (（ページ） 194)と クラウド・プラグインの構成 (（ページ） 219)を参照してください。

認証としてリレーを適用するオプションによるセキュリティーの強化

BigFix 管理者は、デプロイメント時に認証としてリレーのインストールを選択できるようになりました。このオプションを使用することで、インターネットに接続されたリレーを保護および構成するベスト・プラクティスを合理化し、環境とデータを脅威から保護できます。

詳しくは、認証リレー (（ページ） 478)を参照してください。

REST API 呼び出し用の複数の Web レポート・サーバーのサポートの改善

お使いの環境に複数の BigFix Web レポート・サーバーがある場合、REST API に送信される特定の照会の優先順位を定義できます。この機能により、運用環境への潜在的な影響を回避しながら、統合を制御する方法がより柔軟になります。

詳しくは、https://developer.bigfix.com/rest-api/api/webreports.htmlを参照してください。

BigFix エージェントのロギングの強化

BigFix エージェント・ログには、追加のエンドポイント識別情報 (OS、ホスト名、IP アドレスを含む) とリレー選択データが含まれるようになったため、保守性の向上とトラブルシューティングの簡素化に役立ちます。

その他の機能拡張

- 「アクションの実行」ダイアログが改善され、デフォルトで「すべてのコンピューター」を対象としないようになりました。
- 予約済みプロパティーとして MAC アドレスが導入されました。
- 次のサポートが追加されました。
  - BigFix サーバー (Windows Server 2019 上)。
  - BigFix リレー (AMD/Intel の SUSE Linux Enterprise Server (SLES) バージョン 15 上)。
  - BigFix リレー (Intel の Red Hat Enterprise Linux バージョン 8 x86 64 ビット上)。
  - BigFix リレー/エージェント (Amazon Linux 2 上)。

    注: Amazon Linux 2 の場合、リレー/クライアント・パッケージは Red Hat Enterprise Linux 6 パッケージです。

  - BigFix エージェント (Intel の Oracle Enterprise Linux 8 上)。
  - BigFix エージェント (Power 8 および 9 の Red Hat Enterprise Linux 8 PPC 64 ビット LE 上)。
  - BigFix エージェント (s390x の SUSE Linux Enterprise Server (SLES) バージョン 15 上)。
- OpenSSL ツールキットのレベルがバージョン 1.0.2u にアップグレードされました。

OS とデータベースのサポートの変更

BigFix 10 では、さまざまな BigFix コンポーネントでサポートされる最小バージョンのオペレーティング・システムとデータベースにいくつかの変更が加えられています。これらの変更の中で注目すべき点は、BigFix 10 サーバーに以下のいずれかが必要になったことです。

- Windows Server 2012 R2 以降 + SQL Server 2012 以降
- または Red Hat Enterprise Linux バージョン 7 + DB2 バージョン 11.5 GA。

詳しくは、Detailed system requirements (（ページ）)を参照してください。

本書で使用される用語

BigFix の用語は、常に BigFix というラベルが付いているとは限りません。

次の用語は、すべて BigFix の用語です。これらの用語は、本書全体で、BigFix というラベルなしで随時使用されます。

エージェント

BigFix クライアントがインストールされているコンピューター

コンソール

BigFix コンソール

クライアント

BigFix クライアント

サーバー

BigFix サーバー

リレー

BigFix リレー

第 2 章. BigFix サイト管理者およびコンソール・オペレーター

BigFix には、次の 2 つの基本的なユーザー・クラスがあります。

サイト管理者

サイト管理者は、BigFix ソフトウェアのインストールと保守、および環境全体に影響を与える管理用タスク (サイト・レベルの署名鍵管理など) の実行を担当します。BigFix 環境のサイト管理者は 1 人のみです。詳しくは、『サイト管理者 (*(ページ) 21)*』を参照してください。

コンソール・オペレーター

BigFix のユーザーであり、BigFix コンソールにアクセスして、許可されている場合には WebUI にアクセスします。BigFix コンソールの管理者ユーザーである**マスター・オペレーター (MO)** や、自身のドメインの日常的な管理を行う**オペレーター (NMO)** の場合もあります。マスター・オペレーターは他のオペレーターを作成して管理権限を割り当てることができますが、オペレーターはいずれも実行できません。詳しくは、『オペレーターの概要 ((ページ))』を参照してください。

注: オペレーターを定義する際に、ユーザー名に無効な文字(`:`、`@`、および `\`) が含まれていないことを確認してください。

サイト管理者

サイト管理者の主な責務を以下に示します。

アクション・サイト認証情報の取得とセキュリティー保護

BigFix をインストールするために、サイト管理者は、秘密鍵の生成、HCL からのライセンス証明書の受け取り、デジタル署名と構成情報を持つマストヘッドの作成を行う必要があります。これは特殊なキーであるため、以下に示すようなサイト・レベルのタスクでのみ使用してください。

- グローバル・システム・オプションの設定
- マストヘッドの編集
- 分散サーバー・アーキテクチャー (DSA) の管理

サーバーの準備

BigFix サーバーがインターネットと外部通信するように、およびクライアントと内部通信するように正しくセットアップする必要があります。またサーバーが BigFix データベース (または SQL Server データベースとして使用できる別のコンピューター) をホストするように構成する必要もあります。

各種コンポーネントのインストール

サイト管理者は、BigFix のクライアント・モジュール、サーバー・モジュール、リレー・モジュール、コンソール・モジュールをインストールして、最初のマスター・オペレーターの資格情報を構成します。マスター・オペレーターは、コンソールに接続して、ライセンスのサブスクリプションを定義し、サブスクライブしたサイトからコンテンツを収集して、BigFix ネットワーク、役割、および他のオペレーターを定義します。

サイト管理者は、災害対応サーバー・アーキテクチャー (DSA) (（ページ）) で、BigFix サーバーの自動フェイルオーバーおよび自動フェイルバックを実行するための複数の BigFix サーバーのセットアップと管理を行います。

サーバーの保守

BigFix サーバーは、SQL Server データベースといくつかの特定のサービス (診断ツールや管理ツールの実行など) を実行します。Fixlet テクノロジーを使用して、アップグレードやフィックスなどの標準的なメンテナンス・タスクが管理されます。これらのタスクは、サイト管理者が手動で実行することもできます。

日常的なコンソール操作を実行するために、サイト管理者はマスター・オペレーター・キーを作成する必要があります。

サイト管理者は以下を実行できません。

- BigFix コンソールにアクセスします。
- インストール時に作成されたオペレーター以外のオペレーターの作成。
- BigFix WebUI にアクセスします。
- BigFix 照会の実行。

コンソール・オペレーター

コンソール・オペレーターには以下の 2 つのタイプがあります。

マスター・オペレーター (MO)

コンソールの管理ユーザーです。マスター・オペレーターは、BigFix 環境で定義されているすべてのコンピューターへのアクセス権と、他のコンソール・オペレーターの作成権限と管理権限を持っています。すべてのマスター・オペレーターが、オペレーターに対してアクションの適用を許可する管理権限の作成、割り当て、取り消しを行うことができます。

オペレーターまたはマスター以外のオペレーター (NMO)

マスター・オペレーターによって管理を許可されたコンピューターのサブセットに対する Fixlet の管理やアクションの適用などを含む、BigFix の日常的な操作を管理します。他のオペレーターを作成することも、管理権限を割り当てることもできません。

デフォルトでは、コンソール・オペレーターは以下を実行できません。

- WebUI へのアクセス（**「WebUI を使用できます」** 権限が**「はい」**に設定されている場合を除く)。
- BigFix 照会の実行依頼 (**「WebUI を使用できます」**と**「照会を送信可能」**の両方の権限が**「はい」**に設定されている場合を除く)。

このような権限は、マスター・オペレーターがオペレーターの説明の「詳細」タブの「権限」領域で設定できます。オペレーターの権限について詳しくは、許可されるアクティビティーと許可とのマッピング (（ページ） *32)*を参照してください。

ベスト・プラクティス

以下の表は、マスター・オペレーター (MO) またはマスター以外のオペレーター (NMO) の役割をいつ使用するかを示しています。

表 1. マスター・オペレーター

MO
BigFix コンソールの管理ユーザーです。マスター・オペレーターは、BigFix 環境で定義されているすべてのコンピューターへのアクセス権と、他のコンソール・オペレーターの作成権限と管理権限を持っています。すべてのマスター・オペレーターが、オペレーターに対してアクションの適用を許可する管理権限の作成、割り当て、取り消しを行うことができます。
ユーザー/オペレーター/役割を作成します。
カスタム・サイトを作成します。
すべてのオペレーターに表示され、ほとんどのコンピューターで使用される可能性がある、カスタム・コンテンツを作成します。
BigFix 環境全体に関連する特定のポリシー・アクションを発行します。マスター・アクション・サイト・サイズに追加されるため、アクションの数を最小限に抑えます。
サイトのサブスクリプションを管理します。
取得したプロパティーを作成します。
コンテンツをグローバルに非表示にします。
すべてのマスター・オペレーターとマスター以外のオペレーターに対してグローバル分析をアクティブ化します。

表 2. マスター以外のオペレーター

NMO
マスター・オペレーターによって管理を許可されたコンピューターのサブセットに対する Fixlet の管理やアクションの適用などを含む、BigFix の日常的な操作を管理し

表 2. マスター以外のオペレーター (続く)

NMO
ます。他のオペレーターを作成することも、管理権限を割り当てることもできません。
パッチのデプロイなどのアクションを実行します。
REST API 呼び出しを行います。
Fixlet、タスク、ベースライン、分析をデプロイ/アクティブ化/非アクティブ化します。
特定の目的のためにカスタム・コンテンツを作成します。
マスター以外のオペレーターの管理対象コンピューターに基づいて、ローカル分析をアクティブ化します。

コンソール・オペレーターを追加するための各種の方法

コンソール・オペレーターを追加して、役割を割り当てる場合、または特定のコンピューターおよびサイトを表示または管理するための権限を付与する場合、いくつかの方法があります。

- オペレーターを 1 人追加する作業は、**「ツール」>「オペレーターの作成」**項目を選択するか、オペレーターの作業域を右クリックしてから**「オペレーターの作成」**を選択することでいつでも行うことができます。詳しくは、ローカル・オペレーターの追加 (（ページ） *26)*を参照してください。
- Active Directory または汎用 LDAP を使用している場合、**「ツール」>「LDAP オペレーターの追加」**項目を選択するか、オペレーターの作業域を右クリックしてから**「LDAP オペレーターの追加」**を選択することで、以前に定義されたユーザーを追加できます。詳しくは、LDAP オペレーターの追加 (（ページ） *51)*を参照してください。

- また、LDAP グループを既存の役割に関連付けることもできます。このようにして、1 回のクリックで、LDAP グループで指定された各ユーザーのオペレーターを追加したり、そのオペレーターを役割に関連付けたりします。この機能について詳しくは、LDAP グループの関連付け (（ページ） 53)を参照してください。

注: LDAP オペレーターと LDAP グループを使用する場合は、まず Active Directory ディレクトリーまたは LDAP ディレクトリーを BigFix に追加する必要があります。

ローカル・オペレーターの追加

ローカルの BigFix アカウントを使用してコンソールにアクセスするオペレーター用に、アカウントを作成することができます。

ローカル・オペレーターを追加するには、以下の手順を実行します。

1. **「ツール」>「オペレーターの作成」**メニュー項目をクリックするか、オペレーター作業域で右クリックして、**「オペレーターの作成」**を選択します。**「ユーザーの追加」**ダイアログが表示されます。
2. 発行者またはオペレーターとして指定するユーザーの**「ユーザー名」**を入力します。
3. **「パスワード」**を作成し、確認のために再入力します。オペレーターにキーを知らせた場合、オペレーターは必要な場合にパスワードを変更できます。
4. **「OK」**をクリックします。**「コンソール・オペレーター」**ウィンドウが開きます。
5. **「詳細」**タブから、オペレーター権限を割り当てます。

BigFix 管理ツールの**「詳細オプション」**の**「defaultOperatorRolePermissions」**オプションを使用して、デフォルト設定を制御できます。詳しくは、詳細オプションのリスト (（ページ） 452)を参照してください。

マスター・オペレーター

当該オペレーターがマスター・オペレーターであるかどうかを指定します。

他のオペレーターのアクションの表示

当該オペレーターが他のオペレーターによって実行依頼されたアクションを表示できるかどうかを指定します。

注: 「他のオペレーターのアクションの表示」権限を持つオペレーターは、次の場合にのみアクションを表示できます。

- アクションの所有者である場合。
- 当該オペレーターの 1 つ以上の管理対象コンピューターで他のオペレーターがアクションを実行依頼し、そのコンピューターが両方のオペレーターによって管理されている場合。このケースでは、コンピューターがデータを BigFix サーバーにレポートする場合にのみ、情報を使用できます。

他のオペレーターのアクションの停止

マスター以外のオペレーター (NMO) が、他のマスター以外のオペレーターによって実行依頼されたアクションを停止できるかどうかを指定できます。詳しくは、『「他のオペレーターのアクションの停止」機能 (（ページ） 30)』を参照してください。

アクションの作成が可能

当該オペレーターがアクションを作成できるかどうかを指定します。

注: 非マスター・オペレーターがデータベースからコンピューターを削除するには、**「アクションの作成が可能」**権限が必要です。

ロック可能

当該オペレーターがターゲットをロックできるかどうかを指定します。これは、他のオペレーターがそのターゲットでアクティビティーを実行できないようにする 1 つの方法です。

複数クライアントへの更新の送信が可能 (Can Send Refresh to Multiple Clients)

当該オペレーターが BigFix コンソール上の**「更新」**ボタンをクリックすることにより複数のターゲットに対して同時に更新を実行できるかどうかを指定します。

照会を実行依頼できます (Can Submit Queries)

当該オペレーターが WebUI ユーザー・インターフェースから BigFix 照会要求を実行依頼できるかどうかを指定します。

カスタム・コンテンツ

カスタム・コンテンツの作成が必要なアクティビティーを当該オペレーターが実行できるかどうかを指定します。

注: 「カスタム・コンテンツ」および**「アクションの作成が可能」**権限を持つ非マスター・オペレーターは、既存のコンピューター設定の編集/削除のみ可能で、新しいコンピューター設定を追加することはできません。

非管理資産

BigFix コンポーネントが 1 つもインストールされていない資産を当該オペレーターが管理できるかどうかを指定します。

「明示的な権限」は、オペレーターに割り当てることになる権限です。**「有効な権限」**は、オペレーターが割り当てられた役割から継承された権限です。同じ権限に対して**「明示的な権限」**と**「有効な権限」**に表示される値が異なる場合は、より制限が少ない権限が適用されます。

再起動やシャットダウンをポスト・アクションとしてトリガーする、または BigFix アクション・スクリプトに含めるオペレーターの権限も設定します。

特定のオペレーターに対して設定したシャットダウンおよび再起動の構成によっては、「アクションの実行」パネルのラジオ・ボタンがそのオペレーターに対して無効になる場合があります。この構成は、タイプが BigFix アクション・スクリプトではないアクションに対しては無効です。
BigFixユーザー・インターフェースにアクセスするための権限を設定することもできます。

6. **「管理対象コンピューター」**タブに、このオペレーターが管理できるコンピューターのリストが表示されます。このリストにデータが取り込まれるのは、**「コンピューターの割り当て」**タブに指定された条件を満たすコンピューターが、その情報を BigFix サーバーに報告した後です。
7. **「割り当てられた役割」**タブで、このオペレーターに適用する役割を選択します。
8. **「サイト」**タブで、このオペレーターにアクセスを許可するサイトを割り当てます。
9. **「コンピューターの割り当て」**タブで、このオペレーターが扱えるコンピューターが合致する必要のあるプロパティーを指定します。マスター・オペレーターの場合、すべてのコンピューターが割り当てられます。
10. **「WebUI アプリケーション」**タブで、オペレーターにアクセスを許可する WebUI アプリケーションを指定します。
11. 変更内容を保存するには、「**変更の保存**」をクリックします。

また、ローカル・オペレーターを、いつでも LDAP オペレーターに変換できます。これを行うには、以下のステップに従ってください。

1. 任意のローカル・オペレーター・リストで、変換したいオペレーターを右クリックします。
2. コンテキスト・メニューで、**「LDAP オペレーターに変換」**を選択します。

「他のオペレーターのアクションの停止」機能

特定の条件が満たされる場合は、マスター以外のオペレーター (NMO) によって送信されたアクションを、マスター以外の他のオペレーターが停止できるようになりました。

マスター以外のオペレーター (NMO) で、アクションを開始する人 (発行者) の要件

この NMO には、少なくともアクションと、割り当てられた役割から継承されたか明示的に割り当てられた一部のコンピューターを作成して実行依頼できる必要がありますが、この機能に関連するその他の特定の制限や要件はありません。

マスター以外のオペレーター (NMO) で、アクションを停止する人 (停止者) の要件

1. この NMO には、**「他のオペレーターのアクションの表示」**と**「他のオペレーターのアクションの停止」**の両方の有効な権限が**「はい」**に設定されている必要があります。

2. この NMO には、割り当てられた役割名のセット (発行者の役割名と同一またはスーパーセット) が必要です。比較は、これらの割り当てられた役割の名前のみに基づいていることに注意してください。発行者は役割を割り当てなくてもかまいません。この場合、停止者にも役割を割り当てる必要はありません。
3. この NMO には、発行者のスーパーセットまたは同一の明示的な「コンピューターの割り当て」定義のセットが必要です。明示的な「コンピューターの割り当て」定義は、オペレーターに割り当てられているターゲットのリストではなく、それらのコンピューター割り当ての定義であることに注意してください。NMO には、複数の明示的な割り当てを同時に関連付けることができます。**「すべてのコンピューター」**の明示的な割り当ては、他のコンピューター割り当て定義のスーパーセットではありません。役割に関しては、発行者は明示的に割り当てられたコンピューターを持つことはできません。この場合、停止者にも明示的な割り当てを行う必要はありません。

その他の考慮事項

さらに、割り当てられた役割 (NMO の**「割り当てられた役割」**タブで指定) から継承された割り当てと、明示的に割り当てられた割り当て (NMO の**「コンピューターの割り当て」**タブで指定) が個別に評価されることを考慮してください。

ここからは、**「割り当てられた役割」** と**「コンピューターの割り当て」**の例をいくつか示します。

次のスクリーンショットは、NMO に割り当てられた複数の役割 (myrole1 と myrole2) を示しています。

次のスクリーンショットは、NMO に割り当てられた複数のコンピューター割り当て定義を示しています。

- 「コンピューター・タイプ」プロパティー ->「仮想」
- 「BES リレー選択方法」プロパティー ->「マニュアル」
- 「グループ」->「mygroup1」

許可されるアクティビティーと許可とのマッピング

次の表に、オペレーター定義の「詳細」タブで許可を割り当てることにより、オペレーターに許可できるアクティビティー、許可できないアクティビティー、および特定の条件下で許可できるアクティビティーを示します。

オペレーターの特定の許可について詳しくは、「ローカル・オペレーターの追加 (（ページ） 26)」を参照してください。

表 3. 許可されるアクティビティーとオペレーター許可とのマッピング

アクティビティー	オペレーター
Fixlet サイトの管理	いいえ
クライアントのハートビートの変更	いいえ
Fixlet の作成	**「カスタム・コンテンツ」**が「はい」に設定されている場合
タスクの作成	**「カスタム・コンテンツ」**が「はい」に設定されている場合
分析の作成	**「カスタム・コンテンツ」**が「はい」に設定されている場合
ベースラインの作成	**「カスタム・コンテンツ」**が「はい」に設定されている場合
分析のアクティブ化/非アクティブ化	管理対象
Fixlet/タスク/ベースライン・アクションの実行	管理対象
カスタム・アクションの実行	**「カスタム・コンテンツ」**が「はい」、および**「アクションの作成が可能」**が「はい」に設定されている場合
アクションの停止	管理対象
管理権限の管理	いいえ
グローバル取得プロパティーの管理	いいえ
Fixlet の表示	管理対象
タスクの表示	管理対象
分析の表示	管理対象

表 3. 許可されるアクティビティーとオペレーター許可とのマッピング (続く)

アクティビティー	オペレーター
コンピューターの表示	管理対象
ベースラインの表示	管理対象
コンピューター・グループの表示	管理対象
非管理資産の表示	管理対象
アクションの表示	管理対象
コメントの作成	管理対象
コメントの表示	管理対象
全体で非表示/再表示	いいえ
ローカルで非表示/再表示	はい
ウィザードの使用	**「カスタム・コンテンツ」**が「はい」に設定されている場合
データベースからコンピューターを削除	**「アクションの作成が可能」**が「はい」に設定されている場合
手動コンピューター・グループの作成	**「アクションの作成が可能」**が「はい」に設定されている場合
手動コンピューター・グループの削除	**「カスタム・コンテンツ」**が「はい」に設定されている場合
自動コンピューター・グループの作成	**「カスタム・コンテンツ」**が「はい」に設定されている場合
自動コンピューター・グループの削除	**「カスタム・コンテンツ」**が「はい」に設定されておりかつ管理対象の場合

表 3. 許可されるアクティビティーとオペレーター許可とのマッピング (続く)

アクティビティー	オペレーター
「カスタム・サイトの作成」	いいえ
カスタム・サイト所有者の変更	いいえ
カスタム・サイト閲覧者/作成者の変更	サイト所有者
マスター・オペレーターの作成	いいえ
WebUI の使用	**「WebUI を使用できます」**が「はい」に設定されている場合
BigFix 照会の実行依頼	**「WebUI を使用できます」**と**「照会を実行依頼できます 」**の両方が**「はい」**に設定されている場合

管理対象: オペレーターはアクセス許可を所有している必要があります。

カスタムの作成が必要: コンソール経由でサイト管理者によって付与されます。

オペレーターと分析

オペレーターが分析をアクティブにするとき、および非アクティブにするとき、各種の権限および制限が適用されます。

- 通常のオペレーターは、他のオペレーターが管理するコンピューターで、それらの他のオペレーターがアクティブにした分析を非アクティブにすることはできません。
- マスター・オペレーターは、通常のオペレーターが作成したカスタム分析を直接アクティブにすることはできません。ただし、マスター・オペレーターは、分析のコピーを作成し、そのコピーをアクティブにすることはできます。

オペレーターのモニター

ユーザーがマスター・オペレーターの場合 (BigFix 管理ツールで作成され、適正に認証されたユーザー名を持っている必要があります)、他のオペレーターがどのような処理を実行しているか、およびどのコンピューターの管理権限を持っているのかをモニターできます。

属性のうち各オペレーターを表すのは、**「名前」**、**「ユーザーの種類」**、および**「ログインの種類」**です。コンソール・オペレーターのリストを表示するには、**「すべてのコンテンツ」**ドメインを選択し、次にドメイン・パネルから**「オペレーター」**のラベルが付いたノードをクリックします。右のリスト・パネルに、すべての現行オペレーターがリストされます。

リスト・パネルでオペレーターをクリックすると、**「オペレーター」**作業域が開きます。

以下の複数のタブから選択できます。

- **詳細:** 名前と種類別にオペレーターが記述されており、ログインの種類を選択できます。ここで、オペレーターの権限を表示および変更することもできます。
- **「管理対象コンピューター」:** 選択されたコンソール・オペレーターに現在割り当てられているコンピューターのリストが表示されます。
- **「発行されたアクション」:** 選択されたコンソール・オペレーターが発行したアクションのリストが表示されます。
- **「割り当てられた役割」:** 現在割り当てられている役割が表示され、役割の再割り当てができます。
- **「サイト」:** このオペレーターに現在割り当てられているサイトが表示され、サイトの再割り当てができます。そのサイトがカスタム・サイトの場合、読み取り/書き込み/所有者権限を設定することもできます。
- **「コンピューターの割り当て」:** このオペレーターが管理できるコンピューターが一致する必要があるプロパティーをリストします。一致すべきプロパティーを指定すると、そのプロパティーに一致するようにコンピューターが変更された場合はいつでも、オペレーターに割り当てられているコンピューターのリストにそのコンピュー

ターが追加されます。また、コンピューターが変更されてそのプロパティーに一致しなくなると、そのコンピューターはリストから削除されます。

このタブを使用できるのは、マスターではないオペレーターのみです。

第 3 章. LDAP との統合

Lightweight Directory Access Protocol (LDAP) の関連付けを BigFix に追加することができます。

これにより、自分と他のユーザーがその資格情報を使用してコンソールにログインできるようになります。この利点は Web レポートにも当てはまります。

BigFix を汎用 LDAP または Active Directory と統合する方法については、以降のトピックの説明を参照してください。

注: SSL を使用して BigFix を汎用 LDAP サーバーまたは Active Directory サーバーに統合する場合、BigFix はロード・バランサーまたは DNS 別名を介した LDAP サーバーまたは Active Directory サーバーへの SSL 接続をサポートしていないことを考慮してください。

これらの 2 つのタイプの LDAP のいずれかと統合するステップを完了した後、LDAP のユーザーまたはグループを BigFix コンソール・オペレーターまたは役割に関連付けることができます。詳しくは、LDAP オペレーターの追加 (（ページ） 51)およびLDAP グループの関連付け (（ページ） 53)を参照してください。

汎用 LDAP との統合

既存の LDAP ドメインをコンソールに追加して、汎用 LDAP との統合を構成する方法。

以下の手順を実行します。

1. **「ツール」**メニューから**「LDAP ディレクトリーの追加」**を選択するか、作業域で右クリックして、**「LDAP ディレクトリーの追加」**を選択します。**「LDAP ディレクトリーの追加」** ダイアログが表示されます。

2. 名前を入力し、「種類」プルダウンから**「汎用 LDAP サーバー」**が選択されていることを確認します。汎用 LDAP サーバーの場合、**グローバル・カタログ**・オプションは使用できません。

3. LDAP インストール済み環境に関連する情報を入力します。**「サーバー」** に、サーバーのホスト名または IP アドレスを入力します。
4. ポート番号を入力します。Secure Sockets Layer (SSL) を使用している場合は、通常 636 です。
5. 基本識別名 (**「ベース DN」**) を、`dc=example,dc=com` の形式で入力します。
6. ボタンをクリックして、**匿名で接続**するか、または**資格情報を使用して接続**します。資格情報を使用して接続することを選択した場合は、**「ユーザー DN」** と **「パスワード」** を入力します。
7. **「テスト」** をクリックして、入力した情報が正しいこと、および LDAP への接続を確立できることを確認します。
8. ユーザー・フィルターまたはグループ・フィルターを組み込む場合は、**「詳細設定を表示」**リンクをクリックします。フィルターを指定すると、それ以降のすべての LDAP 検索に適切なフィルターが適用されるようになります。
9. **「追加」**をクリックして、LDAP の設定を完了します。

これで LDAP サーバーの構成が完了し、コンソールで使用できるようになります。

Active Directory との統合

Microsoft Active Directory (AD) を使用して、BigFix での認証を処理することができます。

これにより、自分と他のユーザーが Active Directory 資格情報を使用してコンソールにログインできるようになり、既存の認証ポリシーを利用できます。この利点は Web レポートにも当てはまります。

BigFix プラットフォーム・バージョン 9.5 パッチ 14 以降では、LDAP チャネル・バインディングおよび LDAP 署名で構成されている Active Directory との統合がサポートされています。

注: Windows プラットフォームでは、Active Directory の呼び出しを管理するインスペクターにより、BESClient プロセスに必要な 52311 ポートに加えて、一時的なポートがユーザー・データグラム・プロトコル (UDP) に割り当てられます。このポートは、netstat -an コマンドの出力で確認できます。

Windows サーバーと Active Directory との統合

既存の Active Directory をコンソールに追加する方法。

次のステップに従ってください。

1. **「ツール」**メニューで**「LDAP ディレクトリーの追加」**を選択します。**「LDAP ディレクトリーの追加」** ダイアログが表示されます。

2. Active Directory の名前を入力し、「種類」プルダウンから **「Microsoft Active Directory」** が選択されていることを確認します。
3. **「サーバー」** に、サーバーのホスト名、IP アドレス、または完全修飾ドメイン名を入力します。
4. セキュア接続 (SSL) を構成する場合は、**「SSL の使用」**をクリックします。
5. Active Directory フォレスト全体にアクセスする場合は、**「これはグローバル・カタログ・サーバーです」** をクリックします。
6. ボタンをクリックして、**ルート・サーバーのサービス・ユーザーとして接続**するか、または**資格情報を使用**して接続します。資格情報を使用して接続することを選択した場合は、Active Directory の **「ユーザー名」** と **「パスワード」** を入力します。
7. **「テスト」** をクリックして、入力した情報が正しいこと、および Active Directory サーバーへの接続を確立できることを確認します。
8. **「追加」**をクリックして、Active Directory の設定を完了します。

注: LDAP サーバーを**「Microsoft Active Directory」**として追加する場合は、LDAP サーバー上で、各ユーザーの**「ユーザーのログイン名」**に対応する `UserPrincipalName` 属性を定義していることを確認してください。この属性値は、各ユーザー認証用に BigFix コンソールで使用されます。

これで Active Directory サーバーの設定が完了し、コンソールで使用できるようになります。

Windows でのみ実行される BigFix サーバーの AD ドメイン/フォレスト機能レベル

BigFix 10.0.7 は、以下の構成で SSL を使用する AD ドメイン/フォレスト機能環境で完全にサポートされています。

- BES サーバーは Windows でのみ実行する必要があります。
- Active Directory Windows 2016 または 2019 は、2016 ドメイン機能レベルで定義され、パッチ・レベルが更新されています。
- グローバル・カタログがインストールされているすべての Active Directory。
- ルート・ドメインにインストールされているエンタープライズ認証機関。
- 以下の特性を持つ CA の証明書を作成します。
  - さまざまなドメイン拡張子を持つ AD フォレスト (例: ルート・ドメインの BIGFIX.ACME.COM と子ドメインの CHILD.BIGFIX.ACME.COM) では、ドメイン名の共通部分を考慮し、「*」で始まる共通名を持つ証明書を作成します (以下に例を示します)。

    ```
    CN = *.BIGFIX.ACME.COM
    ```

  - 次に、同じ証明書で DNS を定義し、すべての AD サーバーをリストします。これにより、作成された証明書の「詳細」タブには、値を含む「所有者代替名」フィールドが表示されます (以下に例を示します)。

    ```
    DNS Name=MyRootAD.BIGFIX.ACME.COM
    ```

    ```
    DNS Name=MyChildAD.CHILD.BIGFIX.ACME.COM
    ```

証明書は、SAN リストのすべての Active Directory にロードする必要があります。

このシナリオは、Active Directory にインストールされた DNS で認証されました。

Linux サーバーと Active Directory との統合

Kerberos 認証の構成

Linux BigFix サーバーと Active Directory の間でセキュア通信を確保するには、Kerberos プロトコルを使用します。

Kerberos 付き LDAP 認証を使用して Linux BigFix サーバーを Windows Active Directory ドメインと統合するには、以下の手順を実行します。

1. Linux BigFix サーバーと Active Directory サーバーの両方でホスト名とタイム・サービスが正しく設定されていることの確認
2. NSS ライブラリーおよび PAM ライブラリーのインストール
3. Kerberos LDAP セキュリティーおよび認証の構成
4. ローカル LDAP ネームの変更
5. NSS ライブラリーおよび PAM ライブラリーの構成

事前チェック項目

Red Hat Enterprise Linux 7 または Linux 8 システム上で稼働する BigFix サーバーと Active Directory サーバーの統合を実行する前に、以下のことを確認します。

- Red Hat Enterprise Linux 7 または Linux 8 システムと Active Directory サーバーの両方の DNS ホスト名が正しく解決されるようにします。そのためには、Red Hat Enterprise Linux 7 または Linux 8 システムで以下の手順を実行します。
    1. ファイル `/etc/host` を開き、両方の DNS ホスト名が完全修飾ドメイン名として指定されていることを確認します。
    2. ファイル `/etc/sysconfig/network` を開き、Red Hat Enterprise Linux 7 または Linux 8 システムのホスト名が完全修飾ドメイン名として指定されていることを確認します。
- Active Directory と Linux BigFix サーバーの間で 時刻を同期します。必要な場合は、Red Hat Enterprise Linux 7 または Linux 8 システムのタイム・サービスと Active Directory サーバーをタイム・ソース・サーバーと同期できます。そのためには、以下の手順を実行します。

1. Red Hat Enterprise Linux 7 または Linux 8 システム上のファイル `/etc/ntp.conf` で、以下の行を置換します。

   ```
   server hostname
   ```

   置換後の行

   ```
   server time_source_server_name
   ```

   time_source_server_name には、時刻の同期に使用されるタイム・ソース・サーバーのサーバー・ホスト名または IP アドレスが入ります。
2. DNS ルックアップが信頼できない場合は、`/etc/resolv.conf` ファイルを以下のように編集して、Active Directory サーバーから DNS ルックアップを実行するように Red Hat Enterprise Linux システムを構成します。

   ```
   domain my.domain.com
   search my.domain.com
   nameserver1 ipaddress1
   nameserver2 ipaddress2
   ```
3. 以下の手順に従って、Red Hat Enterprise Linux 7 システム上で変更を有効にします。
   - **ntp** デーモンの停止:

     ```
     service ntpd stop
     ```
   - 時刻の更新:

     ```
     ntpdate Red_Hat_server_IP
     ```
   - **ntp** デーモンの開始:

     ```
     service ntpd start
     ```
4. 以下を入力して、Active Directory サーバーをタイム・ソース・サーバーと同期させます。

   ```
   w32tm /config /manualpeerlist:"time_source_server_name"
         /syncfromflags:manual /update
   ```

time_source_server_name には、Linux サーバーが同期されている NTP タイム・ソースの DNS 名または IP アドレスのリストを指定します。例えば、`time.windows.com` を NTP タイム・サーバーとして指定できます。複数のピアを指定する場合は、スペースを区切り文字として使用し、各ピアの名前を引用符で囲みます。

5. Active Directory サーバー上で、以下のコマンドを実行して時刻がタイム・ソース・サーバーと同期されていることを確認します。

```
w32tm /query /status | find "Source"
w32tm /query /status | find "source"
```

6. Red Hat Enterprise Linux 7 システム上で、**ntpd** デーモンをシステム・ブート時に開始されるように構成します。

```
chkconfig ntpd on
```

NSS ライブラリーおよび PAM ライブラリーのインストール

以下の NSS および PAM パッケージがインストールされていることを確認します。

```
nss-pam-ldapd-0.7.5-18.2.el6_4.x86_64.rpm
pam_krb5-2.3.11-9.el6.x86_64.rpm
```

注: 有効な RHN サブスクリプションがある場合は、以下の例に示すように yum を実行します。

```
yum install nss-pam-ldapd.x86_64 pam_krb5.x86_64
```

認証の構成

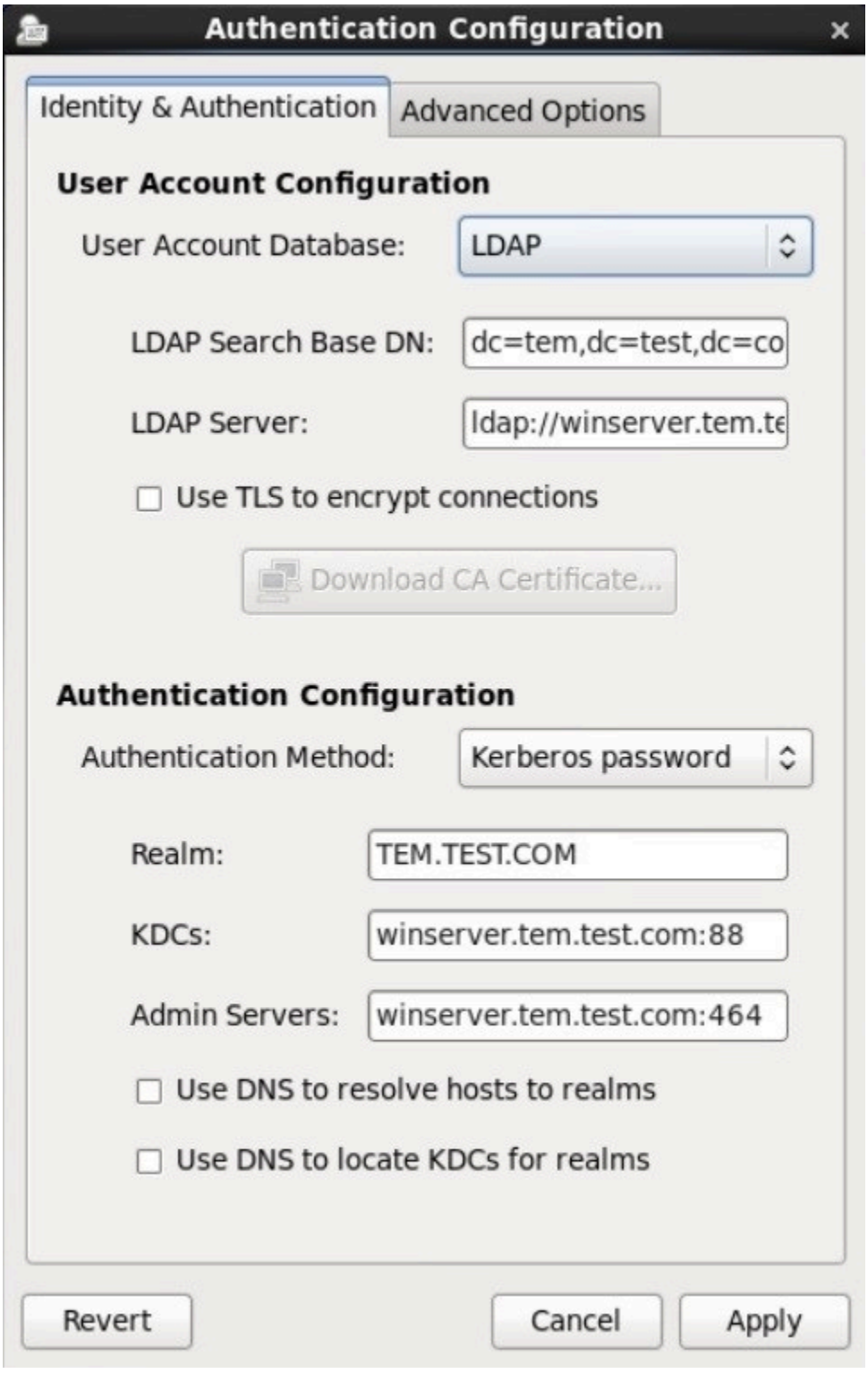
Active Directory の統合のために Kerberos プロトコル、LDAP セキュリティー、および認証ファイルを構成するには、以下のいずれかの方法を使用できます。

- **system-config-authentication** グラフィック・ツール。
- **authconfig** コマンド行ツール。

system-config-authentication グラフィック・ツールの使用

system-config-authentication ツールを使用して認証を構成するには、以下の手順を実行します。

1. **system-config-authentication** グラフィック・ツールを実行して、ユーザー認証用のユーザー・アカウント・データベースとして LDAP を定義します。
2. **「識別と認証 (Identity Authentication)」**で、**「ユーザー・アカウント・データベース (User Account Database)」**ドロップダウン・リストから**「LDAP」**を選択します。**「LDAP」**オプションを選択すると、Kerberos 付き LDAP 認証を使用して Windows Active Directory ドメインに接続するようシステムを構成できます。


Authentication Configuration
Identity & Authentication
Advanced Options
User Account Configuration
User Account Database:
LDAP
LDAP Search Base DN:
dc=tem,dc=test,dc=co
LDAP Server:
ldap://winserver.tem.te
Use TLS to encrypt connections
Download CA Certificate...
Authentication Configuration
Authentication Method:
Kerberos password
Realm:
TEM.TEST.COM
KDCs:
winserver.tem.test.com:88
Admin Servers:
winserver.tem.test.com:464
Use DNS to resolve hosts to realms
Use DNS to locate KDCs for realms
Revert
Cancel
Apply

3. **「LDAP 検索ベース DN (LDAP Search Base DN)」**で、リストされている識別名 (DN) を使用してユーザー情報を取得することを、`dc=tem,dc=test,dc=com` のように指定します。
4. **「LDAP サーバー (LDAP Server)」**で、LDAP サーバーのアドレスを、`ldap://winserver.tem.test.com` のように指定します。
5. **「認証方式」**で、**「Kerberos パスワード」**を選択します。
6. **「レルム (Realm)」**で Kerberos サーバーのレルム (`TEM.TEST.COM` など) を構成します。レルム名は必ず大文字で入力してください。
7. **「KDCs」**で、Kerberos チケットを発行するための鍵配布センター (KDC) を、`winserver.tem.test.com` のように指定します。
8. **「管理サーバー (Admin Servers)」**で、`kadmind` を実行する管理サーバーを、`winserver.tem.test.com` のように指定します。
9. **「適用」**をクリックします。

このツールの使用法について詳しくは、「Launching the Authentication Configuration Tool UI」 を参照してください。

authconfig コマンド行ツールの使用

システム認証に必要なすべての構成ファイルおよびサービスを更新するために、**authconfig** コマンド行ツールを実行できます。

以下に例を示します。

```
authconfig --enableldap --ldapserver=ldap://winserver.tem.test.com:389
 --ldapbasedn="dc=tem,dc=test,dc=com" --enablekrb5
 --krb5realm TEM.TEST.COM --krb5kdc winserver.tem.test.com:88
 --krb5adminserver winserver.tem.test.com:464 --update
```

各部の意味は以下のとおりです。

--enableldap

Kerberos 付き LDAP 認証を使用してシステムを Windows Active Directory ドメインに接続するよう構成することを指定します。

--ldapserver

LDAP サーバーのアドレスを、`ldap://winserver.tem.test.com` のように指定します。

--ldapbasedn

リストされている識別名 (DN) を使用してユーザー情報を取得することを、`dc=tem,dc=test,dc=com` のように指定します。

--enablekrb5

Kerberos パスワード認証方式を有効化します。

--krb5realm

Kerberos サーバーのレルムを、`TEM.TEST.COM` のように構成します。レルム名は必ず大文字で指定してください。

--krb5kdc

Kerberos チケットを発行するための鍵配布センター (KDC) を、`winserver.tem.test.com` のように指定します。

--krb5adminserver

`kadmind` を実行する管理サーバーを、`winserver.tem.test.com` のように指定します。

--update

すべての構成設定を適用します。

このコマンドの使用法について詳しくは、「Configuring Authentication from the Command Line」 を参照してください。

ローカル LDAP ネームの変更

ローカル LDAP ネームを変更するには、以下の手順を実行します。

1. 以下のようにして、LDAP 構成ファイルのバックアップ・コピーを作成します。

```
cp -p /etc/nslcd.conf /etc/nslcd.conf.bk
```

2. 以下の例に示すように、`/etc/nslcd.conf` ファイルの `base` 設定および `uri` 設定の値を変更します。

```
base dc=tem,dc=test,dc=com
uri ldap://winserver.tem.test.com
```

3. ローカル LDAP ネーム・サービス・デーモンを再開します。

```
service nslcd restart
```

4. ローカル LDAP ネーム・サービス・デーモン (`nslcd`) がサーバーと同時に開始されるよう設定されているようにします。

```
 chkconfig nslcd on
```

NSS ライブラリーおよび PAM ライブラリーの構成

Linux システムでユーザーを認証するために LDAP データベースを使用する方法。

/etc/nsswitch.conf を編集し、`passwd`、`shadow`、および `group` 項目を SSSD デーモン (**sss**) から LDAP に変更します。

```
passwd:  files sss
shadow:  files sss
group:   files sss
```

上記を LDAP (**ldap**) に変更します。

```
passwd:  files ldap
shadow:  files ldap
group:   files ldap
```

PAM ライブラリーを構成するには、`/etc/pam.d/system-auth` ファイルと `/etc/pam.d/password-auth` ファイルを編集して、以下のように `pam_krb5.so` ライブラリー項目を追加します。

```
auth     sufficient                                  pam_krb5.so
use_first_pass
...
account  [default=bad success=ok user_unknown=ignore] pam_krb5.so
...
```

```
password sufficient                                    pam_krb5.so
use_authtok
...
session  optional                                      pam_krb5.so
```

注: SSSD ライブラリーの項目 (`pam_sss.so`) を削除します。

Red Hat の統合について詳しくは、「Integrating Red Hat Enterprise Linux 6 with Active Directory」を参照してください。

サーバーと Active Directory との統合

BigFix サーバーと Active Directory の統合方法

1. **「ツール」**メニューで**「LDAP ディレクトリーの追加」**を選択します。**「LDAP ディレクトリーの追加」** ダイアログが表示されます。

2. Active Directory の名前を入力し、「種類」プルダウンから **「Microsoft Active Directory」** が選択されていることを確認します。
3. **「サーバー」** に、サーバーのホスト名、IP アドレス、または完全修飾ドメイン名を入力します。
4. セキュア接続 (SSL) を構成する場合は、**「SSL の使用」**をクリックします。
5. Active Directory フォレスト全体にアクセスする場合は、**「これはグローバル・カタログ・サーバーです」** をクリックします。
6. ボタンをクリックして、**ルート・サーバーのサービス・ユーザーとして接続**するか、または**資格情報を使用**して接続します。資格情報を使用して接続することを選択した場合は、Active Directory の **「ユーザー名」** と **「パスワード」** を入力します。
7. **「テスト」** をクリックして、入力した情報が正しいこと、および Active Directory サーバーへの接続を確立できることを確認します。
8. **「追加」**をクリックして、Active Directory の設定を完了します。

注: LDAP サーバーを**「Microsoft Active Directory」**として追加する場合は、LDAP サーバー上で、各ユーザーの**「ユーザーのログイン名」**に対応する `UserPrincipalName` 属性を定義していることを確認してください。この属性値は、各ユーザー認証用に BigFix コンソールで使用されます。

LDAP オペレーターの追加

既存の Active Directory アカウントまたは LDAP アカウントを使用して、コンソールにアクセスするオペレーター用のアカウントを作成することができます。

このオプションを選択すると、LDAP ディレクトリーで指定されている名前と同じ名前のオペレーターが、BigFix コンソールのドメイン・パネルの「オペレーター」ノードに追加されます。その後、これらのオペレーターは、以下の表記のいずれかを使用して、通常どおりにログインできるようになります。

username username@domain domain\username

LDAP ディレクトリーの当該ユーザーに割り当てられた許可は、新規に作成されたオペレーターに継承されません。必要な許可をオペレーターに割り当てるか、オペレーターを既存の役割に割り当てる必要があります。

注: Web UI と Web レポートへのアクセスについてはバージョン 9.2.6 以降、コンソールへのアクセスについてはバージョン 9.5 以降で、BigFix LDAP オペレーターに以下を提供するために、BigFix を SAML V2.0 と統合できます。

- Common Access Card (CAC)、Personal Identity Verification (PIV) カード、またはその他の要素を使用した 2 要素認証 (ID プロバイダーによって要求された場合)。
- ID プロバイダーのログイン URL からの Web ベースのシングル・サインオン認証方式。

詳しくは、『LDAP オペレーター用の SAML V2.0 認証の有効化 (*(ページ) 55)*』を参照してください。

LDAP オペレーターを追加するには、以下の手順を実行します。

1. 必要な Active Directory または LDAP ディレクトリーが BigFix 環境に追加されていることを確認してください。
2. **「ツール」>「LDAP オペレーターの追加」**メニュー項目をクリックするか、作業域で右クリックして、**「LDAP オペレーターの追加」**を選択します。「LDAP ユーザーの追加」ダイアログが表示されます。

3. 「検索」フィールドと 2 つのラジオ・ボタンを使用して、指定された LDAP サーバーで定義されているユーザーの照会とフィルタリングを行うことができます。
4. LDAP オペレーターとして追加するユーザーを見つけたら、そのユーザーを選択して**「追加」**をクリックします。「コンソール・オペレーター」パネルが開きます。

5. **「詳細」**タブから、オペレーター権限を割り当てます。

   オペレーターがポストアクションとして再起動およびシャットダウンをトリガーできるようにしたり、再起動およびシャットダウンを BigFix アクション・スクリプトに含めたりすることができます。特定のオペレーターに対して設定したシャットダウンおよび再起動の構成に応じて、「アクションの実行」パネルの「ポストアクション」タブのラジオ・ボタンは、そのオペレーターに対して無効になる場合があります。この構成は、アクション・スクリプトのタイプが BigFix アクション・スクリプトではないアクションに対しては無効です。

   また、BigFix コンソールおよび REST API にアクセスする権限を設定することもできます。
6. **「管理対象コンピューター」**タブには、このオペレーターによって管理されているコンピューターがリストされます。
7. **「割り当てられた役割」**タブで、このオペレーターについて、割り当てる役割、または割り当てを解除する役割を選択します。
8. **「サイト」**タブで、このオペレーターにアクセスを許可するサイトを割り当てます。または割り当てを解除します。

9. **「コンピューターの割り当て」**タブで、このオペレーターが扱えるコンピューターが合致する必要のあるプロパティーを指定します。
10. 変更内容を保存するには、「**変更の保存**」をクリックします。

また、ローカル・オペレーターを、いつでも LDAP オペレーターに変換できます。このためには、次のステップを実行します。

1. 任意のローカル・オペレーター・リストで、変換したいオペレーターを右クリックします。
2. コンテキスト・メニューで、**「LDAP オペレーターに変換」**を選択します。

LDAP グループの関連付け

既存の Active Directory ディレクトリーまたは LDAP ディレクトリーで定義されている LDAP ユーザーまたは LDAP グループを、コンソールのオペレーターまたは役割に関連付けることができます。

このようなグループを追加するには、以下の手順を実行します。

1. 必要な Active Directory または LDAP ディレクトリーが BigFix 環境に追加されていることを確認してください。
2. **「ツール」>「役割の作成」**を選択するか、作業域で右クリックして**「役割の作成」**を選択して、新規グループを受け入れるための役割を作成します。

   グループの名前を入力して**「OK」**をクリックします。
3. **「役割」**パネルが表示されます。

   「LDAP グループ」タブをクリックします。
4. この役割に割り当てる LDAP グループを選択し、**「LDAP グループの割り当て」**をクリックします。
5. 変更内容を保存するには、「**変更の保存**」をクリックします。

LDAP グループを役割に割り当てると、そのグループのすべてのユーザーがコンソールにログインできるようになります。実際にログインしたユーザーだけがアカウントを提供され、オペレーターのリストに表示されます。これにより、不要なアカウントが作成されることがなくなります。オペレーターには、ユーザーのすべての役割とアクセス許可のうち、最も高い特権が付与されます。例えば、あるユーザーが役割 1 からコンピューター・セット A とサイト X にアクセスでき、役割 2 からコンピューター・セット B とサイト Y にアクセスできる場合、オペレーターには、コンピューター・セット A と B の両方でサイト X と Y に対するアクセス許可が付与されます。

第 4 章. LDAP オペレーター用の SAML V2.0 認証の有効化

BigFix のバージョン 9.5.5 以降では、LDAP を基盤とした SAML ID プロバイダー経由の SAML V2.0 認証をサポートしています。

構成後、SAML V2.0 のサポートにより、以下のことが可能になります。

- Common Access Card (CAC)、Personal Identity Verification (PIV) カード、またはその他の要素を使用した、BigFix の 2 要素認証 (ID プロバイダーによって要求された場合)。
- ID プロバイダーのログイン URL からの Web ベースのシングル・サインオン認証方式。要求に応じて、ログイン・ユーザーは再度ログインしなくても、SAML V2.0 認証をサポートしている Web ベースのコンポーネントに自動的にリダイレクトされます。

SAML 2.0 とは

OASIS の Security Assertion Markup Language (SAML) は、XML ベースのフレームワークを使用して、オンライン・エンティティー間でセキュリティー情報を記述し交換します。

SAML 2.0 は以下をサポートします。

Web ベースのシングル・サインオン

サーバーの DNS ドメインに関係なく、Web サーバー間でユーザーについての情報を転送するための、ベンダーに依存しない標準的な文法およびプロトコルを提供します。

ID 連携

パートナー・サービスがユーザーの共通名 ID について合意し、この ID を設定して、組織の境界を越えてユーザー自身についての情報を共有できるようにします。

このタイプの共有は、ID 管理コストを減らすのに役立ちます。

連携された ID は FIPS 201 を実装して、米国政府全体にわたって相互利用可能な ID 資格情報を定義します。この資格情報は Personal Identity Verification

(PIV) と呼ばれ、連邦機関の施設への物理的アクセスおよび連邦情報システムへの論理的アクセスを制御するために使用されます。

CAC PIV カードは、PIV カード所有者認証用のマルチアプリケーション・スマート・カードであり、これには、線形バーコード、2 次元バーコード、磁気ストライプ、カラーのデジタル写真、および印刷テキストが組み込まれています。このカードは、以下に対応するトークンとして機能します。

- コンピューター・システムへの論理的アクセス
- 個人の識別
- 建物への物理的アクセス
- 署名用、暗号化用、および否認防止用の Public Key Infrastructure (PKI)。

Web サービスおよびその他の業界標準

SAML により、そのセキュリティー・アサーション形式を「ネイティブ」の SAML ベースのプロトコル・コンテキストの外部で使用できるようになります。このモジュール性は、許可サービス (IETF、OASIS)、ID フレームワーク、Web サービス (OASIS、Liberty Alliance) などを扱う他の業界の取り組みにとって有益であることが実証されています。

SAML の仕組み

SAML 仕様では、以下の 3 つの関係者が定義されています。

これらは次のとおりです。

- プリンシパル。通常はユーザーです。
- ID プロバイダー (IdP)。LDAP を基盤とした SAML ID プロバイダーです。
- サービス・プロバイダー (SP)。今回のケースでは、BigFix サービスです。

SAML 規格は、ID アサーションが上記の 3 つの関係者の間でどのように交換されるかを制御します。SAML は、ID プロバイダーでの認証方法は指定しません。

SAML では、1 つの ID プロバイダーが多数のサービス・プロバイダーに SAML アサーションを提供できます。

SAML V2.0 のユース・ケースのシナリオについて詳しくは、「SAML V2.0 の概要」を参照してください。

SAML V2.0 と統合される BigFix ユーザー・インターフェースの種類

SAML 認証機能拡張が構成されると、Web UI、Web レポート、および BigFix バージョン V9.5.5 以降の BigFix コンソールにアクセスする、すべての BigFix LDAP 管理対象ユーザーに影響を及ぼします。

BigFix と SAML V2.0 との統合の仕組み

SAML V2.0 との統合では、passport-saml 認証プロバイダーを使用して、ID プロバイダー (IdP) が開始した認証とサービス・プロバイダー (SP) が開始した認証の両方に対応します。

SAML をサポートするどの BigFix ユーザー・インターフェースについても、SAML の使用および要求は WebUI コンポーネントによって管理されます。

SAML との統合を構成する方法は、以下のように計画している使用法によって異なります。

- SAML 認証を Web レポートおよび BigFix コンソールでのみ使用して、どの WebUI アプリケーションでも使用する必要がない場合は、WebUI を SAML 専用モードで開始できます。この SAML 構成では、リソース消費量を最小限に抑えることができます。この構成のセットアップ方法について詳しくは、「SAML 専用モードでの WebUI の有効化」を参照してください。
- WebUI コンポーネントの完全なセットまたは WebUI ETL 処理を含む、すべてのBigFix ユーザー・インターフェースで SAML 認証を使用するには、BigFix バージョン 9.5.5 以降を使用している場合は WebUI のインストールに記載されている手順に従います。

BigFix 環境で 1 つの LDAP サーバーをユーザー・リポジトリーとして使用している場合、ユーザー・プロビジョニングはこの統合による影響を受けません。管理者は引き続き、BigFix サービスを使用する権限を与えるために、オペレーターおよび役割を定義しま

す。ご使用の BigFix 環境のオペレーターが複数の LDAP サーバーで定義されている場合は、前提事項と要件 (（ページ） 58)に記載されている情報をよくお読みください。

SAML 2.0 と統合した場合、既存の監査シナリオは維持され、`server_audit.log` ファイルに SAML 認証のユーザー・エントリーが含まれます。

以下のサンプル・ユース・ケースを確認してください。

1. ユーザーが BigFix にサービスを要求します。例えば、Web UI、Web レポート、または BigFix コンソールを使用して、ページにアクセスしたり、ログインを試行したりします。
2. BigFix は、LDAP を基盤とした SAML ID プロバイダーに ID アサーションを要求します。
3. ID アサーションを配信する前に、LDAP を基盤とした SAML ID プロバイダーは、何らかのユーザー認証情報、例えば、ユーザー名とパスワードや、別の形式の認証 (多要素認証 を含む) を要求する可能性があります。ディレクトリー・サービス (LDAP や Active Directory など) は、ID プロバイダーでの代表的な認証トークンのソースです。
4. ID プロバイダーが提供するID アサーションに基づいて、BigFix は、そのユーザーによって要求されたサービスを実行するかどうかを決定します。
5. 認証情報は維持され、割り当てられた許可に応じて、ユーザーが BigFix によって提供されるサービスに自動でアクセスできるようにするために使用されます。

前提事項と要件

SAML V2.0 を使用できるよう BigFix を構成する前に、以下の前提事項および要件のリストをよくお読みください。

- BigFix は、SAML V2.0 準拠の ID プロバイダー (Active Directory フェデレーション・サービス (ADFS) など) による SAML V2.0 認証をサポートします。
- SAML V2.0 認証は、以下のように制限されています。
    - 1 つ以上の LDAP ディレクトリーによって、1 つの SAML IdP のみがサポートされます。ご使用の BigFix 環境で複数の LDAP サーバーをユーザー・リポジトリーとして既に定義している場合、SAML 認証を有効にした後は、選択した IdP によって管理されるユーザーおよびグループのみが BigFix 環境に引き続き認識されることに注意してください。この場合、ユーザー・リポジトリーとし

て使用する別の LDAP 環境のユーザーを SAML IdP (ADFS または ISAM) が認証できるように、IdP 環境が正しく構成されていることを確認してください。
    - ID プロバイダーは、セキュア・ハッシュ・アルゴリズムとして SHA256 を使用します。
    - Web レポート・サーバーは、1 つのデータ・ソース (ルート・サーバー) のみに接続し、SSL を使用するように構成されます。
- SAML 認証を構成して使用するには、WebUI がインストールされている必要があります。Web レポートおよび BigFix コンソール用の SAML 認証を提供するためにのみ WebUI を使用する場合、Web UI を SAML 専用モードで開始してリソース消費量を削減できます。Web UI を SAML 専用モードで開始する方法について詳しくは、「WebUI ユーザーズ・ガイド」を参照してください。
- DSA アーキテクチャーでは、構成がレプリカの DSA サーバーに複製されます。ただし、レプリカはプライマリーでない DSA での SAML 用の WebUI を有効にしません。複数 WebUI 構成はサポートされていないためです。
- パッチ 1 以降、サーバーの署名キーとして使用される X.509 証明書は、ルート・サーバーの DNS 名と IP を含む *subjectAltName* フィールドとともに生成されます。これにより、認証プロセス中に「セキュリティ証明書の名前が無効であるか、サイト名と一致しない」セキュリティー警告が現れるのを防ぎます。
    - フレッシュ・インストールの場合は、プロセス中に新しい証明書が作成されます。
    - アップグレードの場合は、古い証明書が所定の位置に残されます。セキュリティー警告を防ぐには、以下のステップを実行します。
        - 接続しているサーバーのサーバー署名キーをローテーションします。パラメーターについて詳しくは、「追加の管理コマンド」を参照してください。DSA アーキテクチャーでは、すべてのサーバーのキーをローテーションする必要はありません。

          重要: この操作により、すべての既存のコンテンツを再署名します。非常に大規模なデプロイメントでは、数時間かかることが

あります。日々のデプロイメント操作への影響を最小限にするには、メンテナンス期間を計画します。

- 代替方法として、「BigFix ユーザーの観点から見た変更内容 ((ページ) 60)」で説明されている回避策を適用します。

- Web レポートを実行時、SAML が有効な場合は、参照者に関するチェックは実行されません。設定 `_HTTPServer_Referrer_CheckEnabled` を使用して、参照者チェックを有効または無効にできます。参照者は、HTTP プロトコルのオプションのヘッダーです。これは、要求されているリソースへのリンク元である Web ページのアドレス (URI または IRI) を識別します。BigFix がどのように参照者チェックを管理するかについては、設定のリストと詳細な説明 ((ページ))を参照してください。

BigFix ユーザーの観点から見た変更内容

BigFix ユーザー・インターフェースの観点から見て、この機能拡張が影響を及ぼすのは認証のみです。

LDAP ユーザーに対する SAML 認証を有効にした後:

LDAP オペレーター:

- SAML ID プロバイダーのみから Web UI および Web レポートへの認証を行う必要があります。このためには、以下の URL にアクセスします。

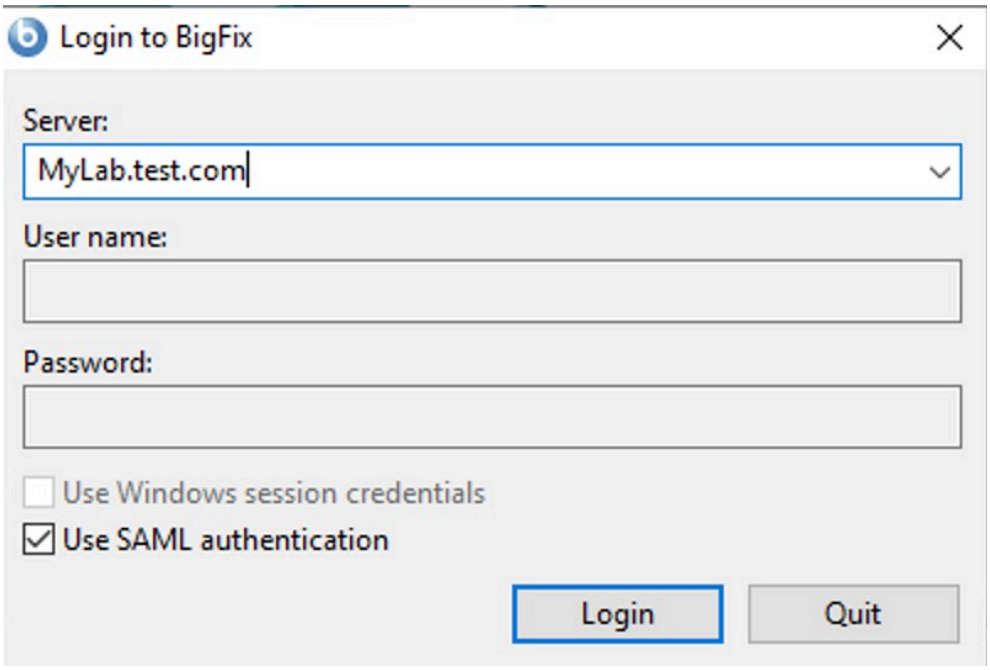
  `https://<WebUI_server>` (Web UI サーバーの場合。ポート 443 を使用していると想定)

  `https://<Web_Reports_server>:8083` (各 Web レポート・サーバーの場合。ポート 8083 を使用していると想定)

  注: Web UI および Web レポートからログアウトするためのボタンおよびリンクは、これらのユーザーをあるページにリダイレクトします。そのページで「再認証」ボタンをクリックすると、IdP のログイン・タイムアウトにならない限り、再度ログオンしなくても、Web UI ページおよび Web レポート・ページ

に戻ることができます。ログイン・タイムアウトの場合は、IdP のログイン・ページに戻されます。

- BigFix サーバーが SAML V2.0 と統合するように構成されている場合、コンソール・ログイン・パネルで**「SAML 認証を使用する」**チェック・ボックスを有効にする必要があります。



選択内容は、今後のログイン要求のために、BigFix によって自動的に検証され保存されます。

ローカルの非 LDAP オペレーター:

- 以下の通常のログイン URL にアクセスして、Web UI または Web レポートにログインします。

  `https://<WebUI_server>/login` (Web UI がポート 443 に設定されていると想定)

  `https://<Web_Reports_server>:8083/login` (各 Web レポート・サーバーの場合。Web レポートがポート 8083 に設定されていると想定)

- 通常のログイン・パネルから BigFix コンソールにログインします。その際、**「SAML 認証を使用する」**チェック・ボックスが選択されていないことを確認してください。

  注: 環境内で SAML が有効化されていない場合は、**「SAML 認証を使用する」**チェック・ボックスがグレーで表示されます。

SAML が構成され有効化された後で初めて、ローカルの 非 LDAP ユーザーが API を使用してログインできるようになります。4-Eye 認証の承認者はローカル・アカウントでなければなりません。

SAML 2.0 と統合するように BigFix を構成する方法

SAML ID プロバイダーと BigFix サーバーを構成する方法。

統合を構成する前に、以下を確認してください。

- BigFix サーバーが、ID プロバイダーのログイン・ページ用の URL で使用されているホスト名を解決できる。
- ID プロバイダー (ADFS サーバー、またはサポートされている他のタイプの SAML 認証プロバイダー) が、Web UI、Web レポート、および BigFix コンソールと通信するために使用されるリダイレクト URL で指定されている BigFix ルート・サーバーのホスト名を解決できる。
- Web UI は有効かつアクティブである。

全体的な構成は、以下の 2 つの部分から成ります。

- 明示的な 2 要素認証用の SAML ID プロバイダーの構成。これは、ID プロバイダー管理者の責任下で行います。この部分については、以下を確認してください。
  - リダイレクト URL が、インデックス付きの証明書利用者信頼に、バインディング HTTPS_POST を使用して以下の形式で追加されている。

    `https://<WebUI_server>/saml` (Web UI サーバーの場合。ポート 443 で listen していると想定)

`https://<Web_Reports_server>:8083/saml` (各 Web レポート・サーバーの場合。ポート 8083 で listen していると想定)

`https://<Bigfix_server>:52311/saml` (BigFix コンソールの場合)

注: ID プロバイダーが ADFS である場合、ADFS の証明書利用者信頼のプロパティーの「エンドポイント」タブに SAML アサーション・コンシューマー・エンドポイントとしてリダイレクト URL を追加する必要があります。

- ID プロバイダー構成で、ログイン設定が FORMS ログイン用に設定されている必要があります。
- スマート・カード認証を使用する予定の場合は、ID プロバイダーが多要素認証を使用するように正しく構成されていることを確認してください。例えば、ADFS を使用する場合は、グローバル認証ポリシー構成で、証明書認証と Windows 認証 (Windows 統合認証を使用する場合) の間で少なくとも 1 つが有効になっていることを確認してください。
- Active Directory ユーザー認証の場合、ID プロバイダーの要求規則を以下のように設定します。

  属性ストア:

  Active Directory

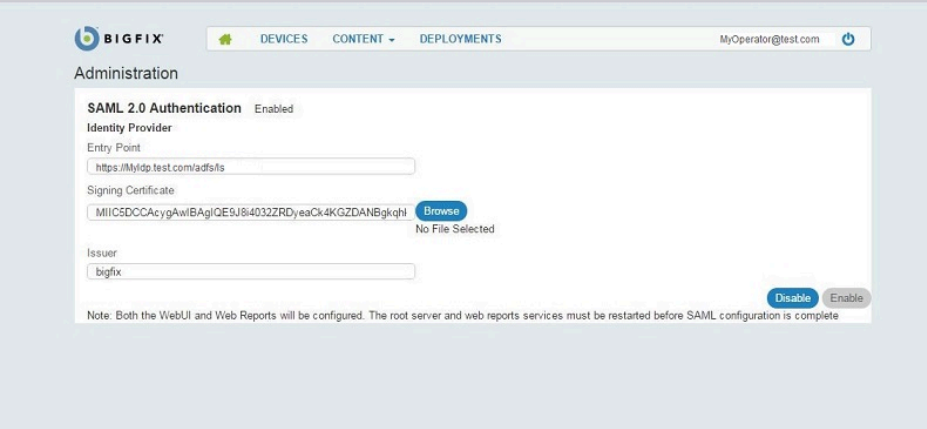
  発信要求タイプへの LDAP 属性のマップ:

  - LDAP 属性: User-Principal-Name
  - 発信要求: 名前 ID

- BigFix サーバーが SAML 認証を使用できるようにするための構成。これは、マスター・オペレーター (MO) および Web レポート管理者の責任下で行います。このタスクを実行するには、以下の手順を実行します。
  1. BigFix コンソールで Active Directory を使用して LDAP を構成します。詳しくは、『Windows サーバーと Active Directory との統合 (（ページ） 40)』を参照してください。
  2. LDAP オペレーターを定義します。詳しくは、『LDAP オペレーターの追加 (（ページ） 51)』を参照してください。

3. Web レポートのユーザー管理ページで Web レポートの LDAP オペレーターを定義します。
4. 「管理」ページにアクセスして、SAML 2.0 との統合を構成します。
   a. Web UI サーバーにログインします。
      - Web UI がポート 443 で listen している場合: `https://<WebUI_server>`
      - Web UI が 443 以外のポートで listen している場合: `https://<WebUI_server>:<webui_port_number>`

   b. 管理者ページを開きます。
      - Web UI がポート 443 で listen している場合: `https://<WebUI_server>/administrator`
      - Web UI が 443 以外のポートで listen している場合: `https://<WebUI_server>:<webui_port_number>/administrator`



5. 「管理」ページで、以下を指定します。

   エントリー・ポイント:

   ID プロバイダーのログイン URL。オペレーターがログインに使用できる URL で、オペレーターはここから Web UI または Web レポートにリダイレクトされます (例: `https://<idp_fqdn>/adfs/ls`)。

   署名証明書:

証明書ファイルを参照するか、Base-64 エンコード X.509 (.CER) 形式の ID プロバイダー証明書のキーをこのフィールドに貼り付けます。

発行者:

ID プロバイダーの ID をテキスト形式で入力します。例えば「BigFix」です。ADFS 構成を構成している場合、この値は ADFS 証明書利用者 ID 設定と一致している必要があります。

6. すべてのフィールドに入力したら、**「有効化」** をクリックします。
7. WebUI が別のリモート・サーバーにインストールされている場合は、WebUI 証明書の件名と一致することを確認し、BigFix サーバー・コンピューターの `_WebUI_AppServer_Hostname` キーを、WebUI がインストールされているコンピューター (WebUI サーバー・コンピューター) のホスト名、完全修飾ドメイン・ネーム (FQDN)、または IP アドレスに設定します。デフォルトの WebUI ポートが変更された場合 (`_WebUIAppEnv_APP_PORT`) は、新しい WebUI ポートを使用するように BigFix サーバー・コンピューターの `_WebUI_Monitor_Port` キーを設定する必要があります。

   注: WebUI サーバーのポート (デフォルトの HTTPS 5000) が BigFix ルート・サーバーから到達可能であることを確認してください。

8. Web ベースのシングル・サインオン (SSO) 認証方式を有効にする場合は、WebUI マシンで _WebUIAppEnv_SAML_SSO_ENABLE キーを 1 に設定します。
9. SAML 認証方式としてスマート・カードの使用を有効にする場合は、WebUI サーバー・コンピューターで、`_WebUIAppEnv_SAML_AUTHNCONTEXT` 設定を以下の 2 つの値のいずれかに設定します。
   - `urn:oasis:names:tc:SAML:2.0:ac:classes:TLSClient` (ID プロバイダーが Transport Layer Security (TLS) 暗号プロトコルを使用するように設定されている場合)。
   - `urn:federation:authentication:windows` (ID プロバイダーが統合 Windows 認証 (IWA) を使用するように設定されている場合)。
10. BigFix ルート・サーバーを再始動します。

11. BigFix Web レポート・サービスを再始動します。
12. WebUI サービスを再起動します。

この構成をセットアップして BES ルート・サーバーを再始動した後に、Web UI が再始動するまでに、少し時間がかかることがあります。

上記の手順が正常に実行されたら、これらのサービスのすべての LDAP オペレーターを、構成された ID プロバイダーを介して認証する必要があります。

管理者は「管理」ページを使用して、既存の構成を更新することもできます。

注: これらの手順を完了した後、BigFix コンソールにログオンする際のエラーを防ぐ処置として、分数で指定される `_BESDataServer_AuthenticationTimeoutMinutes` 構成設定に 5 分を超える値を設定してください。

第 5 章. ローカル・オペレーターの無効化

BigFix バージョン 10.0.8 以降では、ローカル・オペレーターが BigFix コンソールにログインできないようにして、代わりに LDAP オペレーターを使用できます。

ローカル・オペレーターを使用できないようにする方法について詳しくは、追加の管理コマンド (（ページ）) および BigFix 管理ツールの実行 (（ページ）)で説明されている **securitysettings** を参照してください。

ローカル・オペレーターを使用不可にした後、ローカル・オペレーターが BigFix コンソールにログインしようとすると、コンソールでは以下のエラー・メッセージが表示されます。

```
The login operation using a local operator is not allowed.
```

さらに、LDAP ユーザーを使用してログインし、ローカル・オペレーターが無効化されている場合、ローカル・オペレーターおよび LDAP ユーザーの両方によって同じ操作が行われることを回避するため、いくつかのメニュー項目、ボタン、フォームがグレー表示されます。LDAP グループを BigFix ロールに関連付け、BigFix ロールに関連付けられた LDAP グループに属する LDAP ユーザーを使用してログインすることは、引き続き可能です。

ロールは、LDAP グループ - BigFix ロールの関連付けに沿って継承されます。

「ツール」では、以下のメニュー項目がグレー表示されます。

「オペレーター」フォームのコンテキスト・メニューでは、次の項目がグレー表示されます。LDAP オペレーターに対して**「 削除 」**ボタンは引き続き有効になります。

オペレーターを選択すると、「オペレーター」フォームで次のボタンがグレー表示されます。**変更の保存、変更の破棄、削除、パスワードのリセット**。LDAP オペレーターに対して**「削除」**ボタンは引き続き有効になります。

「オペレーター」フォームで、**「詳細」**タブが変更されます。**「アクセス制限」**グループと**「明示的な権限」**列は非表示になります。さらに、**「有効な権限」**列は**「権限」**で名前変更されます。

「オペレーター」フォームの「割り当てられた役割」タブでは、**「役割の割り当て」**ボタンと**「役割の削除」**ボタンがグレー表示されます。

「オペレーター」フォームの「サイト」タブでは、次のボタンがグレー表示されます。**サイトの割り当て、サイトの削除、所有者、作成者、読者**。さらに、**「明示的な権限」**列は非表示になり、**「権限」**の**「有効な権限」**列の名前が変更されます。

「オペレーター」フォームでは、**「コンピューターの割り当て」**タブが非表示になります。

「オペレーター」フォームの「WebUI アプリ」タブでは、**「権限」**ボタンと**「なし」**ボタンがグレー表示されます。さらに、**「明示的な権限」**列は非表示になり、**「許可」**の**「有効な権限」**列の名前が変更されます。

「役割」フォームの「オペレーター」タブでは、**「ユーザーの割り当て」**ボタンと**「ユーザーの削除」**ボタンがグレー表示されます。

WebUI アプリ・フォームの「オペレーターの権限」タブでは、**「権限」**ボタンと**「なし」**ボタンがグレー表示されます。さらに、**「明示的な権限」**列は非表示になり、**「権限」**の**「有効な権限」**列の名前が変更されます。

第 6 章. 複数サーバー (DSA) の使用

複数のサーバー・インストールの一部の重要な要素。

- 追加のサーバーをインストールするプラットフォームに応じて、追加 Windows サーバーのインストール (DSA) (（ページ　　)または追加 Linux サーバーのインストール (DSA) (（ページ　　)に説明されている手順を実行します。
- サーバーは、定期的なスケジュールに基づいて通信し、データを複製します。現在の状況を確認し、レプリケーション間隔を調整するには、Windows システムでのレプリケーションの管理 (DSA) (（ページ　74)またはLinux システムでのレプリケーションの管理 (DSA) (（ページ　75)を参照してください。
- 各サーバーは、適用環境内の他のサーバーから複製を行う準備ができると、適用環境内の他のすべてのサーバーへの最短パスを計算します。プライマリー・リンクには長さ 1 が、セカンダリー・リンクには長さ 100 が、ターシャリー・リンクには長さ 10,000 がそれぞれ割り当てられます。前回使用されたときに接続が失敗に終わったリンクは未接続と見なされます。
- 停止またはその他の問題が原因でネットワークの分断が発生した場合、カスタム Fixlet または取得プロパティーは、分断されたネットワークの両側で独立して変更できます。ネットワークが再接続されると、最も小さいサーバー ID を持つサーバーのバージョンが優先されます。
- **Web レポート**の複数のコピーがインストールされている場合、それらは独立して動作します。各 Web レポート・サーバーはすべて、データベースの同等のビューを所有しているため、最も便利なサーバーに接続できます。
- デフォルトでは、サーバー 0 (ゼロ) がマスター・サーバーです。マスター・サーバーに接続している場合、Windows での**BigFix管理ツール**や Linux での **BESAdmin** コマンドを使用すると、特定の管理用タスク (ユーザーの作成、削除など) のみ実行できます。

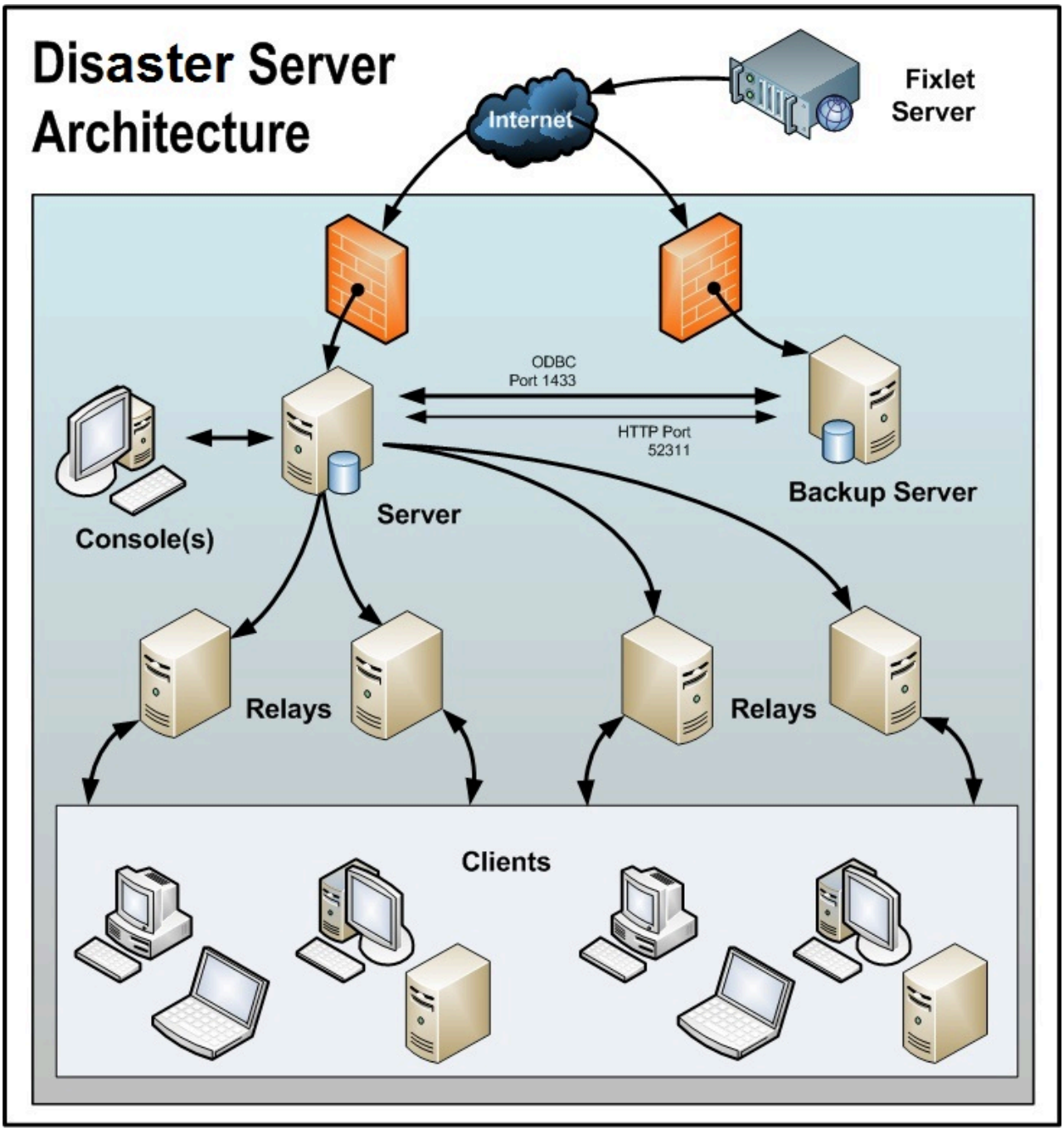
災害対策サーバー・アーキテクチャー (DSA)

以下の図は、2 つのサーバーを使用した典型的な DSA セットアップを示しています。それぞれのサーバーがファイアウォールの背後にあります。複数のサーバーを単一のオフィス

にセットアップすれば簡単ですが、ここでは、それらのサーバーは別々のオフィスにあると思われます。

これらのサーバーでは、BigFix データを複製するための高速な接続が必要になります (通常は、10 Mbps から 100 Mbps の LAN 速度が必要です)。BigFix サーバーは、ODBC および HTTP プロトコルを介して通信します。

フェイルオーバーが発生した場合は、特定の構成済みリレーが自動的にバックアップ・サーバーを検出し、ネットワークを再接続します。リレーの構成について詳しくは、リレー・フェイルオーバーの構成 (（ページ） 71)を参照してください。



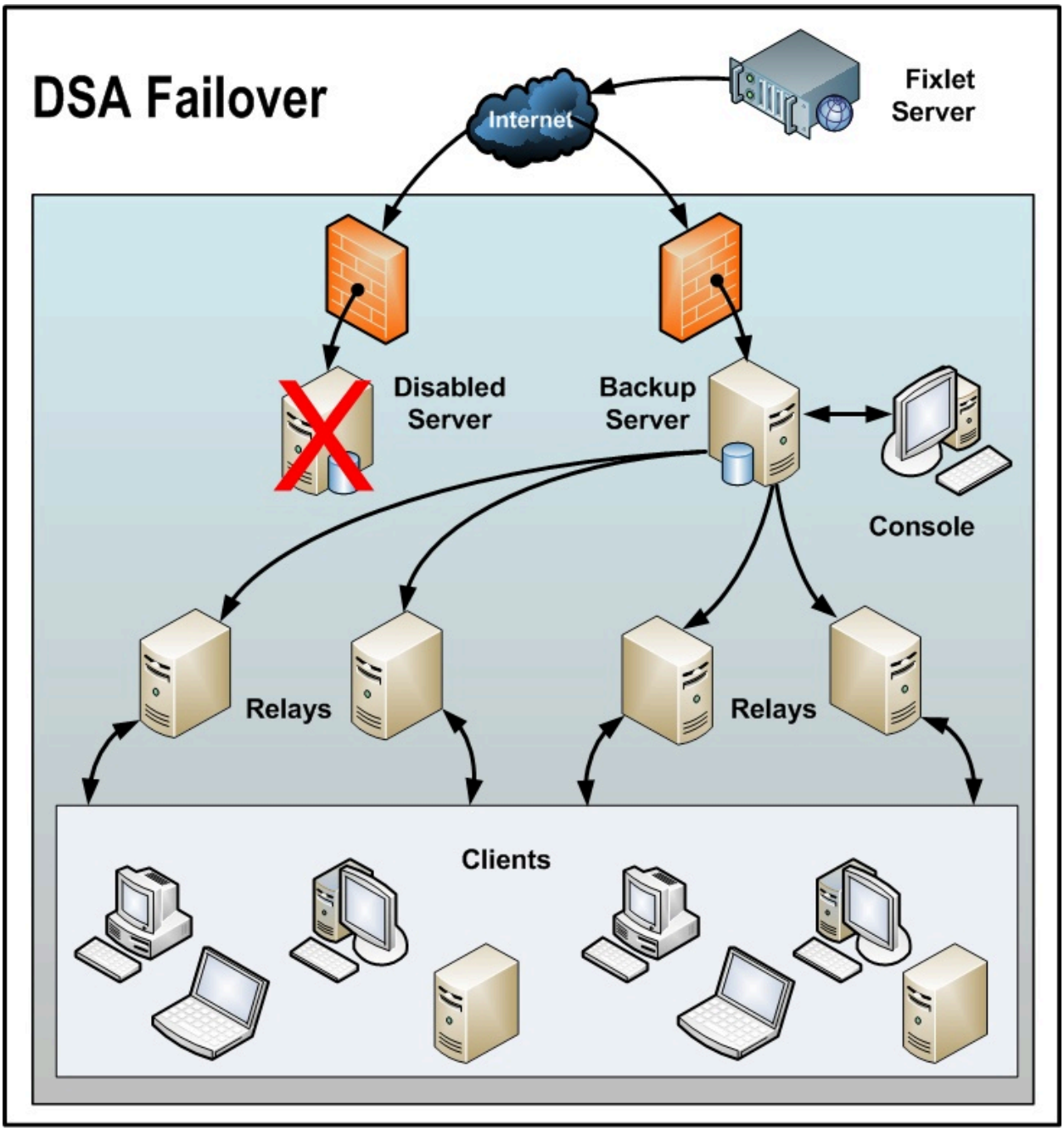
リレー・フェイルオーバーの構成

災害や計画的メンテナンスが原因で BigFix サーバーがダウンした場合に、DSA サーバーを使用して新規のサーバー接続が検出されることがあります。使用不可になっていたサー

バーがオンラインに戻ると、そのサーバーのデータは自動的に正常なサーバー上のデータとマージされます。

フェイルオーバー処理が正常に行われるためには、クライアント設定で最上位リレーに対して `__RelayServer2` を使用して (あるいはコンソール・コンピューターから右クリックで設定するユーザー・インターフェースを使用して) DSA サーバーをセカンダリー・リレーとして設定します。プライマリー BigFix サーバーで障害が発生して、下位の BigFix リレーが報告できない場合は、通常のリレー選択プロセスでセカンダリー BigFix リレー値を使用してセカンダリー BigFix サーバーが検出され、そのサーバーに報告が行われます。

注: クライアント・システムで指定された設定 `_BESClient_RelaySelect_ResistFailureIntervalSeconds` は、フェイルオーバーのタイミングに影響を及ぼす可能性があります。その値の範囲は 0 秒から 6 時間であり、これにより、クライアントがエラー報告を無視する時間 (秒数) を定義します。これを過ぎると、クライアントは別の親リレーの検出を試行します。デフォルト値は 10 分です。フェイルオーバー構成の場合は、`_BESClient_RelaySelect_ResistFailureIntervalSeconds` (定義されている場合) を小さい値に設定するようにしてください。



メッセージ・レベルの暗号化と DSA

メッセージ・レベルの暗号化が有効であり、かつクライアントが**「タスク: BES クライアントの設定: レポートの暗号化」**を使用して設定されている場合は、BigFix サーバー暗号化キーをセカンダリー BigFix DSA サーバーに移動します。

これにより BigFix DSA サーバーは、通常の操作中、またはプライマリー BigFix サーバーで障害が発生した場合に、暗号化された BigFix クライアントからのレポートを処理できるようになります。

以下の BigFix サーバー・ディレクトリーから暗号化キー (`.pvk`) を選択します。

- Windows サーバー: `%PROGRAM FILES%\BigFix Enterprise\BES Server\Encryption Keys\`
- Linux サーバー: `/var/opt/BESServer/Encryption Keys`

DSA セカンダリー・サーバーにコピーします。

Windows システムでのレプリケーションの管理 (DSA)

追加の Windows サーバーをインストールするには、追加 Windows サーバーのインストール (DSA) (（ページ）　　)で説明されている手順に従います。

場合により、間隔の変更や、サーバーのさまざまな割り当てを行う必要があります。これらの変更のほとんどは、BigFix 管理ツールを通じて実行します。以下の方法により、サーバーの現在の設定を確認すること、および適切な変更を加えることができます。

Windows システムでのレプリケーション間隔の変更

Windows システムで、適用環境内に複数のサーバーが存在する場合は、それぞれのレプリケーションを行うタイミングをスケジュールすることができます。

デフォルトは 5 分ですが、この時間を短くしてリカバリー可能性を向上させることや、時間を増やしてネットワーク・アクティビティーを制限することができます。

1. **BigFix管理ツール** を起動します。
2. **「レプリケーション」**タブを選択します。
3. 「更新」ボタンをクリックして、最新の「**レプリケーション・グラフ (Replication Graph)**」を表示します。
4. ドロップダウン・メニューから目的のサーバーを選択します。レプリケーション間隔を長くすると、サーバーがデータを複製する頻度が低くなりますが、1 回に転送する

データの量は多くなります。レプリケーション間隔は、「サーバーからの複製」の場合と「サーバーへの複製」の場合で異なる可能性があることに注意してください。

5. 右側のメニューからレプリケーション間隔を選択します。
6. **「OK」**をクリックします。

Windows システムでのマスター・サーバーの切り替え

デフォルトでは、サーバー 0 (ゼロ) がマスター・サーバーです。マスター・サーバーに接続している場合のみ、管理者ツールを使用して特定の管理用タスク (ユーザーの作成や削除など) を実行することができます。

マスターを別のサーバーに切り替える場合、適用オプション **masterdatabaseServerID** をその別サーバーの ID に設定する必要があります。その方法を以下に示します。

1. **BigFix管理ツール**を起動します。
2. **「詳細オプション」**タブを選択し、**「追加」**をクリックします。
3. 名前として masterDatabaseServerID と入力し、値として別サーバーの ID を入力します。
4. **「OK」**をクリックします。

値が新しいサーバーに正常に複製された後は、その新しいサーバーがマスター・サーバーになります。サーバーがマスターである間にそのサーバーで障害が発生した場合、データベースの ADMINFIELDS テーブルを直接操作することにより、別のサーバーをマスター・サーバーにする必要があります。この詳細は本書の対象範囲外ですが、簡単に言うと、SQL Enterprise Manager などのツールを使用して ADMINFIELDS テーブルを表示し、変更します。変数名 masterDatabaseServerID を目的の値に設定します。

Linux システムでのレプリケーションの管理 (DSA)

追加の Linux サーバーをインストールするには、追加 Linux サーバーのインストール (DSA) (（ページ　　）で説明されている手順に従います。

場合により、間隔の変更や、サーバーのさまざまな割り当てを行う必要があります。これらの変更のほとんどは、`iem` コマンド・ラインを通じて実行します。以下の方法により、サーバーの現在の設定を確認すること、および適切な変更を加えることができます。

Linux システムでのレプリケーション間隔の変更

Linux システムで、適用環境内に複数のサーバーが存在する場合は、それぞれのレプリケーションを行うタイミングをスケジュールすることができます。

デフォルトは 5 分ですが、この時間を短くしてリカバリー可能性を向上させることや、時間を増やしてネットワーク・アクティビティーを制限することができます。

レプリケーション間隔を変更するには、以下の手順を実行します。

1. `/opt/BESServer/bin` コマンド・プロンプトから、以下のようにコマンド・ラインを開始します。

   ```
   ./iem login --server=servername:serverport --user=username
   --password=password
   ```

2. `/opt/BESServer/bin` コマンド・プロンプトから、以下のコマンドを実行します。

   ```
   ./iem get replication/server/0 > /appo/replicationServer0.xml
   ```

3. `/appo/replicationServer0.xml` ファイルで、以下のキーワードを編集します。

   ```
   <ReplicationIntervalSeconds>300</ReplicationIntervalSeconds>
   ```

   これにより、レプリケーション間隔の秒数の値が変更されます。レプリケーション間隔を長くすると、サーバーがデータを複製する頻度が低くなりますが、1 回に転送するデータの量は多くなります。

   ```
   <?xml version="1.0" encoding="UTF-8"?>
   <BESAPI xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

    xsi:noNamespaceSchemaLocation="BESAPI.xsd">
      <ReplicationServer
    Resource="http://9.87.126.68:52311/api/replication

    /server/0">
              <ServerID>0</ServerID>
              <URL>http://mycompany.com:52311</URL>
              <DNS>mycompany.com</DNS>
   ```

```
      <ReplicationIntervalSeconds>300</ReplicationIntervalSeconds>
      <ReplicationLink
 Resource="http://9.87.126.68:52311/api/replication
       /server/0/link/3">
              <SourceServerID>0</SourceServerID>
              <DestinationServerID>3</DestinationServerID>
              <Weight>1</Weight>
              <IsConnected>0</IsConnected>
              <LastReplication>Fri, 01 Mar 2013 11:17:12 +0000
              </LastReplication>
              <LastError>19NoMatchingRecipient - Fri, 01 Mar 2013
 11:17:12 +0000
              </LastError>
      </ReplicationLink>
      <ReplicationLink
 Resource="http://9.87.126.68:52311/api/replication/server/
                                        3/link/0">
                  <SourceServerID>3</SourceServerID>
                  <DestinationServerID>0</DestinationServerID>
                  <Weight>1</Weight>
                  <IsConnected>1</IsConnected>
                  <LastReplication>Fri, 01 Mar 2013 11:17:18 +0000
                  </LastReplication>
      </ReplicationLink>
   </ReplicationServer>
</BESAPI>
```

4. 以下のコマンドを実行して、変更したファイルをアップロードします。

```
./iem post /appo/replicationServer0.xml  replication/server/0
```

Linux システムでのマスター・サーバーの切り替え

デフォルトでは、サーバー 0 (ゼロ) がマスター・サーバーです。

マスターを別のサーバーに切り替えるには、以下のようにして適用オプション `masterDatabaseServerID` をその別のサーバーの ID に設定します。

1. `/opt/BESServer/bin` コマンド・プロンプトから、以下のようにコマンド・ラインを開始します。

   ```
   ./iem login --server=servername:serverport --user=username
    --password=password
   ```

2. `/opt/BESServer/bin` コマンド・プロンプトから、以下のコマンドを実行します。

   ```
    ./iem get admin/fields > /appo/switchmaster.xml
   ```

3. `/appo/switchmaster.xml` ファイルで、以下のキーワードとその値を追加または編集します。

   ```
     <Name>masterDatabaseServerID<Name>
     <Value>0</Value>
   ```

   以下のようにマスター・サーバーを別のマスター・サーバーに切り替えます。

   ```
   <?xml version="1.0" encoding="UTF-8"?>
   <BESAPI xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:noNamespaceSchemaLocation="BESAPI.xsd">
       <AdminField Resource="http://9.87.126.68:52311/api/admin/field
        /masterDatabaseServerID">
          <Name>masterDatabaseServerID</Name>
          <Value>3</Value>
       </AdminField>
   </BESAPI>
   ```

4. 以下のコマンドを実行して、変更したファイルをアップロードします。

   ```
    ./iem post /appo/switchmaster.xml admin/fields
   ```

値が新しいサーバーに正常に複製された後は、その新しいサーバーがマスター・サーバーになります。サーバーがマスターである間にそのサーバーで障害が発生した場合、データ

ベースの ADMINFIELDS テーブルを直接操作することにより、別のサーバーをマスター・サーバーにする必要があります。

アップグレード時に再生成されるスキーマ・ネクスト・テーブル

Windows と Linux システムの両方で DSA 環境を古いバージョンから BigFix バージョン 10 にアップグレードする場合、WebUI コンポーネントで使用されるスキーマ・ネクスト・テーブルは、複製プロセスが正常に完了しなかった場合、レガシー・データベース・テーブルから再生成されます。

この動作は、`DisableReplicationOfNextTables` レジストリー設定 (Windows) または `DisableReplicationOfNextTables` 構成設定 (Linux) を有効にした DSA 環境にのみ影響します。この設定は、お使いの環境でのスキーマ・ネクスト・テーブルの複製を防ぎます。

アップグレード前

`DisableReplicationOfNextTables` レジストリー設定 (Windows) または `DisableReplicationOfNextTables` 構成設定 (Linux) を有効にした場合、スキーマ・ネクスト・テーブルの再生成にかかる時間を短縮するために、「BigFix 管理者ツール」タブの「クリーンアップ」を使用して、環境のクリーンアップを実行することをお勧めします。詳しくは、クリーンアップ (（ページ）　　)を参照してください。このオプションはアップグレードにかかる時間を短縮します。

アップグレード後

`DisableReplicationOfNextTables` レジストリー/構成設定は自動的に無効になります。

第 7 章. サーバー・オブジェクト ID

BigFix サーバーは、Fixlet、タスク、ベースライン、プロパティー、分析、アクション、ロール、カスタム・サイト、コンピューター・グループ、管理権限、サブスクリプションなど、作成するオブジェクトの一意の ID を生成します。

これらの ID は、プラットフォーム・データベースに 32 ビット・フィールドとして保管されます。次に例を示します。

- ActionID
- FixletID
- ID
- ContentID
- RoleID

ID はコンソール、Web レポートおよび WebUI インターフェースに表示され、REST API および AdminTool や PropertyIDMapper などのツールで使用されます。

現行の実装の前は、1 つのサーバーに使用可能なオブジェクト ID の最大数は 16.777.215 であったため、使用可能な DSA サーバーの最大数は 256 でした。

オブジェクト ID の上限に達するのを避けるために、Fixlet、タスク、ベースラインなどを作成する際、オブジェクト ID に使用されるビットは次のように再配置されます。

|_x__|___y__|_____________z_____________|

各部の意味は以下のとおりです。

- x = 3 ビットで、そのうち 2 ビットはカウンターに使用されます (最初のビットは、エージェントの内部処理に使用されます)
- y = サーバー ID (以前の 8 ビットではなく 5 ビット)
- z = カウンターの最初の 24 ビット (不変)

このようにして、使用可能なオブジェクト ID の数が 1.627.389.951 に増加し、結果として使用可能な DSA サーバーの数は 32 に減少しました。

このソリューションには、32 ビットのオブジェクト ID を保持して、下位互換性の問題を回避するという利点があります。

第 8 章. 収集のための HTTPS のカスタマイズ

HTTPS プロトコルをサーバー上またはエアー・ギャップ環境内で使用すると、ライセンス更新および外部サイトを収集できます

HTTPS はデフォルトのプロトコルです。

HTTPS を有効にすると、信頼する証明書のパッケージを作成または (curl Web サイトから) ダウンロードできます。curl Web サイトでは、Mozilla に付属しているものと同じ証明書が含まれている事前作成パッケージが提供しています。

BigFix サーバーは、収集中に証明書の検証を開始します。その際、提供された証明書を信頼します。

HTTPS の管理

HTTPS プロトコルを使用して外部サイトを収集するには、以下の手順を実行します。

BigFixサーバー上で:

クライアント・プロパティー `_BESGather_Use_Https` を `0`、`1`、または `2` に設定します。

プロパティーを 0 に設定すると、サーバーは URL で定義されているプロトコルを使用します。

プロパティーを 1 に設定すると、サーバーは HTTPS プロトコルのみを使用してすべてのサイトの収集を試みます。

プロパティーを 2 に設定すると、サーバーは最初に HTTPS プロトコルを使用してすべてのサイトの収集を試みます。HTTPS を使用して収集できないサイトがあった場合、サーバーは HTTP プロトコルを使用してもう一度収集を試みます。HTTPS から HTTP へのフォールバックは、`http://` から始まる URL を持つサイトにのみ適用されます。

この設定のデフォルト値は 2 です。

エアー・ギャップ環境内で:

以下のように `Airgap` コマンドを起動します。

```
Airgap
```

サーバーは、最初に HTTPS プロトコルを使用してすべてのサイトの収集を試みます。失敗した場合、サーバーは HTTP プロトコルを使用してサイトを収集します。このリダイレクトは、URL が HTTP でハードコードされている場合のみ適用されます。これが、 デフォルトの動作です。

```
Airgap -usehttps
```

サーバーは、HTTPS プロトコルのみを使用してすべてのサイトの収集を試みます。

```
Airgap -no-usehttps
```

サーバーは URL で定義されているプロトコルを使用します。

HTTPS 証明書の検証

デフォルトでは、HTTPS 接続を有効にするために使用される HTTPS 証明書は、BigFix サーバーのインストール済み環境に含まれている証明書バンドルを使用して検証されます。

Windows のデフォルト・パスは以下のとおりです。

```
C:\Program Files (x86)\BigFix Enterprise\BES Server\Reference\ca-bundle.crt
```

Linux のデフォルト・パスは以下のとおりです。

```
/opt/BESServer/Reference/ca-bundle.crt
```

HTTPS 収集の前に、信頼された証明書のカスタム・バンドルを使用して HTTPS 証明書を検証するには、以下の手順を実行します。

1. 信頼された証明書セットを作成するか、ダウンロードします (例えば、http://curl.haxx.se/ca/cacert.pem)。使用できる証明書は、以下のとおりです。
    - 「VeriSign Universal Root Certification Authority」 (サイト収集が目的の場合)
    - 「thawte Primary Root CA - G3」 (ライセンス更新の確認が目的の場合)
2. **サーバー上で:**

   HTTPS プロトコルを使用するために、クライアント・プロパティー `_BESGather_Use_Https` を `1` または `2` に設定します。`_BESGather_CACert` キー

ワードを、ダウンロード済みの信頼された証明書セットのパス (Windows システムでは `c:\TEM\certificates\custom-ca-bundle.crt`、Linux システムでは `/TEM/certificates/custom-ca-bundle.crt` など) に設定します。

エアー・ギャップ環境内で:

以下のように、オプション `-cacert <path>` を指定してエアー・ギャップ・ツールを起動します。

```
Airgap -cacert <path>
```

ここで、`<path>` は、保存済みの信頼された証明書セットのパスです。

第 9 章. DHE/ECDHE 鍵交換方式の使用

デフォルトでは、BigFix 10.0 パッチ 1 コンポーネントは、SSL 通信の反対側の BigFix コンポーネントのバージョンで DHE/ECDHE 鍵交換方式が許可されている場合に、この方式を使用します。

すべての SSL 通信で DHE/ECDHE を使用するには、すべての BigFix コンポーネントが以下のいずれかのバージョンである必要があります。

- バージョン 10.0 パッチ 1 以上
- バージョン 9.5 パッチ 16 以上

その他の考慮事項

- BigFix HTTPS サーバーは、認証と鍵交換の両方に RSA を使用します。
- BigFix 10.0 パッチ 1 により、鍵交換に一時的な Diffie-Hellman (DHE) および一時的な楕円曲線 Diffie-Hellman (ECDHE) を使用できます (認証には RSA)。
- 「一時」は、TLS 接続ごとに、永続ストレージに書き込まれることがない新しいランダムな非対称鍵が選択されることを意味します。
- TLS 接続が終了すると、鍵は安全に消去されます。
- したがって、RSA 秘密鍵が漏えいしたとしても、その鍵を使用して、記録された TLS セッションを復号化することはできません。
- BigFix コンソール・パスワード、REST API パスワード、Web レポート・パスワード、セッション・トークンなど、それらの TLS セッションで交換されたシークレットは、RSA 秘密鍵が漏えいしても漏えいしません。
- プロトコルを特定するために、Nmap ユーティリティーを使用できます。呼び出し例を次に示します。

```
nmap -p 8083 --script ssl-cert,ssl-enum-ciphers <address>
```

第 10 章. セキュア通信の構成

カスタム証明書の構成

カスタム証明書を構成する際に考慮すべき事項です。

秘密鍵および証明書のフォーマット

秘密鍵と証明書ファイルのフォーマットと構造が以下のようになっていることを確認してください。

プライベート・キーのフォーマット

PEM エンコード形式で、パスワード保護機能なし。pvk フォーマットはサポートされていません。プライベート・キー (*private.key*) が以下の文で囲まれていることを確認します。

```
-----BEGIN PRIVATE KEY-----
<<base64 string from private.key>>
-----END PRIVATE KEY-----
```

X509 証明書のフォーマット

PEM エンコード形式。中間証明書とルート証明書を別々のファイルとして受け取った場合、これらすべてを 1 つのファイルに結合する必要があります。例えば、プライマリー証明書ファイル (*certificate.crt*) および中間証明書ファイル (*ca_intermediate.crt*) がある場合、以下に示すように最初にプライマリー証明書、次に中間証明書という順序でこれらを結合する必要があります。

```
-----BEGIN CERTIFICATE-----
<<primary certificate: base64 string from certificate.crt>>
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
<<intermediate certificate: base64 string from ca_intermediate.c
rt>>
-----END CERTIFICATE-----
```

中間証明書のほかにルート証明書 (*ca_root.crt*) も受け取った場合は、これらを以下のように結合します。

```
-----BEGIN CERTIFICATE-----
<<primary certificate: base64 string from certificate.crt>>
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
<<intermediate certificate: base64 string from ca_intermediate.crt>>
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
<<root certificate: base64 string from ca_root.crt>>
-----END CERTIFICATE-----
```

単一ファイル (秘密鍵と証明書) のフォーマット

PEM エンコード形式。このファイルでは、秘密鍵と 1 次証明書の両方、または秘密鍵と証明書のチェーンが次の順序で結合され、それぞれの証明書に開始タグと終了タグがあります。

- 秘密鍵と 1 次証明書:

```
-----BEGIN CERTIFICATE-----
<<primary certificate: certificate.crt>>
-----END CERTIFICATE-----
-----BEGIN PRIVATE KEY-----
<<private key: base64 string from private.key>>
-----END PRIVATE KEY-----
```

- 秘密鍵、1 次証明書、および中間証明書:

```
-----BEGIN CERTIFICATE-----
<<primary certificate: base64 string from certificate.crt>>
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
```

```
<<intermediate certificate: base64 string from ca_intermedi
ate.crt>>
-----END CERTIFICATE-----
-----BEGIN PRIVATE KEY-----
<<private key: base64 string from private.key>>
-----END PRIVATE KEY-----
```

- 秘密鍵、1 次証明書、中間証明書、およびルート証明書:

```
-----BEGIN CERTIFICATE-----
<<primary certificate: base64 string from certificate.crt>>
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
<<intermediate certificate: base64 string from ca_intermedi
ate.crt>>
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
<<root certificate: base64 string from ca_root.crt>>
-----END CERTIFICATE-----
-----BEGIN PRIVATE KEY-----
<<private key: base64 string from private.key>>
-----END PRIVATE KEY-----
```

ファイルが DER でエンコードされているか、その他のフォーマットである場合は、OpenSSL などを使用して PEM フォーマットに変換できます。

証明書署名要求 (CSR) の作成

証明書を登録する手順。

1. 以下のような有効な構成ファイルが必要です。

```
[ req ]
default_bits = 4096
default_keyfile = keyfile.pem
```

```
distinguished_name = req_distinguished_name
attributes = req_attributes
prompt = no
output_password = bigfix

[ req_distinguished_name ]
C = US
ST = California
L = Emeryville
O = BigFix
OU = Development
CN = Common
emailAddress = admin@bigfix.com

[ req_attributes ]
challengePassword = bigfix
```

2. `Common` を Web レポート・サーバーの完全修飾ドメイン名で置き換えます。
3. 以下のコマンドを使用して証明書要求 `cert.csr` を作成します。

   ```
   openssl req -new -config "c:\mynewconfig.conf" > cert.csr
   ```

   これにより、`keyfile.pem` という名前のプライベート・キーも生成されます。
4. プライベート・キー・ファイル `keyfile.pem` からパスワードを削除し、以下のコマンドを使用して新規プライベート・キー (`nopwdkey.pem`) を生成します。

   ```
   openssl rsa -in keyfile.pem -out nopwdkey.pem
   ```

自己署名証明書の生成

自己署名証明書 (`cert.pem`) を証明書要求ファイル (`cert.csr`) から生成します。

実行する手順

1. 証明書署名要求 (`cert.csr`) を作成します。
2. 以下のコマンド (365 日間有効) を使用して、プライベート・キー・ファイル (`nopwdkey.pem`) と証明書要求ファイル (`cert.csr`) から証明書ファイル (`cert.pem`) を作成します。

```
openssl x509 -in cert.csr -out cert.pem -req -signkey nopwdkey.pem
 -days 365
```

重要: 以下のステップでは、プライベート・キー・ファイルと署名証明書ファイルを組み合わせる方法について説明します。必要に応じてそれらを個別に使用し、以下のステップをスキップできます。

注: プライベート・キーがパスワードで保護されていない場合に限り、BigFix Inventory および License Metric Tool 用に生成されたキーのペアを Web レポートにも使用できます。

3. プライベート・キー・ファイルの `nopwdkey.pem` をメモ帳などのテキスト・エディターで開きます。
4. その内容をコピーして、以下の例のように `cert.pem` の証明書の下に貼り付けます。

```
-----BEGIN CERTIFICATE-----
...
-----END CERTIFICATE-----
-----BEGIN RSA PRIVATE KEY-----
...
-----END RSA PRIVATE KEY-----
```

ここで、`...` はテキストを表しています。

5. 「Web レポートでの HTTPS のカスタマイズ (（ページ）　　)」の説明に従い、Web レポート・サーバー上の `cert.pem` で、証明書パスのレジストリー設定を確認します。

認証局に対する証明書の要求

ブラウザーが暗黙的に信頼する証明書を使用して HTTPS Web レポートを暗号化するには、以下のように Verisign などの信頼できる認証局 (CA) に対して署名証明書を要求します。

1. 証明書署名要求 (CSR) を作成します。 (（ページ） *87)*
2. `.csr` ファイルを認証局 (CA) に転送します。認証局が、ご使用のサーバーに対して署名証明書 (ブラウザーが信頼する証明書) を発行します。認証局には、トラスト・チェーン全体を含む `.pem` ファイルとして証明書を発行するように依頼してください。

   重要: 以下のステップでは、秘密鍵ファイルと署名証明書ファイルを組み合わせる方法について説明します。必要に応じてそれらを個別に使用し、以下のステップをスキップできます。

   注: プライベート・キーがパスワードで保護されていない場合に限り、BigFix Inventory および License Metric Tool 用に生成されたキーのペアを Web レポートにも使用できます。
3. 署名証明書ファイルを受け取ったら、それを Microsoft のデフォルトの証明書処理機能にはインポートしないでください。
4. パスワード (`nopwdkey.pem`) を削除したプライベート・キー・ファイルを開いて、その内容をクリップボードにコピーします。
5. 署名証明書ファイルを Notepad++ などのテキスト・エディターで開きます。
6. ステップ 4 でコピーした内容を署名証明書ファイルに追加します。これは、最終的な内容の例です。

```
-----BEGIN CERTIFICATE-----
...
-----END CERTIFICATE-----
-----BEGIN RSA PRIVATE KEY-----
...
-----END RSA PRIVATE KEY-----
```

ここで、`...` はテキストを表しています。

7. パブリック証明書とプライベート・キーを含む変更後の `.pem` ファイルを保存します。
8. このファイルをご使用のサーバーに保管して、Web レポートをセットアップする際に参照します。

Web レポートでの HTTPS のカスタマイズ

Web レポートで HTTPS をカスタマイズする方法について詳しくは、『Web レポートでの HTTPS のカスタマイズ (（ページ）　　)』を参照してください。

REST API での HTTPS のカスタマイズ

BigFix ルート・サーバーは、インストール時にデフォルトで HTTPS を使用するように構成され、インストール時に独自の証明書を作成します。これを変更するには、HTTPS を手動で設定する必要があります。

ファースト・ステップ

認証局からの信頼できる SSL セキュリティー証明書およびキーがある場合、BigFix ルート・サーバーでこの証明書とキーを使用してトラステッド接続を使用するように設定できます。自己署名証明書も使用可能です。

信頼された SSL 証明書がある場合、BigFix ルート・サーバーを実行しているコンピューターに.pvkファイル (存在する場合) と .pem ファイルをコピーします。

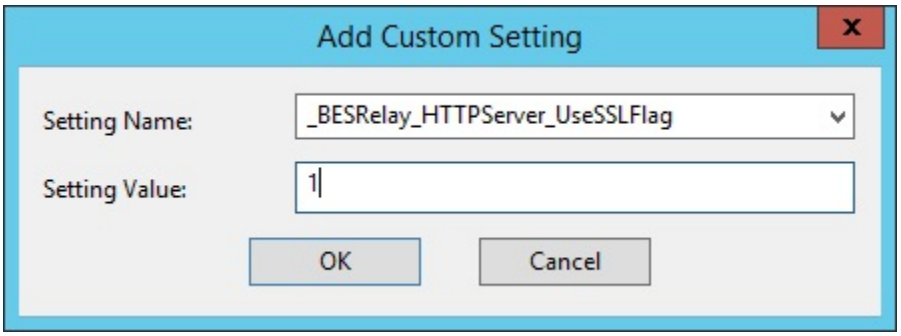
以下のセクションでは、これらのマクロ・ステップを実装する方法を示します。

- セキュア通信を使用することを指定します。
- SSL 証明書ファイルおよび秘密鍵ファイルを配置する場所を指定します。
- 関連サービスを再起動します。

以下のセクションで説明する構成を完了すると、Rest API と BigFix コンソールからの接続にこの信頼できる証明書が使用されます。

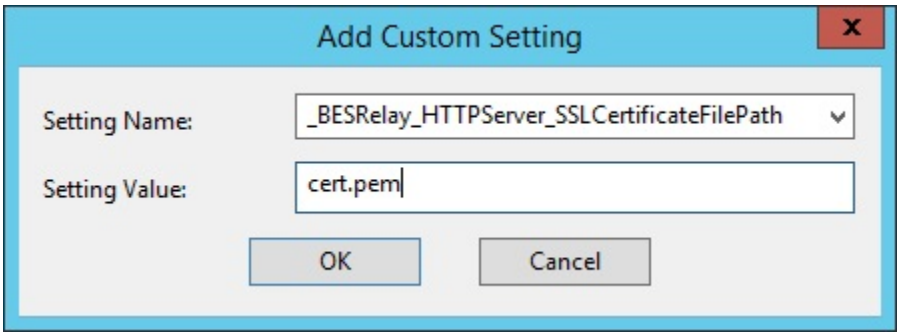
BigFix コンソールを使用した HTTPS のカスタマイズ

1. BigFix コンソールから**「コンピューター」**タブを選択します。
2. Rest API を実行するコンピューター (通常はサーバー) を選択し、**「編集」**メニューから**「コンピューター設定の編集」**を選択します。
3. **_BESRelay_HTTPServer_UseSSLFlag** という設定を探します。存在する場合は、新たに作成せずに、その値を編集して `1` にし、HTTPS を有効にします。存在しない場合は、以下のように追加します。



4. 秘密鍵ファイルと証明書ファイルを組み合わせた場合は、このステップをスキップして、**_BESRelay_HTTPServer_SSLCertificateFilePath** のみを設定してください。

   _BESRelay_HTTPServer_SSLPrivateKeyFilePath という設定を探します。この設定が存在する場合は、2 つ目を作成せずに、その値を、秘密鍵 (サーバーの秘密鍵が格納されている .pvk ファイル) の絶対パス名に変更します。この秘密鍵にパスワードを設定することはできません。この設定が存在しない場合は、追加します。
5. **_BESRelay_HTTPServer_SSLCertificateFilePath** という設定を探します。この設定が存在する場合は、2 つ目を作成せずに、その値を .pem ファイルの絶対パス名に変更します。このファイルには、サーバーの証明書と秘密鍵の両方が格納される場合もあれば、証明書だけが格納される場合もあります。この設定が存在しない場合は、以下のように追加します。



`.pem` ファイルが標準の OpenSSL PKCS7 .pem ファイル形式になっていることを確認してください。

証明書は、サーバーから接続クライアントに渡されます。これらのクライアントは、証明書の情報を示すダイアログをユーザーに対して表示します。証明書が、接続するクライアントの信頼性要件をすべて満たす場合、クライアントはユーザーが介入しなくても接続します。証明書がクライアントの信頼性要件を満たしていない場合、接続を続行するかどうかを尋ねるダイアログがユーザーに対して表示されます。ユーザーは、証明書に関する情報にアクセスすることができます。信頼された証明書とは、Verisign などの信頼できる認証局によって署名され、正しいホスト名を持ち、期限切れになっていない証明書のことです。

6. TLS12 を要求するために、**_BESRelay_HTTPServer_RequireTLS12** を探します。存在する場合は、新たに作成せずに、その値を編集して `1` にします。

   注: REST API コンポーネントは、ローカルの設定やマストヘッドの設定に関係なく、BigFix サーバーと通信する際に必ず TLS 1.2 を使用します。

7. **BES ルート・サーバー**・サービスを再始動します。
   - Windows の場合は、**「サービス」**を開き、**「BES Root Server」**を選択し、**「操作」**メニューで**「再起動」**をクリックします。
   - Linux の場合は、プロンプトで次を実行します。`service besserver restart` または `/etc/init.d/besserver restart`。
8. BES ルート・サーバーと Web レポートの間の接続を復元するには、Web レポートから、証明書が変更されたデータ・ソースのデータ・ソース設定を以下のように編集します。

a. **「管理」>「データ・ソース設定」>「編集」**を選択します。
b. 該当するフィールドにパスワードを入力し、フォームを送信して証明書を交換し、要求の警告を承諾します。

注: これらの設定は、キー **HKLM/Software/WoW6432Node/BigFix/EnterpriseClient/Settings/Client** の下にあるレジストリーに保管されます。

HTTPS の手動カスタマイズ

認証局から信頼できる SSL セキュリティーと鍵 (`.pem` ファイル) が発行されている場合は、REST API を実行するコンピューター (通常はサーバー) を構成して、信頼できる接続をカスタマイズすることができます。

Windows システム

Windows システムで HTTPS を手動でカスタマイズするには、以下の手順を実行します。

1. **regedit** を実行し、`HKEY_LOCAL_MACHINE\Software\Wow6432Node\BigFix\EnterpriseClient\Settings\Client` を表示します。

   HTTPS フラグ用、および SSL 証明書のロケーション用のサブキーを追加または変更する必要があります。
2. **クライアント**のサブキー `_BESRelay_HTTPServer_UseSSLFlag` を作成します (まだ存在しない場合)。「value」という名前のストリング値 (reg_sz) を追加し、それを 1 に設定して HTTPS を有効にします。
3. **重要: セキュリティー上の理由から、ファイル・アプリケーションでは添付ファイル配置が使用されます。秘密鍵ファイルと証明書ファイルを組み合わせた場合は、ステップ 4 に進みます。**

   クライアントのサブキー `_BESRelay_HTTPServer_SSLPrivateKeyFilePath` を作成します (まだ存在しない場合)。「value」という名前のストリング値 (reg_sz) をキーに追加し、それを秘密鍵 (サーバーの秘密鍵を含む .pvk ファイル) の絶対パス名に設定します。
4. クライアントのサブキー `_BESRelay_HTTPServer_SSLCertificateFilePath` を作成します (まだ存在しない場合)。「value」という名前のストリング値 (reg_sz) を追加し、それを SSL 証明書 (cert.pem) の絶対パス名に設定します。

5. **TLS 1.2 を要求する場合:** クライアントのサブキー `_BESRelay_HTTPServer_RequireTLS12` を作成します (まだ存在しない場合)。「value」という名前のストリング値 (reg_sz) を追加し、それを 1 に設定して TLS 1.2 を有効にします。
6. `BES Root Server` サービスを再開します。
7. BES ルート・サーバーと Web レポートの間の接続を復元するには、Web レポートから、証明書が変更されたデータ・ソースのデータ・ソース設定を以下のように編集します。
    a. **「管理」>「データ・ソース設定」>「編集」**を選択します。
    b. 該当するフィールドにパスワードを入力し、フォームを送信して証明書を交換し、要求の警告を承諾します。

Linux システムの場合

Linuxシステムで HTTPS を手動でカスタマイズするには、以下のステップを実行します。

ファイル `cert.pem` と `pvtkey.pvk` (ファイルがある場合) をファイル・システムの保護領域に保存します。この領域では、BigFix besserver プロセスによってアクセスできます 。例えば、 `/etc/opt/BESServer/` です。

以下の項目を追加して、`/var/opt/BESServer/besserver.config` ファイルを編集します。

重要: セキュリティー上の理由から、ファイル・アプリケーションでは添付ファイル配置が使用されます。秘密鍵ファイルと証明書ファイルを組み合わせた場合は、これらの設定をスキップします。

```
[Software\BigFix\EnterpriseClient\Settings\Client\_BESRelay_HTTPServer_SSLP
rivateKeyFilePath]
value = /etc/opt/BESServer/pvtkey.pvk
```

```
[Software\BigFix\EnterpriseClient\Settings\Client\_BESRelay_HTTPServer_SSLC
ertificateFilePath]
value = /etc/opt/BESServer/cert.pem
```

HTTPS を有効化するには、以下をカスタマイズします。

```
[Software\BigFix\EnterpriseClient\Settings\Client\_BESRelay_HTTPServer_UseSSLFlag]
value = 1
```

TLS 1.2 を要求する場合:

```
[Software\BigFix\EnterpriseClient\Settings\Client\_BESRelay_HTTPServer_RequireTLS12]
value = 1
```

BigFix ルート・サーバーを停止してから再始動します。

BES ルート・サーバーと Web レポートの間の接続を復元するには、以下の手順を行います。

Web レポートから、証明書が変更されたデータ・ソースのデータ・ソース設定を以下のように編集します。

1. **「管理」>「データ・ソース設定」>「編集」**を選択します。
2. 該当するフィールドにパスワードを入力し、フォームを送信して証明書を交換し、要求の警告を承諾します。

第 11 章. リアルタイム AV 除外

BigFix アーキテクチャーのコンソール、サーバー、およびリレー・コンポーネントは、大量のファイル操作を実行します。このアクティビティーは、これらの BigFix アーキテクチャー・コンポーネントが提供する機能の重要な部分です。

アンチウィルスまたはヒューリスティック・タイプのアプリケーション (HIPS など) によってファイル操作が中断または「shim 処理」されると、これらのコンポーネントのパフォーマンスが大きな影響を受けます。これにより、エラーが発生し不安定になることがあります。また、BigFix クライアントはマシンを継続的に評価し、そのために API、レジストリー、およびファイルの操作が大量に作成されます。クライアントも同じ問題による悪影響を受け、その結果、コンテンツの評価時間が大幅に長くなる可能性があります。

この問題に対処するには、以下のディレクトリーおよびプロセスを除外するようにアンチウィルスおよびヒューリスティック・アプリケーション (HIPS など) を構成します。以下の仕様は、リアルタイム・スキャンおよびヒューリスティックのフォルダー・パスおよびプロセスの除外に関連していることに注意してください。ただし、セキュリティーの観点から引き続きスケジュール済みスキャンを構成して有効にすることをお勧めします。

重要な注意事項

以下は、BigFix プラットフォーム・コア・コンポーネントにのみ適用され、BigFix インベントリー、ILMT、OSD などのソリューションを除外します (AV 例外に関して独自のガイダンスを持つ場合があります)。また、これはデフォルトのインストール・パスを使用していることを前提としています。使用していない場合は、ご使用の環境の構成に応じて適切に調整する必要があります。

AV 除外について詳しくは、『Windows での AV 除外 (（ページ） 98)』と『Linux での AV 除外 (（ページ） 100)』を参照してください。

この除外ルールの設定方法について詳しくは、ウィルス・スキャナーの説明を参照してください。

詳しくは、技術情報「BigFix クライアントおよび BigFix Inventory スキャナーを除外するようにウィルス・スキャナーを構成する」を参照してください。

Windows での AV 除外

BigFix プラットフォームのコア・コンポーネントの Windows OS に AV 除外を適用する方法を説明します。

注: *<installation path>* のデフォルト値は `C:\Program Files (x86)\BigFix Enterprise` です。

- **BigFix サーバーの場合**

  以下のフォルダーとサブフォルダーのパスを除外する必要があります。

  <installation path>\BES Server*

  C:\Windows\Temp\tem*.tmp*

  さらに、以下のプロセスも除外する必要があります。

  <installation path>\BES Server\BESRootServer.exe

  <installation path>\BES Server\BESWebReportsServer.exe

  <installation path>\BES Server\BESAdmin.exe

  <installation path>\BES Server\FillDB.exe

  <installation path>\BES Server\GatherDB.exe

- **BigFix リレーの場合**

  以下のフォルダーとサブフォルダーのパスを除外する必要があります。

  <installation path>\BES Relay*

  さらに、以下のプロセスも除外する必要があります。

  <installation path>\BES Relay\BESRelay.exe

- **BigFix クライアントの場合**

  以下のフォルダーとサブフォルダーのパスを除外する必要があります。

  <installation path>\BES Client*

  さらに、以下のプロセスも除外する必要があります。

  <installation path>\BES Client\BESClient.exe

<installation path>\BES Client\BESClientUI.exe

オプションで、BES クライアント・ディレクトリー内で QNA コンポーネントを利用する場合には、以下のプロセスも除外する必要があります。

<installation path>\BES Client\qna.exe

- **BigFix コンソールの場合**

  以下のフォルダーとサブフォルダーのパスを除外する必要があります。コンソールのこのプライマリー AV 例外は、コンソール・キャッシュ・ディレクトリーに関連しています。このディレクトリーは、デフォルトではユーザーのプロファイル・パス内にあります。例:

  %LOCALAPPDATA%\BigFix*

  ユーザー BigFix コンソール・キャッシュの場所は、レジストリー設定でも構成できます (これにより、一部の AV およびヒューリスティックス製品では AV 除外の適用が容易になります)。この構成について詳しくは、こちらを参照してください:BigFix コンソール・キャッシュの場所の変更

  また、以下のプロセスとファイルも除外する必要があります。

  <installation path>\BES Console\BESConsole.exe

  %LOCALAPPDATA%\Temp\tem*.tmp

  オプションで、BigFix コンソール・ディレクトリー内で QNA コンポーネントを利用する場合には、以下のディレクトリーも除外する必要があります。

  <installation path>\BES Console\QNA*

  さらに、以下のプロセスもあります。

  <installation path>\BES Console\QNA\FixletDebugger.exe

- **BigFix WebUI サーバーの場合**

  以下のフォルダーとサブフォルダーのパスを除外する必要があります。

  <installation path>\BES WebUI*

  さらに、以下のプロセスを除外する必要があります。

  <installation path>\BES WebUI\WebUIService.exe

<installation path>\BES WebUI\WebUI\node.exe

- **BigFix プラグイン・ポータルの場合**

  以下のフォルダーとサブフォルダーのパスを除外する必要があります。

  <installation path>\BES Plugin Portal*

  さらに、以下のプロセスを除外する必要があります。

  <installation path>\BES Plugin Portal\BESPluginPortal.exe

Linux での AV 除外

BigFix プラットフォームのコア・コンポーネントの Linux OS に AV 除外を適用する方法を説明します。

- **BigFix サーバーの場合**

  以下のフォルダーとサブフォルダーのパスを除外する必要があります。

  /opt/BESServer/

  /opt/BESWebReportsServer/

  /var/opt/BESServer/

  /var/opt/BESInstallers/

  /var/opt/BESWebReportsServer/

  /var/log/

  /etc/opt/BESServer/

  /etc/opt/BESWebReportsServer/

  /etc/init.d/

  /usr/lib/systemd/system

  さらに、以下のプロセスも除外する必要があります。

  /opt/BESServer/bin/BESFillDB

  /opt/BESServer/bin/BESGatherDB

/opt/BESServer/bin/BESRootServer

/opt/BESServer/bin/BESAdmin.sh

/opt/BESServer/bin/BESAdmin

/opt/BESServer/bin/iem

/opt/BESServer/bin/Airgap

/opt/BESServer/bin/Airgap.sh

/opt/BESWebReportsServer/bin/WebReportsInitDB.sh

/opt/BESWebReportsServer/bin/BESWebReportsServer

- **BigFix リレーの場合**

  以下のフォルダーとサブフォルダーのパスを除外する必要があります。

  /opt/BESRelay/

  /var/opt/BESRelay/

  /var/log/

  /etc/init.d/

  /usr/lib/systemd/system

  さらに、以下のプロセスも除外する必要があります。

  /opt/BESRelay/bin/BESRelay

- **BigFix クライアントの場合**

  以下のフォルダーとサブフォルダーのパスを除外する必要があります。

  /opt/BESClient/

  /var/opt/BESClient/

  /var/opt/BESCommon/

  /etc/opt/BESClient/

  /etc/init.d/

  /usr/lib/systemd/system

さらに、以下のプロセスも除外する必要があります。

/opt/BESClient/bin/BESClient

/opt/BESClient/bin/qna

/opt/BESClient/bin/XBESClientUI

/opt/BESClient/bin/XOpenUI

/opt/BESClient/bin/xqna

- **BigFix WebUI サーバーの場合**

  以下のフォルダーとサブフォルダーのパスを除外する必要があります。

  /opt/BESWebUI/

  /var/opt/BESWebUI/

  /etc/init.d/

  /usr/lib/systemd/system

  さらに、以下のプロセスも除外する必要があります。

  /opt/BESWebUI/bin/BESWebUI

  /var/opt/BESWebUI/node

- **BigFix ポータルの場合**

  以下のフォルダーとサブフォルダーのパスを除外する必要があります。

  /opt/BESPluginPortal/

  /var/opt/BESPluginPortal/

  /var/log/

  /etc/init.d/

  /usr/lib/systemd/system

  さらに、以下のプロセスも除外する必要があります。

  /opt/BESPluginPortal/bin/BESPluginPortal

第 12 章. エアー・ギャップ環境でのファイルのダウンロード

エアー・ギャップ環境で、メインの BigFix サーバーにファイルをダウンロードおよび転送するには、Airgap ユーティリティーおよび BES Download Cacher ユーティリティーを使用します。

概要

エアー・ギャップ環境 (パブリック・インターネットや非セキュア・ローカル・エリア・ネットワークなどのセキュアではないネットワークからセキュアなネットワークが物理的に分離され、エアー・ギャップの反対側にあるコンピューター同士は通信できない環境) の場合、Airgap ユーティリティーと BES Download Cacher ユーティリティーを使用して、メインの BigFix サーバーにファイルをダウンロードおよび転送することができます。

注: Airgap ユーティリティーでは、メインの BigFix サーバーとは別個にクライアントがエアー・ギャップされている構成はサポートされません。メインの BigFix サーバーからネットワーク全体にわたってクライアントをまとめて機能させるには、クライアントがメインの BigFix サーバーと一緒にエアー・ギャップされている必要があります。

BigFix バージョン 9.5.5 以降、エアー・ギャップ環境で 2 つの異なるモードを使用できます。バージョン 9.5.5 より前に既に使用可能であった「抽出使用」モードと新しい「非抽出使用」モードを使用できます。

非抽出使用の概要

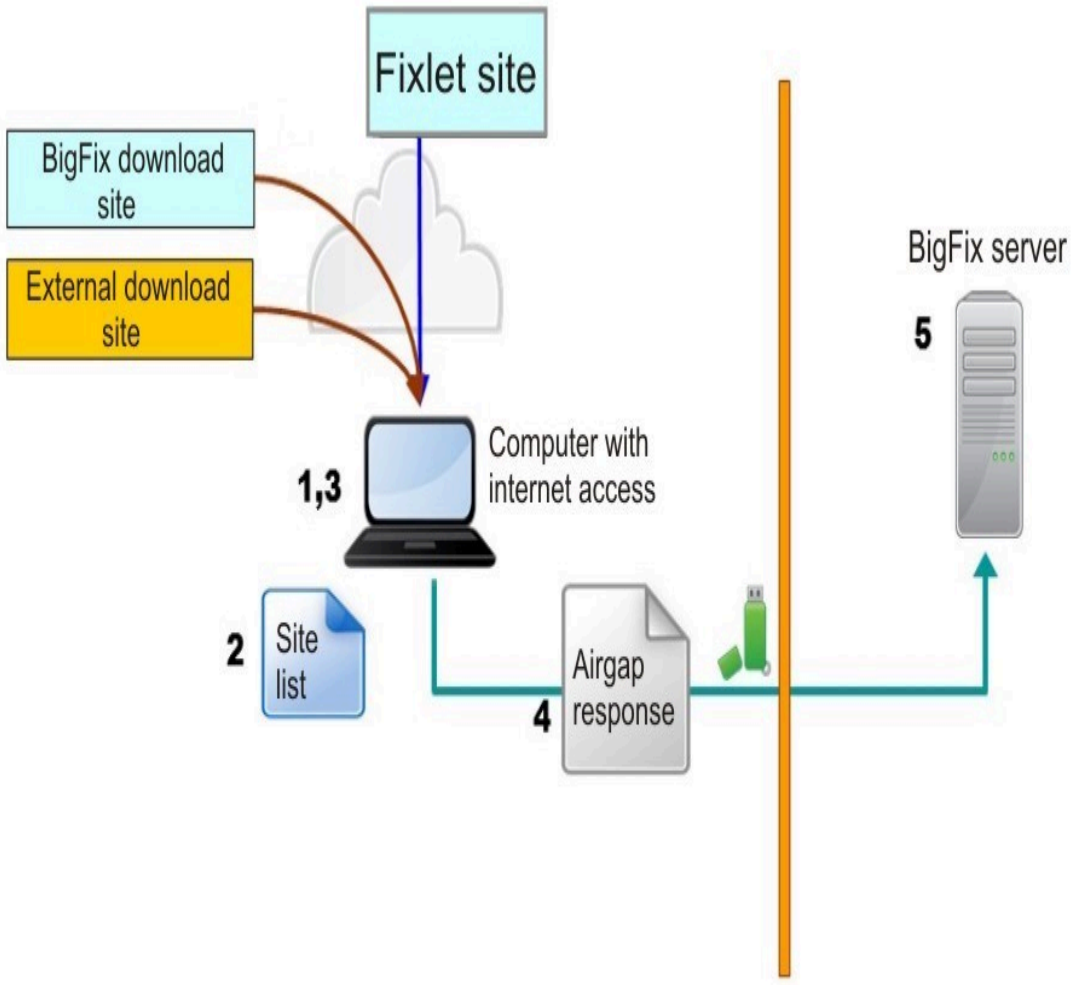
「非抽出使用」モードは、BigFix バージョン 9.5.5 以降でのみ使用できます。

場所によっては、ルールによってセキュア・ネットワーク内の情報を抽出してインターネットなどの外部ネットワークに移動することが禁止されているため、エアー・ギャップが BigFix サーバーから情報を抽出せずに機能しなければならないことがあります。このような要件を満たすために、エアー・ギャップ・ツールは、エアー・ギャップ要求を作成せずに機能できるようになりました。

エアー・ギャップ・ツールは、以下の 3 つの方法で使用できます。

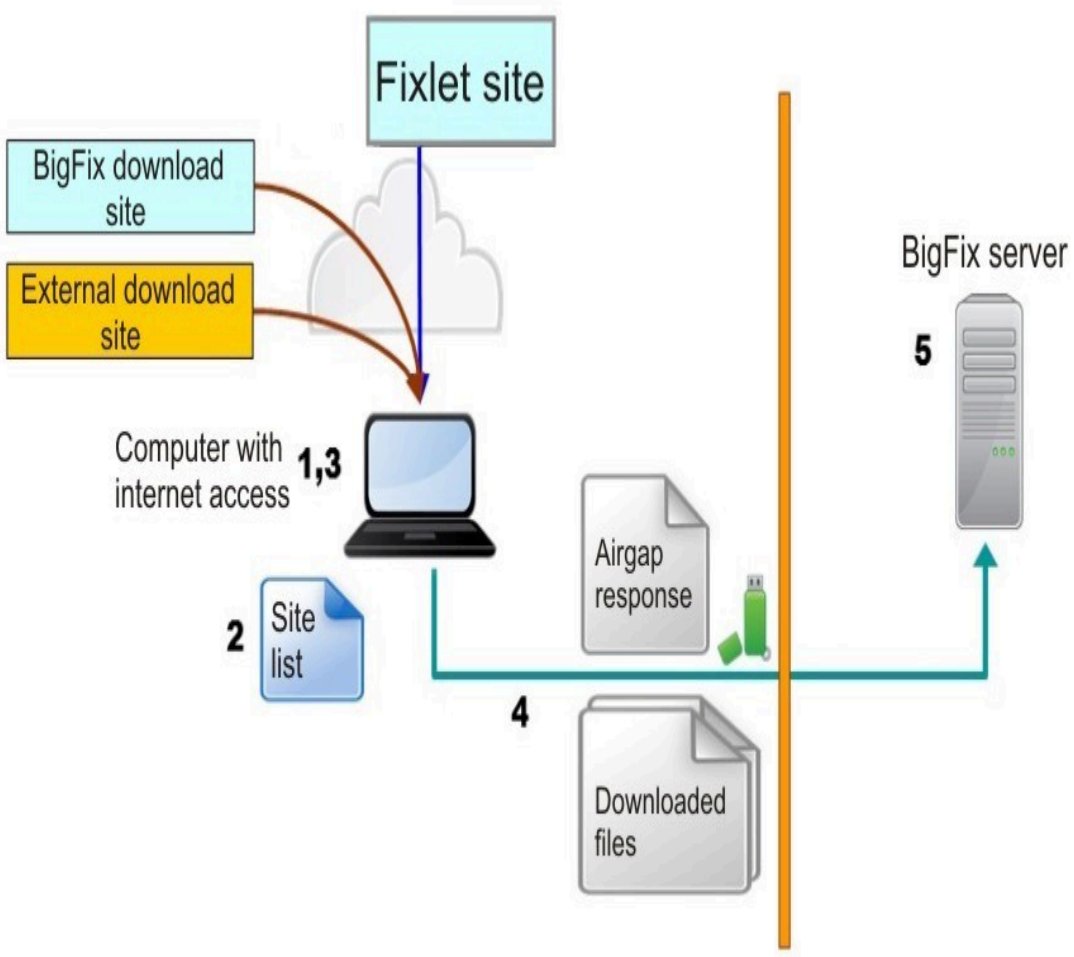
サイト・コンテンツの収集

1. インターネットにアクセスできるコンピューターでエアー・ギャップ・ツールを実行して、ライセンス情報を収集し、ライセンス交付を受けているサイトに関連した情報が入ったサイト・リスト・ファイルを作成します。
2. サイト・リスト・ファイルを編集して、フラグを変更し、コンテンツの収集元のサイトを指定します。
3. インターネット側のコンピューターでエアー・ギャップ・ツールを実行して、サイト・リスト・ファイルで指定されているようにライセンス情報およびサイト・コンテンツを収集し、エアー・ギャップ応答に入れます。
4. エアー・ギャップ応答を BigFix サーバーに移動します。
5. BigFix サーバーでエアー・ギャップ・ツールを実行して、エアー・ギャップ応答を BigFix サーバーにロードします。



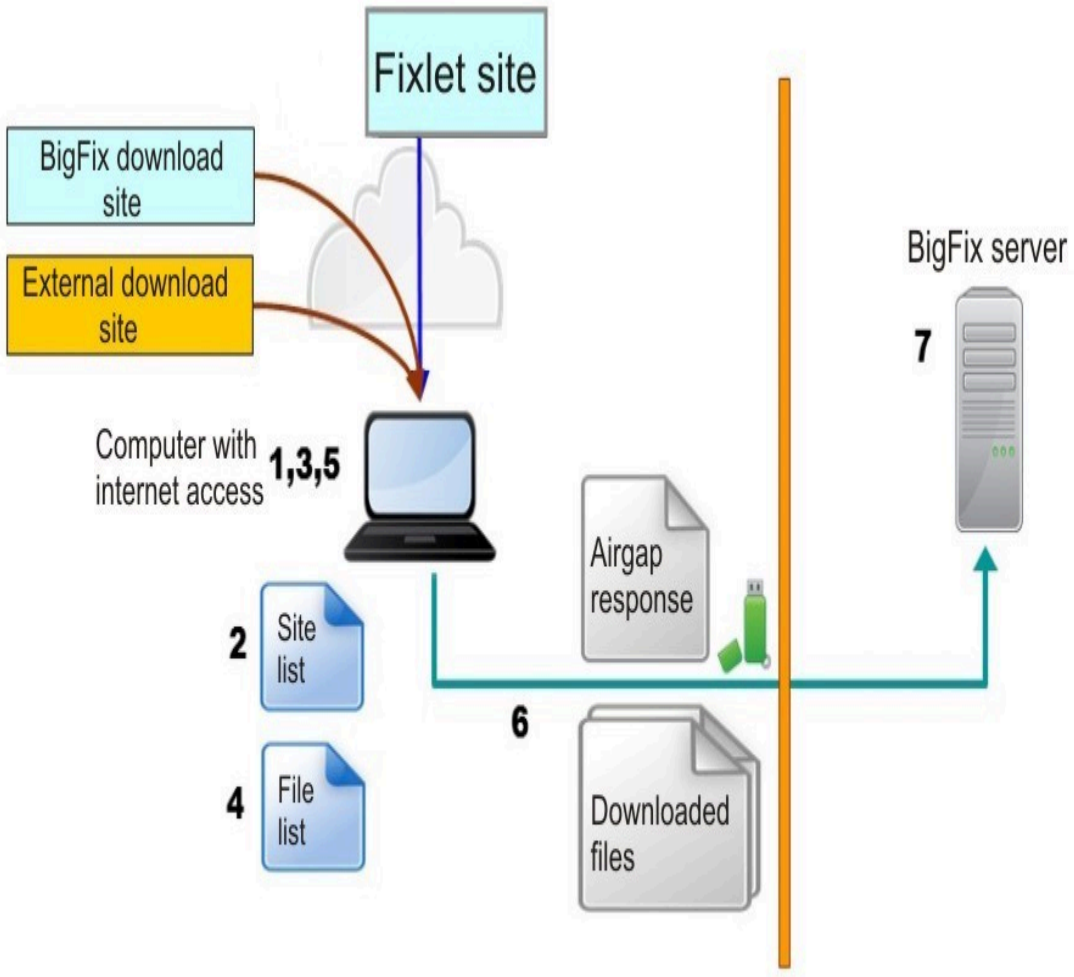
サイト・コンテンツの収集およびファイルのダウンロード

1. インターネットにアクセスできるコンピューターでエアー・ギャップ・ツールを実行して、ライセンス情報を収集し、ライセンス交付を受けているサイトに関連した情報が入ったサイト・リスト・ファイルを作成します。
2. サイト・リスト・ファイルを編集して、フラグを変更し、コンテンツの収集元のサイト、および参照されているファイルのダウンロード元のサイトを指定します。
3. インターネット側のコンピューターでエアー・ギャップ・ツールを実行して、サイト・リスト・ファイルで指定されているようにライセンス情報およびサイト・コンテンツを収集し、エアー・ギャップ応答に入れてから、Fixlet によって参照されるファイルをダウンロードします。
4. エアー・ギャップ応答およびダウンロードしたファイルを BigFix サーバーに移動します。
5. BigFix サーバーでエアー・ギャップ・ツールを実行して、エアー・ギャップ応答を BigFix サーバーにロードし、ダウンロードしたファイルを BigFix サーバーのキャッシュ・フォルダーにコピーします。



サイト・コンテンツの収集およびファイルの選択的なダウンロード

1. インターネットにアクセスできるコンピューターでエアー・ギャップ・ツールを実行して、ライセンス情報を収集し、ライセンス交付を受けているサイトに関連した情報が入ったサイト・リスト・ファイルを作成します。
2. サイト・リスト・ファイルを編集して、フラグを変更し、コンテンツの収集元のサイト、および参照されているファイルのダウンロード元のサイトを指定します。
3. インターネット側のコンピューターでエアー・ギャップ・ツールを実行して、サイト・リスト・ファイルで指定されているようにライセンス情報およびサイト・コンテンツを収集し、エアー・ギャップ応答に入れてから、参照されているファイルに関する情報が入ったファイル・リスト・ファイルを作成します。
4. ファイル・リスト・ファイルを編集して、ダウンロードするファイルを指定します。
5. インターネット側のコンピューターでエアー・ギャップ・ツールを実行して、ファイル・リスト・ファイルで指定されているようにファイルをダウンロードします。
6. エアー・ギャップ応答およびダウンロードしたファイルを BigFix サーバーに移動します。
7. BigFix サーバーでエアー・ギャップ・ツールを実行して、エアー・ギャップ応答を BigFix サーバーにロードし、ダウンロードしたファイルを BigFix サーバーのキャッシュ・フォルダーにコピーします。

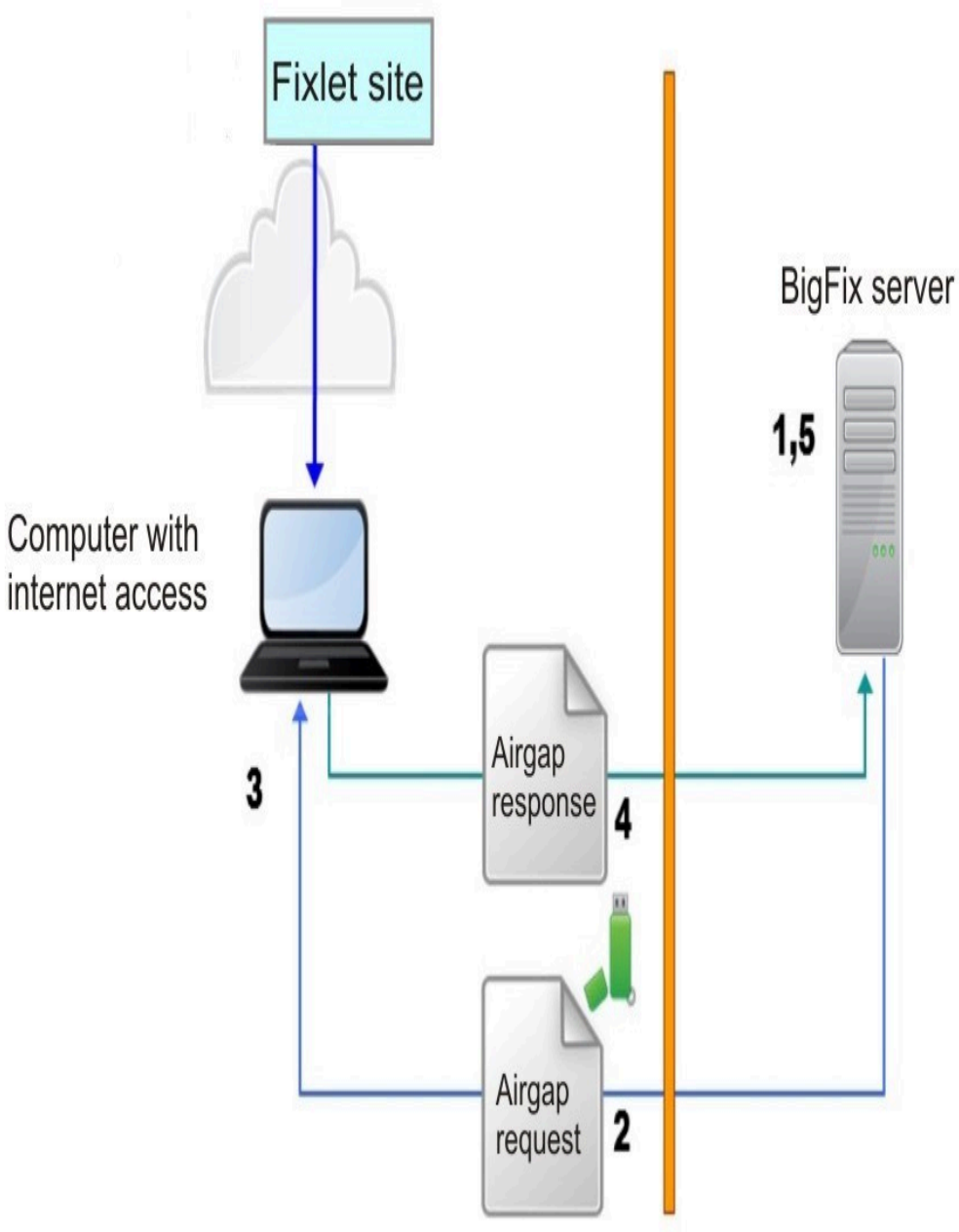

Fixlet site
BigFix download site
External download site
BigFix server
7
Computer with internet access
1,3,5
Airgap response
2
Site list
6
4
File list
Downloaded files

抽出使用の概要

このモードでは、エアー・ギャップ・ツールは BigFix サーバーから情報を抽出します。

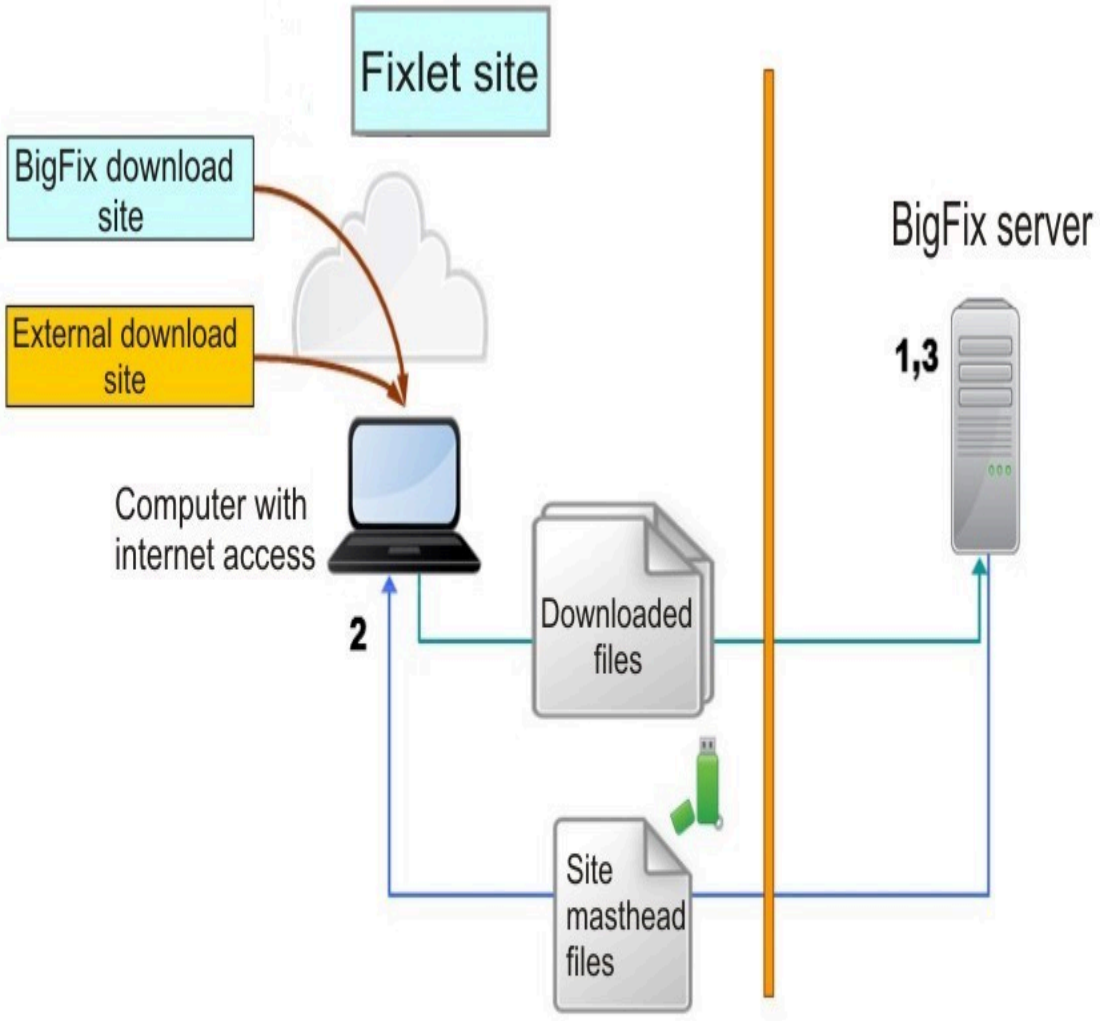
以下のステップを実行することにより、BigFix サーバーから開始してエアー・ギャップ・ツールを実行します。

1. エアー・ギャップ・ツールを BigFix サーバーで実行して、エアー・ギャップ要求を作成します。
2. エアー・ギャップ要求をインターネット側のコンピューターに移動します。
3. インターネット側のコンピューターでエアー・ギャップ・ツールを実行して、ライセンス情報およびサイト・コンテンツを収集し、エアー・ギャップ応答に入れます。
4. エアー・ギャップ応答を BigFix サーバーに移動します。
5. BigFix サーバーでエアー・ギャップ・ツールを実行して、エアー・ギャップ応答を BigFix サーバーにロードします。


Fixlet site
BigFix server
1,5
Computer with
internet access
3
Airgap
response
4
Airgap
request
2

このモードでは、エアー・ギャップは、ファイルではなく、サイトのコンテンツを収集します。パッチ・モジュールなど、Fixlet によって参照されるファイルをダウンロードするには、以下のステップを実行して BES Download Cacher ユーティリティーを実行します。

1. ファイルをダウンロードするサイトのサイト・マストヘッド・ファイルを見つけて、インターネットにアクセスできるコンピューターにサイト・マストヘッド・ファイルをコピーします。
2. インターネット側のコンピューターで、各サイト・マストヘッド・ファイルに対して BES Download Cacher ユーティリティーを実行し、サイト・マストヘッド・ファイルが表すサイトから参照されるファイルをダウンロードします。
3. ダウンロードしたファイルを BigFix サーバーのキャッシュ・フォルダーに移動します。


Fixlet site
BigFix download site
External download site
BigFix server
1,3
Computer with internet access
2
Downloaded files
Site masthead files

要件

エアー・ギャップ環境 (パブリック・インターネットや非セキュア・ローカル・エリア・ネットワークなどのセキュアではないネットワークからセキュアなネットワークが物理的に分離され、エアー・ギャップの反対側にあるコンピューター同士は通信できない環境) に BigFix サーバーがインストールされている場合、エアー・ギャップ・ツールを使用して Fixlet サイト・コンテンツをダウンロードし、Fixlet アクション・スクリプトで参照されているファイルをダウンロードするには、パブリック・インターネットにアクセスできるワークステーションが必要です。

このワークステーションは、BigFix サーバーにすることも、BigFix リレーにすることもできません。

エアー・ギャップ・ツールはプラットフォーム依存ですが、`AirgapRequest.xml` ファイル (抽出使用のみ) および `AirgapResponse` ファイルはそうではありません。パブリック・インターネットにアクセスできるワークステーションで、さまざまなオペレーティング・システムを BigFix サーバーに使用できます。

収集されるサイトによっては、`AirgapResponse` ファイルのサイズが 4 GB を超える可能性があります。ワークステーションには、エアー・ギャップ・ツール、`AirgapResponse` ファイル、およびダウンロードするファイルを保存するのに十分な空きディスク・スペースが必要です。

Windows コンピューターでエアー・ギャップ・ツールを実行するには、以下のライブラリーおよびファイルがインストールされている必要があります。

```
BESAirgapTool.exe
libBEScrypto.dll
libBEScryptoFIPS.dll
msvcm90.dll
msvcp90.dll
msvcr90.dll
Microsoft.VC90.CRT.manifest
ca-bundle.crt
```

上記のすべてのファイルは、Utilities ページから圧縮ファイル (Airgap Tool) をダウンロードすることで入手できます。

Linux コンピューターでエアー・ギャップ・ツールを実行するには、以下のファイルがインストールされている必要があります。

```
Airgap
Airgap.sh
libBEScrypto.so
libBEScryptoFIPS.so
ca-bundle.crt
```

パブリック・インターネットにアクセスできる Linux コンピューターに DB2 がインストールされていない場合、エアー・ギャップ・ツールを実行するには、`db2setup` コマンドを使用して HCL Data Server Client または HCL Data Server Runtime Client をインストールしておく必要があります。DB2 インスタンスはユーザー `db2inst1` で作成される必要があります。

エアー・ギャップ・ツールの使用

非抽出使用

エアー・ギャップ・コマンド・ライン・インターフェースは、BigFix サーバーにアクセスする必要なしにサイト情報を収集でき、オプションでダウンロード・キャッシャーを経由せずにファイルをダウンロードできます。

非抽出使用では、エアー・ギャップ・ツールは、認証する必要のない Windows などのダウンロード・サイトから Fixlet で指定されたファイルをダウンロードできます。ユーザー ID およびパスワードで認証する必要があるサイトからファイルをダウンロードする必要がある場合、または Fixlet のプリフェッチ・コマンドやダウンロード・コマンドで指定されていないファイルをダウンロードする必要がある場合 (AIX、CentOS、HP-UX、RedHat、Solaris、または SUSE のパッチ・モジュールの場合など) は、ダウンロード・キャッシャーを使用する必要があります。

以下の手順を行うための前提条件として、エアー・ギャップ・ツールの実行に必要なファイルが存在することを確認してください。

Windows では

適切な Airgap ツールのバージョンを「サポート」ページからダウンロードできます。

Linux の場合

BigFix バージョン 10.0.2 以降では、`unixODBC.x86_64` という名前のパッケージをインストールする必要があります。BigFix サーバーにインストールされているのと同じパッケージ・バージョンを、エアー・ギャップ環境の非抽出手順を実行するインターネットに接続したワークステーションにもインストールする必要があります。

BigFix サーバー・コンピューターにアクセスし、`/opt/BESServer/bin` フォルダーを開き、次のコマンドを実行します。

```
# cd /opt/BESServer/bin
# ./Airgap.sh -remotedir directory
```

ここで、*directory* は任意のフォルダーです。

上記のコマンドで生成された出力が入っているディレクトリーに移動し、`airgap.tar` というファイルを見つけ、これを解凍します。このディレクトリーから `AirgapRequest.xml` ファイルを削除し、他のファイルをすべてポータブル・ドライブにコピーします。

BigFix サーバーにアクセスせずにサイト情報を収集するには、以下のステップを実行します。

1. サイト・リストを作成します

パブリック・インターネットにアクセスできるワークステーションで、ライセンスのシリアル番号、ライセンスの登録に使用した E メール・アドレス、およびツールによってライセンスの対象のサイトがリストされたファイルの名前を指定して、ツールを実行します。エアー・ギャップ・ツールが配置されているフォルダーに対する書き込み権限が必要です。次のコマンドを入力します。

Windows オペレーティング・システムの場合:

```
BESAirgapTool.exe -serial serial_number -email
mail_address -createSiteList site_list_filename
 [-proxy
[user:password@]hostname:port] [-usehttps]
[-cacert crt_filename] [-othersites site_foldername]
 [-timeout timeout_seconds]
```

Linux オペレーティング・システムの場合:

```
./Airgap.sh -serial serial_number -email
mail_address -createSiteList site_list_filename
 [-proxy
[user:password@]hostname:port] [-usehttps]
[-cacert crt_filename] [-othersites site_foldername]
 [-timeout timeout_seconds]
```

各表記の意味は次のとおりです。

mail_address

ライセンスで指定したメール・アドレスです。一致しない場合、エアー・ギャップ・ツールは失敗します。オプション `-email` は、オプション `-createSiteList` と一緒の場合のみ使用できます。

-proxy

パブリック・インターネットにアクセスできるワークステーションがプロキシー・サーバー経由でのみ接続できる場合に使用されるオプションです。この場合、`-proxy` オプションの後にプロキシー・サーバーのホスト名およびポートを *hostname:port* 形式で指定します。プロキシーが認証プロキシーである場合は、ユーザー ID およびパスワードも *userid:password@hostname:port* 形式で追加してください。

-usehttps

このオプションが指定された場合、ライセンス・サーバーと通信するために「https」が使用されます。エアー・ギャップ・ツールの実行場所とは別のフォルダーを使用する場合は、オプション `-cacert` を使用してファイル `ca-bundle.crt` を保管するパスを指定します。`-usehttps` オプションを使用する場合、または Fixlet 内で URL が「https」で始まっている場合、サーバー証明書を検証するためにファイル `ca-bundle.crt` が使用されます。

-cacert

このオプションは、オプション `-usehttps` と一緒の場合のみ使用できます。

-othersites

ご使用のライセンスで AllowOtherSites の使用が許諾されている場合は、このオプションを使用して任意のサイトをサイト・リストに含めます。サイト・リストを作成する場合は、フォルダーを作成して、ライセンスに含まれていないマストヘッドに関連するマストヘッド・ファイル (*.efxm ファイル) をすべてそのフォルダーにコピーし、このフォルダーの名前とオプション `-othersites` を指定します。

-timeout

このオプションは、V9.5.7 から使用可能になりました。これは、http タイムアウト間隔を秒単位で指定します。値は 30 から 3600 までの範囲です。デフォルト値は 30 です。プロキシーの使用中にエラー「HTTP エラー 28: タイムアウトに達しました」が発生する場合は、オプション `-usehttps` も使用してみてください。こうすると、プロキシーはトンネリング・モードで作動して、タイムアウトの回避に役立つ可能性があるためです。

ツールの実行後、*`site_list_filename`* に指定した名前でファイルが作成されます。

注: 作成されたサイト・リスト・ファイルは、ライセンスを変更するまで、または HCL が既存のライセンスに新規サイトを追加するまで使用できます。何らかの理由でサイト・リスト・ファイルを削除しても、ライセンスのシリアル番号が変更されない限り、ダウンロードしたファイルの履歴が維持されるため、同じコマンドでファイルを再作成することができます。

2. サイト・リスト・ファイルを編集します

ステップ 1 で作成されたファイルの各行には、以下のように 3 つの情報が 2 つのコロンで区切られて入っています。

```
flag::site_name::site_url
```

flag パラメーターのみを編集でき、以下のいずれかの値を指定できます。

A

サイト・コンテンツは、より新しいサイト・バージョンが使用可能になったときに収集されて、`AirgapResponse` ファイルに保管され、ファイルのダウンロードまたはファイル・リストの作成に使用されます。

R

サイト・コンテンツは、サイトのバージョンに関係なく、常に収集されて `AirgapResponse` ファイルに保管され、ファイルのダウンロードに使用されます。

G

サイト・コンテンツは、より新しいサイト・バージョンが使用可能になったときに収集されて、`AirgapResponse` ファイルに保管されますが、ファイルのダウンロードにもファイル・リストの作成にも使用されません。

Q

サイト・コンテンツは、サイトのバージョンに関係なく、常に収集されて `AirgapResponse` ファイルに保管されますが、ファイルのダウンロードにもファイル・リストの作成にも使用されません。

D

サイト・コンテンツは収集されませんが、ファイルのダウンロードまたはファイル・リストの作成に使用されます。このフラグは、サイトの現在のコンテンツを更新せずに維持した状態で、ファイルをダウンロードして現行のサイトで Fixlet を実行する場合に有用です。このオプションは、サイト・コンテンツが既に収集されている場合にのみ有効です。

N

サイトは無視されますが、サイト情報は今後参照できるようにファイル内に保持されます。

注: サイト・リスト・ファイルを作成する際、BES サポートおよび Web UI Common のコンポーネントのデフォルト値は G に設定されています。Web UI コンポーネントが必要でない場合は、デフォルトの Web UI Common 値を G から N に変更してください。その他のコンポーネントのデフォルト値は N に設定されています。BigFix サーバーをインストールした後、初回実行時に、ライセンス情報、BES サポートおよび Web UI Common のコンポーネントが収集される必要があります。パブリック・インターネットにアクセスできるワークステーションで生成されたこの最初のエアー・ギャップ応答を BigFix サーバーに移動した後でのみ、コンソールの「ライセンスの概要」ダッシュボードからアクセスできる他のコンポーネントを有効にして、プロセスを続行することができます。収集する前に、デフォルト以外の必要なコンポーネントを必ず有効にしてください。

3. サイト・コンテンツを収集してエアー・ギャップ応答ファイルを作成します

サイト・リスト・ファイルでフラグを編集した後、エアー・ギャップ・ツールを再び実行して、以下のいずれかのサイト操作を実行します。

a. サイト・コンテンツの収集

フラグ **A**、**R**、**G**、または **Q** が指定されたサイトのサイト・コンテンツを収集するには、以下のコマンドを実行します。

Windows オペレーティング・システムの場合:

```
BESAirgapTool.exe -site site_list_filenam
e
```

Linux オペレーティング・システムの場合:

```
./Airgap.sh -site site_list_filename
```

完了時に、`Airgapresponse` ファイルが作成されます。

b. サイト・コンテンツの収集およびファイルのダウンロード

フラグ **A**、**R**、**G**、または **Q** が指定されたサイトのサイト・コンテンツを収集し、フラグ **A**、**R**、または **D** が指定されたサイト上の Fixlet によって参照されているファイルをダウンロードするには、以下のコマンドを実行します。

Windows オペレーティング・システムの場合:

```
BESAirgapTool.exe -site site_list_filenam
e -download
[-cache cache_name]
```

Linux オペレーティング・システムの場合:

```
./Airgap.sh -site site_list_filename -dow
nload
[-cache cache_name]
```

ここで、*cache_name* は、ダウンロードしたファイルを保管するフォルダー・パスです。完了時に、`Airgapresponse` ファイル

が作成され、ファイルが *cache_name* フォルダーにダウンロードされます。

c. サイト・コンテンツの収集およびファイルの選択的なダウンロード

フラグ **A**、**R**、**G**、または **Q** が指定されたサイトのサイト・コンテンツを収集し、フラグ **A**、**R**、または **D** が指定されたサイト上の Fixlet によって参照されているファイルのリストを作成するには、以下のコマンドを実行します。

Windows オペレーティング・システムの場合:

```
BESAirgapTool.exe -site site_list_filenam
e
-createFileList referenced_list
```

Linux オペレーティング・システムの場合:

```
./Airgap.sh -site site_list_filename
-createFileList referenced_list
```

完了時に、`Airgapresponse` ファイル、および *referenced_list* で指定された名前のファイル・リストが作成されます。

いずれの場合も、フラグ **A**、**R**、**G**、または **Q** が指定されたサイトで収集されたサイト・コンテンツは `AirgapResponse` ファイル内に配置されます。エアー・ギャップ・ツールを初めて実行する際、フラグ **A**、**R**、**G**、または **Q** が指定されたすべてのサイトが収集されます。それ以降は、フラグ **A** または **G** が指定されたサイトのコンテンツが以前に収集されていない場合、またはより新しいサイトのバージョンが使用可能になった場合に限り、それらのコンテンツが収集されます。フラグ **R** または **Q** が指定されたサイトのコンテンツは常に収集されます。

オプションで、以下のオプションも指定できます。

`-usehttps`

ライセンス情報およびサイト・コンテンツは「https」を使用して収集されます。『b. サイト・コンテンツの収集およびファ

イルのダウンロード』の場合、「http」で始まるすべての URL で強制的に「https」が使用されます。Fixlet 内の一部の URL が「https」で始まり、一部のパッチ・サイトが「https」で始まる URL に要求をリダイレクトする可能性があることに注意してください。

`-proxy [user:password@]hostname:port`

パブリック・インターネットにアクセスできるワークステーションがプロキシー・サーバー経由でのみ接続できる場合に使用されます。この場合、`-proxy` オプションの後にプロキシー・サーバーのホスト名およびポートを *hostname:port* 形式で指定します。プロキシーが認証プロキシーである場合は、ユーザー ID およびパスワードも *userid:password@hostname:port* 形式で追加してください。

`-cacert crt_filename`

エアー・ギャップ・ツールの実行場所とは別のフォルダーを使用する場合は、ファイル `ca-bundle.crt` を保管するパスを指定します。`-usehttps` オプションを使用する場合、または Fixlet 内で URL が「https」で始まっている場合、サーバー証明書を検証するためにファイル `ca-bundle.crt` が使用されます。オプション `-cacert` はオプション `-usehttps` と一緒の場合のみ使用できます。

`-timeout timeout_seconds`

このオプションは、V9.5.7 から使用可能になりました。これは、http タイムアウト間隔を秒単位で指定します。値は 30 から 3600 までの範囲です。デフォルト値は 30 です。プロキシーの使用中にエラー「HTTP エラー 28: タイムアウトに達しました」が発生する場合は、オプション `-usehttps` も使用してみてください。こうすると、プロキシーはトンネリング・モードで作動して、タイムアウトの回避に役立つ可能性があるためです。

b および **c** の場合は、ダウンロードするファイルまたはファイル・リストで収集するファイルの数を減らすために、他のオプションも使用できます。これらのフィルタリング・オプションは、ファイル自体ではなく、ファイルを参照する Fixlet を選択します。例えば、過去 5 日間を指定する場合、過去 5 日間にベンダーによって追加または変更されたファイルではなく、過去 5 日間に変更された Fixlet によって参照されているファイルを意味します。フィルタリング・オプションに使用できる値のリストを作成するには、以下のコマンドを実行します。

Windows オペレーティング・システムの場合:

```
BESAirgapTool.exe -site site_list_filename
 -createfilterList
filter_list
```

Linux オペレーティング・システムの場合:

```
./Airgap.sh -site site_list_filename
 -createfilterList
filter_list
```

使用できる値のリストは、オプション `-fcategory`、`-fcve`、`-fproduct`、`-fseverity`、`-fsource`、および `-fsourceid` に制限されています。以下のオプションをフィルタリングに使用できます。

`-fcategory`

Fixlet カテゴリーのプロパティー。

`-fcve`

セキュリティー・パッチに関連付けられている CVE (共通脆弱性と暴露) ID を指定します。

`-fdays`

最終変更日がコマンドの実行日から指定された日数以内にある Fixlet を選択します。

`-fproduct`

Fixlet を適用できる製品名を指定します (`Win2008` や `Win7` など)。この情報はコンソールには表示されません。このオプションは、Windows オペレーティング・システムのパッチに関連したサイトでのみ使用できます。

`-fseverity`

ベンダーがセキュリティー・パッチに関連付ける重大度を指定します。

`-fsource`

ファイルのプロバイダー (BigFix、Adobe、または Microsoft など)。

`-fsourceid`

プロバイダーによって指定された ID。

`-includeCorrupt`

このオプションが指定されない場合はデフォルトで除外される、「破損」のマークが付けられた Fixlet を組み込みます。

`-includeSuperseded`

このオプションが指定されない場合はデフォルトで除外される、「置き換え済み」のマークが付けられた Fixlet を組み込みます。

複数のフィルター条件が指定された場合、すべての条件を満たす Fixlet のみが選択されます。オプション `-fsource`、`-fsourceid`、`-fcve`、`-fcategory`、および`-fseverity` では、複数のコンマ区切り値を指定できます (例えば、`-fseverity "Critical, Important"`. コンマを使用して値を区切る場合、または値にスペースが含まれている場合は、上記の例のように、パラメーターを二重引用符で囲んでください。値の大/小文字が区別されることに注意してください。

4. ファイル・リストを編集します

ステップ 3 の「**c. サイト・コンテンツの収集およびファイルの選択的なダウンロード**」にのみ適用されます。

`-createFileList` オプションにより、ファイルのリストが入ったファイルを作成できます。リストの各行には、以下のように情報が 2 つのコロンで区切られて入っています。

```
flag::site_name::Fixlet_id::site_url::
size::hash_value::hash algorithm
```

例:

```
N::site=site_name::fixletid=fixlet_id::
url=url_address::size=file_size::hash=hash_value::
hashtype=hash_type
```

`flag` 値のみを編集できます。ファイルをダウンロードする場合は **Y** に変更し、ファイルをダウンロードしない場合は **N** に変更します。

5. インターネット側のワークステーションでツールを実行してファイルをダウンロードします

ステップ 3 の「**c. サイト・コンテンツの収集およびファイルの選択的なダウンロード**」にのみ適用されます。
ステップ 4 でファイル・リストを編集した後、ファイル・リストでフラグ **Y** が指定されたファイルのみをダウンロードするには、以下のコマンドを発行してエアー・ギャップ・ツールを実行します。

Windows オペレーティング・システムの場合:

```
BESAirgapTool.exe -file file_list_filename
 -download
-cache cache_foldername
[-proxy [user:password@]hostname:port] [-usehttps]
[-cacert crt_filename]
```

Linux オペレーティング・システムの場合:

```
./Airgap.sh -file file_list_filename -download
-cache cache_foldername
```

```
[-proxy [user:password@]hostname:port] [-usehttps]
[-cacert crt_filename]
```

ここで、*cache_foldername* は、ダウンロードしたファイルを保管するフォルダー・パスです。既にキャッシュ・フォルダー内にあるファイルが再びダウンロードされることはありません。

6. エアー・ギャップ応答ファイルを BigFix サーバーに移動して、BigFix サーバーでエアー・ギャップ・ツールを実行します

`AirgapResponse` ファイル、およびステップ 3 で作成したファイル・リストまたはステップ 5 で収集してダウンロードしたファイルをポータブル・ドライブにコピーして、それらを BigFix サーバー・コンピューターに転送します。`AirgapResponse` ファイルがエアー・ギャップ・ツールと同じフォルダーにあることを確認し、以下のコマンドを発行して実行します。

Windows オペレーティング・システムの場合:

```
BESAirgapTool.exe -run [-temp temp_folder]
```

Linux オペレーティング・システムの場合:

```
./Airgap.sh -run [-temp temp_folder]
```

これにより、Fixlet コンテンツとライセンスの更新を持つ応答ファイルが現在の適用環境にインポートされます。

注: エアー・ギャップ・ツールは、BigFix サーバーの GatherDB コンポーネントにサイト・コンテンツを応答ファイルで渡します。GatherDB コンポーネントは、サイト・コンテンツをインポートします。WebUI サイト以外のサイトの場合、インポートの進行状況を GatherDB コンポーネントの DebugOut (デフォルトの名前は `GatherDB.log`) でモニターできます。

ダウンロードしたファイルを BigFix サーバーのキャッシュ・フォルダーにもコピーします。キャッシュ・フォルダーのデフォルトの場所は以下のとおりです。

Windows オペレーティング・システムの場合:

```
%PROGRAM FILES%\BigFix Enterprise\BES Server\wwwrootbes
\bfmirror\downloads\sha1
```

Linux オペレーティング・システムの場合:

```
/var/opt/BESServer/wwwrootbes/bfmirror/downloads/sha1
```

上記のステップを定期的に繰り返して、メインの BigFix サーバーの Fixlet コンテンツが常に更新されるようにしてください。新しい Fixlet メーリング・リストに参加すると、Fixlet の更新時に通知を受け取ることができます。必ず、エアー・ギャップ・ツールのバージョンが、インストールされている BigFix サーバーのバージョンと互換性があることを確認してください。

使用法のヒント:

1. ステップ 1 で使用された AirgapTool とまったく同じバージョンを BigFix ルート・サーバーのディレクトリーの中に解凍します。
2. ディレクトリーに `airgapresponsefile` をコピーします。
3. オプションを指定せずに `BESAirgapTool.exe` を実行します。`airgapresponsefile` のコンテンツがディレクトリーにインポートされます。ステップ 5 で任意のファイルをダウンロードする場合、ルートサーバーの SHA1 ディレクトリーにこれらのファイルもコピーします。エアー・ギャップ・ツールはファイルをダウンロードし、SHA256 の値で名前を付けるため、これが必要になる場合があります。

注: SHA 1 ディレクトリーにペーストした後、SHA256 の値を SHA1 の値として名前変更する必要ありません。

オプションのアクション:

すべての必要なファイルがダウンロードされたことの確認

適用する予定の Fixlet に必要なすべてのファイルをダウンロードしたことを確認するには、エアー・ギャップ・ツールの実行時にオプション `-checkfixlet` を使用します。例:

Windows オペレーティング・システムの場合:

```
BESAirgapTool.exe -site site_list.txt -ch
eckfixlet
-fdays 100 -fseverity Critical -cache MyC
ache
```

Linux オペレーティング・システムの場合:

```
./Airgap.sh -site site_list.txt -checkfix
let
-fdays 100 -fseverity Critical -cache MyC
ache
```

指定されたフィルタリング条件を満たす Fixlet について、ツールは、ダウンロード履歴および宛先フォルダーの内容を検査します。ダウンロードすべきファイルがまだある場合は、Fixlet 名および URL が表示されます。

手動でダウンロードされるファイル

Fixlet によって参照される一部のファイルをダウンロードできない場合があります。それは、ベンダーのサポート・センターに連絡することでのみ取得可能であるため、またはダウンロード・サイトでライセンス条項を明示的に受け入れる必要があり、法律上の理由からそのアクションを自動化できないためです。このような場合、関与するファイルにストリング `MANUAL_BES_CACHING_REQUIRED` を含むダウンロード URL が指定されています。そのファイルは手動でダウンロードする必要があります。このようなファイルのリストを作成するには、次の例のようにオプション `-createmanuallist` を使用します。

Windows オペレーティング・システムの場合:

```
BESAirgapTool.exe -site site_list.txt -cr
eatemanuallist
manual_list -fseverity Critical
```

Linux オペレーティング・システムの場合:

```
./Airgap.sh -site site_list.txt -createma
nuallist
manual_list -fseverity Critical
```

手動でダウンロードする必要があるすべてのファイルが宛先フォルダーに入っているかどうかを確認するために、次の例のように `-checkmanual` オプションを使用することもできます。

Windows オペレーティング・システムの場合:

```
BESAirgapTool.exe -site site_list.txt -ch
eckmanual
-fseverity Critical
-fdays 30 -cache MyCache
```

Linux オペレーティング・システムの場合:

```
./Airgap.sh -site site_list.txt -checkman
ual
-fseverity Critical
-fdays 30 -cache MyCache
```

履歴のリセット

エアー・ギャップ・ツールは、ダウンロードしたファイルの履歴を維持します。ダウンロードしたすべてのファイルをパブリック・インターネット側のワークステーションから BigFix サーバーに移動しても、この履歴は維持され、以前にダウンロードされたファイルが再びダウンロードされることはないため、時間とディスク・スペースを節約できます。以前にダウンロードしたファイルの一部またはすべてを削除した後で、再び必要になった場合は、`-resync` オプションを使用できます。このオプションは、ダウンロード履歴をクリアして、`-cache` オプションで指定されたフォルダー内のファイルを検査します。新しく作成されたダウンロード履歴が `-cache` オプションで指定さ

れたフォルダーに含まれているファイルのみに基づいていることに注意してください。

ライセンスの変更

別のライセンスを管理する場合は、収集したサイトおよびダウンロードしたファイルの履歴を消去する必要があります。このアクションを実行するには、次の例のように `-force` オプションを使用します。

Windows オペレーティング・システムの場合:

```
BESAirgapTool.exe -serial serial_number -email
mail_addess -createSiteList site_list_filename -force
```

Linux オペレーティング・システムの場合:

```
./Airgap.sh -serial serial_number -email
mail_addess -createSiteList site_list_filename -force
```

その他のオプション

デフォルトでは、エアー・ギャップ・ツールは 2 つのファイルを同時にダウンロードします。`-download` オプションの後に数字を指定することにより、同時にダウンロードするファイルの数を変更できます。この数字は 1 から 8 の範囲内にすることができます。例えば、3 つのファイルを同時にダウンロードするには、`-download 3` と指定します。3 つ以上のファイルを同時にダウンロードする場合は、さらに大きな帯域幅が必要になることに注意してください。

Fixlet で指定されている URL が「https」で始まる場合、または `-useHttps` オプションを指定する場合、エアー・ギャップ・ツールは、URL で指定されたサーバーに適切な SSL サーバー証明書があることを検証しようとします。何らかの理由で、こ

の検査をスキップして、エアー・ギャップ・ツールがサーバー証明書を検証できない場合のダウンロードの失敗を回避するには、`-noverify` オプションを使用します。このオプションを使用した場合、エアー・ギャップ・ツールはサーバー証明書の認証を確認しませんが、サーバー証明書が操作対象のURLで指定されたサーバー用であることを確認します。DNS を調べて、ワークステーションがホスト名を正しく変換していることを確認する必要があります。

エアー・ギャップ・ツールが通常よりも多くの情報を出力するようにするには、`-verbose` オプションを使用します。

複数の BigFix サーバーでの作業

テスト・サーバーと実動サーバーなど、複数の BigFix サーバーで同じパブリック・インターネット側のワークステーションを使用する場合は、エアー・ギャップ・ツールを各フォルダーにコピーして、各フォルダーで別々に作業します。別々のフォルダーで同じサイト・リストを共有できますが、各サーバーは独自の履歴をフォルダーで維持します。別々のサーバーで複数のエアー・ギャップ・ツールを使用する場合は、さまざまなサーバーに共通のファイルを 1 回のみダウンロードするためにキャッシュ・フォルダーを共有することもできますが、同時に実行されるエアー・ギャップ・ツールのインスタンスが 1 つのみであることを確認する必要があります。
サイトのセットを収集して、それらをテスト・サーバーにロードしてから、収集したサイトでテストを実行し、最新のサイトではなく、テストしたサイトを実動サーバーにロードする場合、複数の BigFix サーバーに同じ製品 (BigFix Lifecycle、BigFix Compliance など) のライセンスが提供されるときに、それらのサーバーに 1 つの `AirgapResponse` ファイルをロードすることができます。1 つの `AirgapResponse` ファイルを複数の BigFix サーバーにロードする場合、すべての BigFix サーバーで有効になっているサイトのみを収集することをお勧めします。

注: BigFix サーバーをインストールした後、初回実行時に、各インストール済み環境のライセンス情報、BES サポートおよび Web UI Common のコンポーネントが収集される必要があります。ライセンス情報はシリアル番号ごとに固有であるため、このステップでは、BigFix サーバーごとに AirgapResponse ファイルが作成される必要があります。

どのサイトでもバージョンを変更することなく、特定の BigFix サーバーのライセンス情報を更新する場合、行が含まれていないサイト・ファイル、またはすべてのサイトにフラグ **N** が指定されているサイト・ファイルを指定してエアー・ギャップ・ツールを実行することにより、ライセンス情報のみが入った `AirgapResponse` ファイルを作成できます。以下のコマンドを実行します。

Windows オペレーティング・システムの場合:

```
BESAirgapTool.exe -site empty_site_list_filename
-allowemptysite
```

Linux オペレーティング・システムの場合:

```
./Airgap.sh -site empty_site_list_filename
-allowemptysite
```

エアー・ギャップ環境での WebUI の有効化

エアー・ギャップ環境に WebUI をインストールするには、以下のステップを実行します。

1. 最新の BES サポートおよび WebUI Common のサイトを収集して、WebUI サービスのインストールに必要なファイルをダウンロードします。それらのファイルを BigFix サーバーにロードします。
2. BES サポート・サイトのタスク「HCL BigFix WebUI サービスのインストール」を使用して、WebUI サービスをインストールします。
3. インストールが完了した後、WebUI ターゲット・システムで WebUI サービス (Windows オペレーティング・システム) またはプロセス (Linux オペレーティング・システム) がアクティブ化されるまで待ちます。WebUI の初期化が開始されます。完了するまで待ちます。初期化は通常は数分間で完了しますが、30 分以上待ってからステップ 4 に進むことをお勧めします。
4. すべての最新の WebUI サイトを収集して、BigFix サーバーにロードします。WebUI サービスをインストールするタスクを実行する前に WebUI サイトを収集できますが、ロードできるのは、WebUI の初期化が完了した後のみです。

抽出使用

エアー・ギャップ・ツールの「抽出使用」モード。

重要: BigFix 9.5.7 のフレッシュ・インストールを実行する場合、WebUI サイトを使用できるようにするには、以下のステップを実行する必要があります。

1. WebUI をインストールして、エアー・ギャップ・ツールを実行します。
2. WebUI の初期化が完了するまで数分間待機します。
3. エアー・ギャップ・ツールを再実行します。

分離されたネットワークで Fixlet コンテンツと製品ライセンスの更新を使用できるようにするには、以下の手順を実行して、インターネットに接続できるコンピューターからこのユーティリティーを転送する必要があります。

Windows オペレーティング・システム

1. **BigFix サーバーで実行します**

   BigFix サーバーのインストール・ディレクトリーから `BESAirgapTool.exe` をダブルクリックするか、またはパラメーターを指定せずにコマンド・ラインからこのファイルを実行します。グラフィカル・ユーザー・インターフェースが開かれます。

   エアー・ギャップ・ツールのサイト要求と、このツールの実行に必要なすべてのファイルを格納するために、このツールの宛先フォルダーを指定します。エアー・ギャップ・ツールによるファイルのコピーが完了した後、フォルダー全体をポータブル・ドライブにコピーします。
2. **エアー・ギャップ要求を移動して、インターネット側のコンピューターで実行します**

   上記のポータブル・ドライブをインターネットに接続できるコンピューターに移動します。ユーザーには、`BESAirgapTool.exe` が配置されたフォルダーに対する書き込み権限が必要です。このフォルダーに移動して `BESAirgapTool.exe` をダブルクリックしてエアー・ギャップ・ツールを実行するか、またはコマンド・ラインからこのツールを起動します。

   オプションで、以下のコマンド・ライン・パラメーターも指定できます。

   -usehttps

   「http」で始まるすべての URL は、ライセンス情報とサイト・コンテンツを収集するために強制的に「https」を使用するようになります。Fixlet 内の一部の URL が「https」で始まり、一部のパッチ・サイトが「https」で始まる URL に要求をリダイレクトする可能性があることに注意してください。

   -proxy [*user*:*password*@]*hostname*:*port*

   このオプションは、BigFix バージョン 9.5.5 以降でのみ使用可能です。パブリック・インターネットにアクセスできるワークステーションがプロキシー・サーバー経由でのみ接続できる場合に使用されま

す。この場合、`-proxy` オプションの後にプロキシー・サーバーのホスト名およびポートを *hostname:port* 形式で指定します。プロキシーが認証プロキシーである場合は、ユーザー ID およびパスワードも *userid:password@hostname:port* 形式で追加してください。抽出使用では、現行ユーザーのクライアント・レジストリー設定または Internet Explorer 設定でプロキシー・サーバーが構成されていて、`-proxy` オプションが指定されていない場合、プロキシー設定は以前のバージョンのエアー・ギャップ・ツールと同じように使用されます。`-proxy` オプションを使用する場合、他の設定に関係なく、指定された値が使用されます。

-cacert <*full_path_to_ca-bundle.crt_file*>

エアー・ギャップ・ツールの実行場所とは別のフォルダーを使用する場合に、ファイル `ca-bundle.crt` を保管するパスを指定します。`-usehttps` オプションを使用する場合、または Fixlet 内で URL が「https」で始まっている場合、サーバー証明書を検証するためにファイル `ca-bundle.crt` が使用されます。オプション `-cacert` は `-usehttps` オプションと一緒の場合のみ使用できます。

グラフィカル・ユーザー・インターフェースが開かれます。エアー・ギャップ・ツールは、エアー・ギャップ要求で必要なファイルをすべて `BESAirgapTool.exe` と同じフォルダーにダウンロードします。これにより、エアー・ギャップ要求ファイルがエアー・ギャップ応答ファイルに交換されます。エアー・ギャップ応答ファイルをポータブル・ドライブにコピーします。

3. **エアー・ギャップ応答を BigFix サーバーに移動して、BigFix サーバーでエアー・ギャップ・ツールを実行します**
   ポータブル・ドライブを BigFix サーバー・コンピューターに戻し、`BESAirgapTool.exe` をダブルクリックするか、またはパラメーターを指定せずにコマンド・ラインから起動することにより、`BESAirgapTool.exe` をもう一度実行します。この際、必ず以下の権限と許可が与えられたユーザーとしてログオンして実行してください。
   - 管理者権限。
   - BFEnterprise データベースにコンテンツを追加する上で必要なデータベース許可。

グラフィカル・ユーザー・インターフェースが開かれます。

これにより、Fixlet コンテンツとライセンスの更新を持つエアー・ギャップ応答ファイルが現在のデプロイメントにインポートされます。

エアー・ギャップ・ツールは、`TEMP` 環境変数で指定されたフォルダー内に一時ファイルを作成します。一時ファイル用に別のフォルダーを使用する場合は、`BESAirgapTool.exe` を実行する前に、`TEMP` 環境変数をそのフォルダーに設定します。

メインの BigFix サーバーの Fixlet コンテンツを更新するには、上記の手順を定期的に繰り返します。新しい Fixlet メーリング・リストに参加すると、Fixlet の更新時に通知を受け取ることができます。

エアー・ギャップ・ツールのバージョンが、インストールされている BigFix サーバーのバージョンと互換性があることを確認してください。

Linux オペレーティング・システムの場合

1. **BigFix サーバーで実行します**
   Linux コンピューターで、BigFix サーバーがインストールされているのと同じパス内にエアー・ギャップ・ツールが存在することを確認します。デフォルト・パスは `/opt/BESServer/bin` です。Linux 端末を開き、以下のコマンドを入力して、`airgap.tar` という tar ファイルを作成します。このファイルには、BigFix データベースに関する情報に基づいて `AirgapRequest.xml` ファイルが格納されます。

   ```
   # cd /opt/BESServer/bin
   # ./Airgap.sh -remotedir directory
   ```

   各部の意味は以下のとおりです。

   -remotedir *directory*

   Airgap を実行して、指定されたフォルダー内に要求ファイルを生成します。
2. **エアー・ギャップ要求を移動して、インターネット側のコンピューターで実行します**
   `airgap.tar` ファイルをポータブル・ドライブにコピーして、以下のコマンドを発行し、`airgap.tar` ファイルのコンテンツを解凍します。

```
# tar -xf airgap.tar
```

システムに `LD_LIBRARY_PATH` という環境変数が存在し、この環境変数が DB2 ライブラリー `libdb2.so.1` が含まれるフォルダーのパスに設定されていることを確認してください。`Airgap.sh` ファイルと `AirgapRequest.xml` ファイルが同じフォルダーに存在し、そのフォルダーへの書き込み権限を持っていることを確認してください。`Airgap.sh` コマンドを実行します。
オプションで、以下のコマンド・ライン・パラメーターも指定できます。

-usehttps

「http」で始まるすべての URL は、ライセンス情報とサイト・コンテンツを収集するために強制的に「https」を使用するようになります。Fixlet 内の一部の URL が「https」で始まり、一部のパッチ・サイトが「https」で始まる URL に要求をリダイレクトする可能性があることに注意してください。

-proxy [*user*:*password*@]*hostname*:*port*

パブリック・インターネットにアクセスできるワークステーションがプロキシー・サーバー経由でのみ接続できる場合に使用されます。この場合、`-proxy` オプションの後にプロキシー・サーバーのホスト名およびポートを *hostname:port* 形式で指定します。プロキシーが認証プロキシーである場合は、ユーザー ID およびパスワードも *userid:password@hostname:port* 形式で追加してください。

-cacert <*full_path_to_ca-bundle.crt_file*>

エアー・ギャップ・ツールの実行場所とは別のフォルダーを使用する場合に、ファイル `ca-bundle.crt` を保管するパスを指定します。`-usehttps` オプションを使用する場合、または Fixlet 内で URL が「https」で始まっている場合、サーバー証明書を検証するためにファイル `ca-bundle.crt` が使用されます。オプション `-cacert` は `-usehttps` オプションと一緒の場合のみ使用できます。

これにより、エアー・ギャップ要求ファイルがエアー・ギャップ応答ファイルに交換されます。エアー・ギャップ応答ファイルをポータブル・ドライブにコピーします。

エアー・ギャップ・ツールの実行時に次のエラー・メッセージが表示されることがあります。

```
./Airgap: error while loading shared libraries: libdb2.so.1:
cannot open shared object file: No such file or directory
```

この場合、次のコマンドを実行することにより、`LD_LIBRARY_PATH` 変数を作成してエクスポートしてください。

```
export LD_LIBRARY_PATH="$LD_LIBRARY_PATH:/your/path/"
```

各部の意味は以下のとおりです。

/your/path

これは、DB2 ライブラリー `libdb2.so.1` が含まれるフォルダーのパスです。

3. **エアー・ギャップ応答を BigFix サーバーに移動して、BigFix サーバーでエアー・ギャップ・ツールを実行します**
ポータブル・ドライブを BigFix サーバー・コンピューターに再び接続して、`Airgap.sh` コマンドを実行します。これにより、Fixlet コンテンツとライセンスの更新を持つ応答ファイルが現在の適用環境にインポートされます。

```
# cd airgap
# ./Airgap.sh -run
```

オプションで、以下のオプションも指定できます。

-temp *directory*

エアー・ギャップ・ツールは `/tmp` ディレクトリーに一時ファイルを作成しますが、十分なスペースが残っていない場合は、このオプションを使用して、十分なスペースがある別のフォルダーを指定できます。

`Airgap.sh` ファイルと `AirgapRequest.xml` ファイルが同じフォルダー内に存在する必要があることに注意してください。

メインの BigFix サーバーの Fixlet コンテンツを更新するには、上記の手順を定期的に繰り返します。新しい Fixlet メーリング・リストに参加すると、Fixlet の更新時に通知を受け取ることができます。

エアー・ギャップ・ツールのバージョンが、インストールされている BigFix のバージョンと互換性があることを確認してください。

ダウンロードしたファイルの転送

メインの BigFix サーバーに Fixlet を適用するには、インターネットからダウンロードしたパッチやその他のファイルが必要になります。エアー・ギャップ・ツールは、サイト・コンテンツを収集するために抽出使用で使用したり、ファイルをダウンロードするために非抽出使用で使用したりすることができます (非抽出使用で生成される AirgapResponse ファイルを無視できます)。

あるいは、BES Download Cacher ユーティリティーを使用できます。このユーティリティーは以下を行う際に役立ちます。

- ファイルをダウンロードし、メインの BigFix サーバーに転送する。
- Fixlet サイトのパッチ・コンテンツのダウンロード、または URL からの単一ファイル・ダウンロードを行う。

現在のユーティリティーは http://software.bigfix.com/download/bes/util/BESDownloadCacher.exe からダウンロードできます。使用可能なオプションのリストを表示するには、`BESDownloadCacher.exe /?` を実行します。BES Download Cacher ユーティリティーを実行するシステムに BigFix サーバーまたは BigFix リレーがインストールされている場合は、ユーティリティーが関連するローカルの BES 設定を検出してデフォルトとして再使用するため、`-x` ユーティリティー・パラメーターはオプションとなります。

一部のサイトでは、アクセスを制限しているパッチ・ベンダーからコンテンツをダウンロードするための追加ステップが必要になります。追加情報については、以下のリンクをクリックして、それぞれのナレッジ文書を参照してください。これらの文書には、Solaris、Red Hat、SuSE 、および AIX の場合に、ツールを使用してパッチを手動でダウンロードする方法が記載されています。

こうしたサイトでは、3 ステップのプロセスが必要になります。

1. BESAirgapTool.exe を実行して、各サイトの Fixlet とタスクをダウンロードします。
2. BES Download Cacher ユーティリティーを実行して、BigFix からサイト用のツールをダウンロードします。
3. 各ベンダーのダウンロード・ツールを実行して、パッチ・コンテンツをダウンロードします。

Fixlet サイトからのすべてのファイルの転送

Fixlet サイトからファイルを転送するには、以下の手順を実行します。

1. ダウンロードの収集元となるサイトの `efxm` ファイル (`BES Asset Discovery.efxm` など) を探します。
2. 以下のコマンドを使用して BES Download Cacher ユーティリティーを実行します。

```
BESDownloadCacher.exe -m BES Asset Discovery.efxm -x downloads
```

注: このコマンドの実行は、Fixlet サイトで参照されているすべてのファイルがダウンロードされてダウンロード・フォルダーに格納されるため、非常に長時間かかる可能性があります。ダウンロード・フォルダー内にすでに存在しているファイルについては、もう一度ダウンロードされることはありません。ファイルは、その sha1 チェックサムを使用して名前が付けられます。

3. ダウンロードが終了したら、ダウンロード・フォルダーの内容 (フォルダーではなく、ファイルのみ) をメインの BigFix サーバーの sha1 フォルダーにコピーします。sha1 フォルダーのデフォルト・ロケーションは以下のとおりです。
   - Windows システムの場合: `%PROGRAM FILES%\BigFix Enterprise\BES Server\wwwrootbes\bfmirror\downloads\sha1`
   - Linux システムの場合: `/var/opt/BESServer/wwwrootbes/bfmirror/downloads/sha1`

   BigFix サーバーは、インターネットからダウンロードする代わりに上記のファイルを使用します。

注: BES Download Cacher ユーティリティーを後で実行する場合は、ファイルの変更時刻を調べて、どのファイルが最新かを確認することができます。この方法では、毎回すべてのファイルをコピーするのではなく、最新のファイルだけをメインの BigFix サーバーに転送します。

場合によっては、メインの BigFix サーバーによってキャッシュからファイルが削除されないように、サーバー上のキャッシュのサイズを増やす必要があります。BES Download Cacher ユーティリティーを実行してキャッシュのサイズを増やすには、以下のコマンドを使用します。

```
BESDownloadCacher.exe -c 1024
```

キャッシュのデフォルト・サイズは 1024 MB です。

注: `-c` オプションは、BES Download Cacher ユーティリティーを実行するシステムに BigFix サーバーまたはリレーがインストールされている場合にのみ使用します。BigFix コンポーネントがインストールされていない場合は、キャッシュが無制限になります。

これらのファイルが BigFix サーバーの sha1 フォルダーにキャッシュされた後で、ダウンロード済みファイルを参照する Fixlet メッセージ内のアクションをクリックすると、これらのファイルが自動的に BigFix リレーと BigFix クライアントに配信されます。ファイルがキャッシュされていない状態でアクションを適用すると、BigFix コンソールから「`Waiting for Mirror Server`」という状況が返されます。

単一ファイルの転送

単一のファイルを Fixlet サイトから転送するには、以下の手順を実行します。

1. 以下のコマンドを使用して BES Download Cacher ユーティリティーを実行します。

```
BESDownloadCacher.exe -u http://www.mysite/downloads/myplugin.exe -x
 downloads
```

2. ダウンロードが終了したら、ダウンロード・フォルダーの内容 (フォルダーではなく、ファイルのみ) をメインの BigFix サーバーの sha1 フォルダーにコピーします。sha1 フォルダーのデフォルト・ロケーションは以下のとおりです。
    - Windows システムの場合: `%PROGRAM FILES%\BigFix Enterprise\BES Server\wwwrootbes\bfmirror\downloads\sha1`
    - Linux システムの場合: `/var/opt/BESServer/wwwrootbes/bfmirror/downloads/sha1`

場合によっては、メインの BigFix サーバーによってキャッシュからファイルが削除されないように、サーバー上のキャッシュのサイズを増やす必要があります。BES Download Cacher ユーティリティーを実行してキャッシュのサイズを増やすには、以下のコマンドを使用します。

```
BESDownloadCacher.exe -c 1024
```

キャッシュのデフォルト・サイズは 1024 MB です。

注: `-c` オプションは、BES Download Cacher ユーティリティーを実行するシステムに BigFix サーバーまたはリレーがインストールされている場合にのみ使用します。BigFix コンポーネントがインストールされていない場合は、キャッシュが無制限になります。

これらのファイルが BigFix サーバーの sha1 フォルダーにキャッシュされた後で、ダウンロード済みファイルを参照する Fixlet メッセージ内のアクションをクリックすると、これらのファイルが自動的に BigFix リレーと BigFix クライアントに配信されます。ファイルがキャッシュされていない状態でアクションを適用すると、BigFix コンソールから「Waiting for Mirror Server」という状況が返されます。

ログ・ファイル

エアー・ギャップ・ツールは、通常ログ・ファイルとデバッグ・ログ・ファイルの 2 つのタイプのログ・ファイルを作成します。

通常ログ・ファイルには、コマンド・ウィンドウに表示されるメッセージが記録されるため、特定の日付に収集されたサイトなどのエアー・ギャップ・タスクを確認できます。デ

バッグ・ログ・ファイルは、HCL サポート・チーム向けのものです。通常ログ・ファイルの命名規則は次のとおりです。

Windows オペレーティング・システムの場合:

`BESAirgapTool_yyyy-mm-dd.log`

Linux オペレーティング・システムの場合:

`Airgap_yyyy-mm-dd.log`

ここで、*yyyy-mm-dd* はファイルの作成日付です。V9.5.7 以降、30 日間を経過したファイルは削除されます。

デバッグ・ログ・ファイルは、`AirgapDebugOut.txt` です。V9.5.7 以降、このファイルには最終日の情報のみが格納され、それより前のログ・ファイルは `AirgapDebugOutyyyymmdd.txt` に名前変更されます。ここで、yyyymmdd はファイルの作成日付です。10 日間を経過したファイルは削除されます。エアー・ギャップ・ツールは、詳細オプション `-verbose` を使用することで、さらに多くの情報をデバッグ・ログ・ファイルに書き込むことができます。

第 13 章. BigFix Query の使用によるクライアント情報の取得

BigFix Query 機能により、WebUI BigFix Query アプリケーションから、または REST API を使用して、クライアント・ワークステーションに関する情報を取得して関連度の照会を実行できます。

BigFix Query 機能を使用すると、以下のことを行うことができます。

- BigFix 環境のパフォーマンスに影響を与えずにクライアントからデータを素早く収集する。
- 適用可能性の関連度を使用して識別されたターゲット、または一連のターゲット・エージェント ID に関して、Relevance Language で照会を実行する。
- 収集された結果を WebUI Query アプリケーションに表示する (オプションで結果をページングする)。表示される結果は、クライアントから新規の値を受信するにつれて、定期的に更新されます。
- 正規版を展開する前に、いくつかの選択したクライアント上で関連式をテストする。

このガイドには、BigFix Query を使用するために、BigFix を構成する方法に関する情報が記載されています。追加情報は、以下のリンクをクリックすると参照できます。

- BigFix Query
- クエリー (（ページ）) (設定のリストと詳細な説明 (（ページ）) 内)

BigFix Query要件

BigFix Query 要求の対象となるクライアントは、特定の条件を満たす必要があります。

クライアント上で BigFix Query を実行するには、以下の要件を満たす必要があります。

- クライアントが UDP 通知を受信できる。BigFix Query 機能は、プロキシーまたはファイアウォール経由で BigFix サーバーに接続されたコンポーネントをサポートしていません。
- BigFix V9.5 パッチ 2 以降が、クライアント・マシン、およびそのクライアントに到達するために通過しなければならない中間リレーすべてにインストールされている必要がある。

BigFix Query の制約事項

BigFix Query 機能の使用時には、いくつかの制約事項が適用されます。

以下の制約は、BigFix Query 機能の使用に影響を及ぼします。

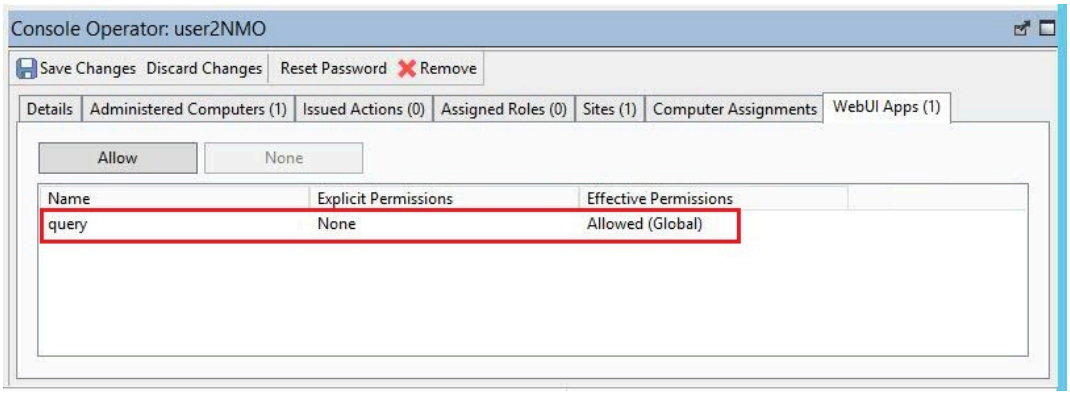
- この機能は、BigFix Lifecycle または BigFix Compliance バージョン 9.5 パッチ 2 以降のバージョンに対してのみ使用可能です。
- バージョン 9.5.13 以降でこの機能は、エージェント・コンテキストを必要とする要求をサポートします。
- 災害対策サーバー・アーキテクチャー (DSA) 内に環境を構成した場合は、以下に注意してください。
    - BigFix Query に関する情報は、複数サーバー間で複製されません。
    - 各サーバーは、BigFix Query 要求を、照会が送信されたサーバーに直接またはリレー経由で接続するクライアント上でのみ実行できます。

BigFix Query を使用できるユーザー

BigFix Query 要求は、マスター・オペレーターおよびマスター以外のオペレーターによって実行できます。オペレーターがこの機能を使用できるようにするには、特定のアクセス許可を設定する必要があります。

WebUI ツールバーから WebUI Query アプリケーションにアクセスする場合:

ユーザーは、オペレーターまたは役割のレベルで、**query** WebUI アプリケーションに対する「有効な権限」が**「許可」**に設定されている必要があります。以下はその例です。

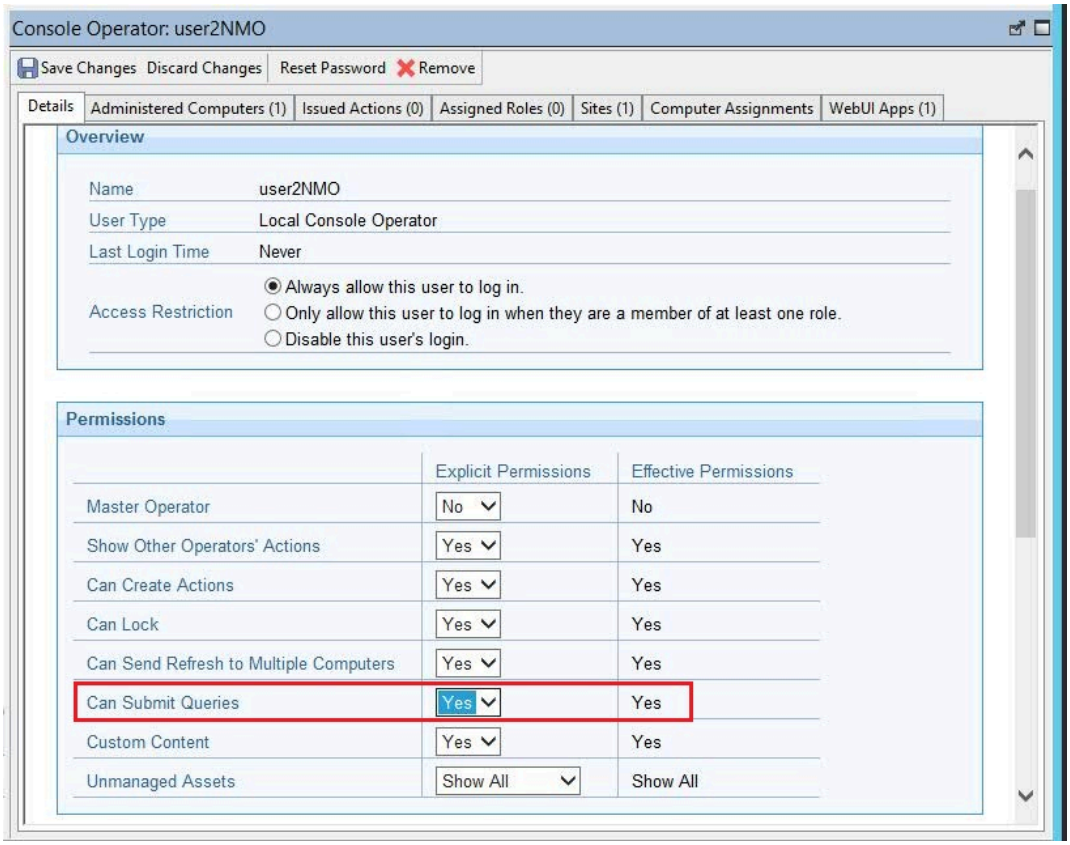


別の方法として、**「WebUi アプリケーション」**ドメインの作業域内で、どの権限が WebUI アプリケーション上のユーザーに割り当てられているか参照できます。**「WebUi アプリケーション」**ドメインは、WebUI を有効にした後、**「すべてのコンテンツ」**の下で使用可能になります。

WebUI Query アプリケーションへのアクセス方法について詳しくは、WebUI からの BigFix Query の実行方法 (（ページ） *148)*を参照してください。

BigFix Query 要求を実行して、それらの結果を参照する場合:

マスター・オペレーターは、デフォルトで照会を実行できます。マスター以外のオペレーターは、オペレーターまたは役割のレベルで、以下のように**「詳細」**タブの**「照会を送信可能」** 権限が**「はい」**に設定されている必要があります。



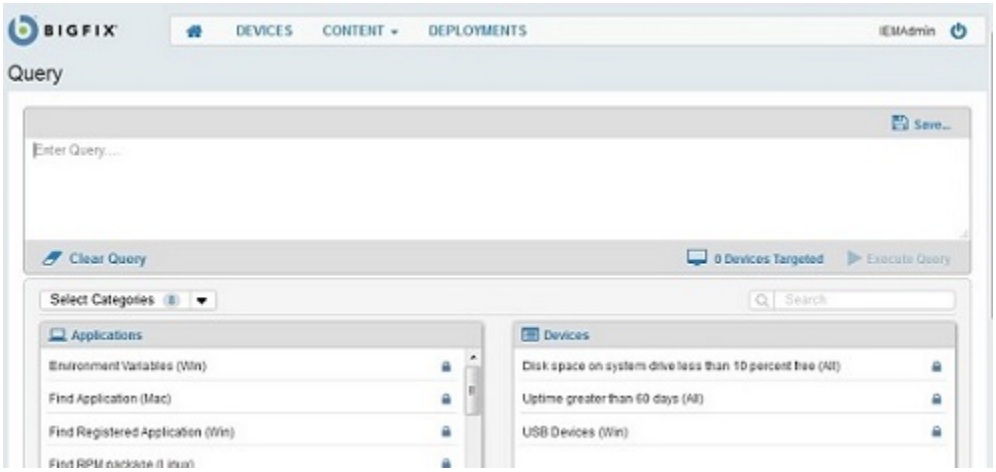
マスター・オペレーター以外の **「照会を送信可能」**権限のデフォルト値は、**「いいえ」**です。

オペレーター権限および役割について詳しくは、「ローカル・オペレーターの追加 (（ページ） 26)」を参照してください。

WebUI からの BigFix Query の実行方法

「コンテンツ」->「クエリー」を選択して、WebUI ユーザー・インターフェース上の BigFix Query にアクセスできます。

以下のような「クエリー」パネルが開きます。

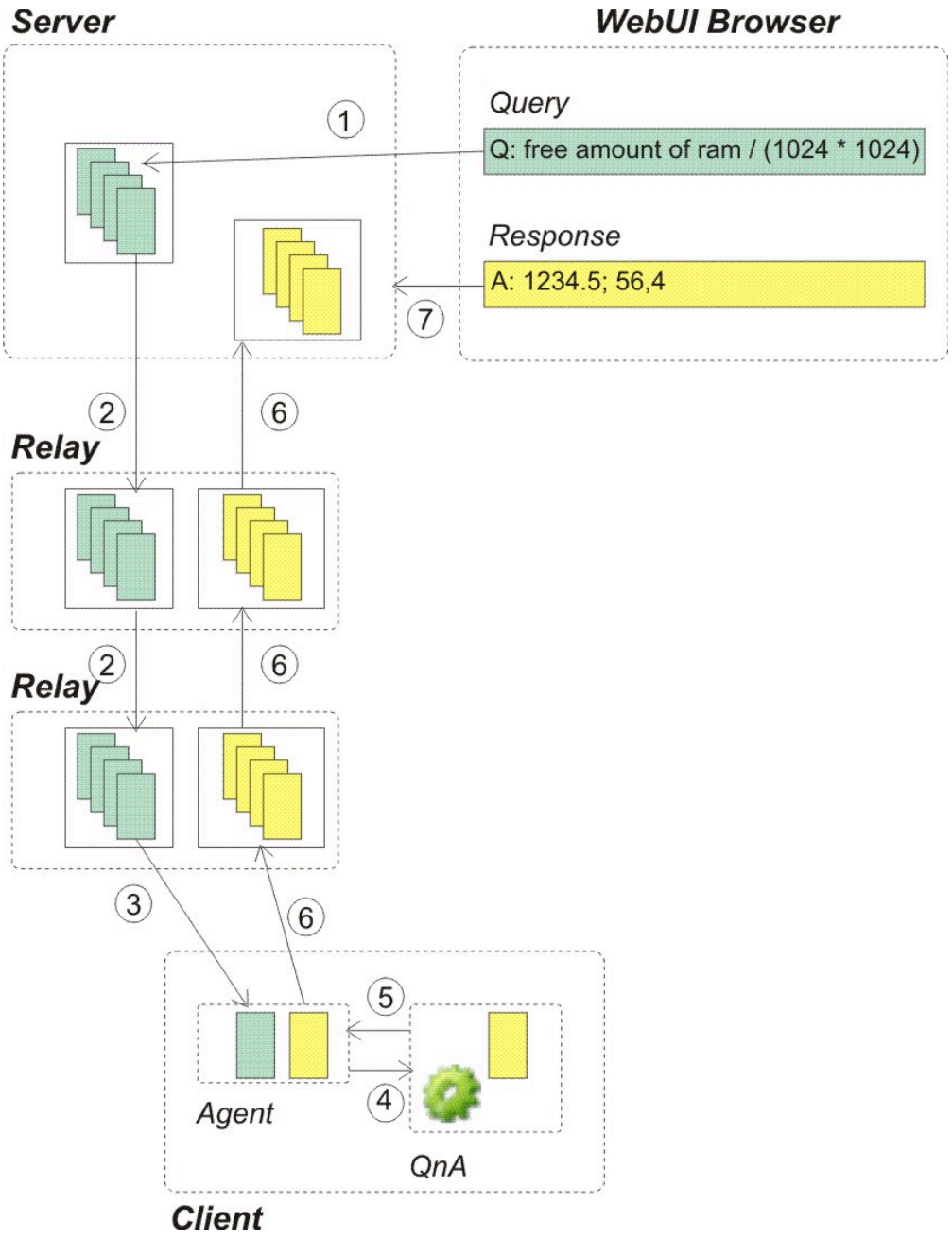


「クエリー」パネルからこの機能を使用する方法については、「WebUI の使用可能化」を参照してください。

BigFix による BigFix Query 要求の管理方法

BigFix Query 要求は、カスタマイズ可能な一連のステップで処理されます。

以下の図に、BigFix Query の内部フローを示します。各ステップには、BigFix Query の要求および応答の管理方法を調整するために、構成できる変数をリストしています。



1. WebUI にログオンしたオペレーターは、BigFix Query アプリケーションから要求を送信します。

   このステップでカスタマイズできること

   このステップをマスター・オペレーターではないオペレーターとして実行することを決定できます。この場合、オペレーター権限、または

そのオペレーターに割り当てられた役割に指定された権限のいずれかに、**「照会を送信可能 」**の値を**「はい」**に設定して必ず含めるようにしてください。

注: 照会を管理するために REST API を使用している場合は、照会を実行するオペレーターのみがその応答を参照できることに注意してください。

2. 送信された要求は、各リレー上の専用メモリー・キューを使用して、リレー階層を経由してターゲット・クライアントに伝搬されます。これにより、通常の BigFix 処理に影響を与えることなく、要求がターゲットに迅速に到達するようになります。ターゲットまたは子リレーが指定された時間内に応答しない場合は、応答するように要求されなくなります。

   このステップでカスタマイズできること

   BigFix コンソールから、サーバー用および各リレー用にメモリー・キューがクリーンアップされる方法をカスタマイズできます。

   クリーンアップ・タスクを実行する頻度。

   デフォルト値は 10 分で、設定の名前は **_BESRelay_Query_RemovalTask** です。

   要求がクリーンアップ・タスクによって削除するまでにキューに存在できる期間。

   デフォルト値は 60 分で、設定の名前は **_BESRelay_Query_MinTime** です。

   BigFix Query 要求の専用のメモリー・キューの最大サイズ。

   クリーンアップ・タスクを実行する前に、BigFix は、このメモリー・キューのサイズが、指定された最大サイズを超えているかどうか検査します。超えている場合は、クリーンアップ・タスクの実行時に、キューのサイズがしきい値内に戻るまでキュー内のエントリーが削除されます。デフォルト値は 100 MB で、設定の名前は **_BESRelay_Query_RemovalTask** です。

これらの設定について詳しくは、「クエリー ((ページ)　　) 」を参照してください。

3. 要求がターゲット・クライアントの親リレーに到達したら、そのリレーは UDP プロトコルを使用して、処理すべき新規要求があることをクライアントに通知します。そして次に、エージェントがその要求を取得します。
4. 応答するターゲットごとに、クライアントは、照会を実行して結果を返すために、照会をローカル QnA に渡します。

このステップでカスタマイズできること

BigFix コンソールから、クライアント用に以下をカスタマイズできます。

要求タイムアウトが経過する前に、QnA がマスター・オペレーターによって発行された照会を処理できる期間

デフォルト値は 60 秒で、設定の名前は **_BESClient_Query_MOMaxQueryTime** です。

要求タイムアウトが経過する前に、QnA がマスター・オペレーター以外によって発行された照会を処理できる期間

デフォルト値は 10 秒で、設定の名前は **_BESClient_Query_NMOMaxQueryTime** です。

QnA が停止するまでに、処理すべき新規照会を待機する時間。

デフォルト値は 600 秒で、設定の名前は **_BESClient_Query_IdleTimeout** です。

QnA プロセスが照会を実行することにより使用する CPU 量。

QnA が実行される時間スロットを定義することで、QnA プロセスで使用される CPU を制限できます。デフォルトでは、照会を実行する QnA は 10 ミリ秒間実行され、その後、480 ミリ秒間スリープします。これは、CPU 使用率 1 から 2 % 未満に相当します。また、この動作を定義

する設定の名前は **_BESClient_Query_WorkTime** および **_BESClient_Query_SleepTime** です。

これらの設定について詳しくは、「クエリー (（ページ）　　) 」を参照してください。

注: これらの設定は、ローカル・ユーザーとしてクライアント・システムに接続された QnA ツールを実行時には考慮されません。

5. エージェントが QnA から応答を受信すると、エージェントはその応答を含むレポートを作成し、他のレポートと同時に親リレーに配信します。
6. レポートは、リレー階層を経由してサーバーに返信されます。各リレー上で、レポートは親リレーに配信されるのを待つ間、メモリー・キューに保管されます。親リレーが使用できない場合、レポートはキューで待機し、親リレーが再び使用可能になるとすぐに配信されます。標準レポートに使用される暗号化および署名の基準と同じ基準が、これらのレポートにも適用されます。

   このステップでカスタマイズできること

   BigFix コンソールから、各リレー用に以下をカスタマイズできます。

   BigFix Query 結果の専用のメモリー・キューの最大サイズ。

   クリーンアップ・タスクを実行する前に、BigFix は、このメモリー・キューのサイズが、指定された最大サイズを超えているかどうか検査します。超えている場合は、クリーンアップ・タスクの実行時に、キューのサイズがしきい値内に戻るまでキュー内のエントリーが削除されます。デフォルト値は 100 MBで、設定の名前は **_BESRelay_Query_ResultsMemoryLimit** です。

   この設定について詳しくは、クエリー (（ページ）　　)を参照してください。
7. サーバーが結果を受信すると、サーバーは専用キューに結果を保管します。そのキューから専用の FillDB スレッドがデータを取得して、データベースに保管します。このように、BigFix サーバー上の通常処理は影響を受けません。

データベースは、指定した期間中、BigFix Query 要求とその応答の両方を保管します。これらは、例えば、フィルタリング、表示、レポート作成を行うために使用できます。適時、BigFix Query アプリケーションは、更新があるかデータベースを確認し、表示されている結果を適宜更新します。

このステップでカスタマイズできること

BigFix 管理ツールから、サーバーに関して以下をカスタマイズできます。

BigFix Query 要求を削除するまでにデータベースに保管する期間。

デフォルト値は 1440 時間 (60 日) で、この詳細オプションの名前は **queryHoursToLive** です。

BigFix Query 応答を削除するまでにデータベースに保管する期間。

デフォルト値は 4 時間で、この詳細オプションの名前は **queryResultsHoursToLive** です。

queryHoursToLive または queryResultsHoursToLive の期間が経過した要求および応答をデータベースから一度に削除する件数。

デフォルト値は 100,000 個のエントリーで、この詳細オプションの名前は **queryPurgeBatchSize** です。

これらの詳細オプションについて詳しくは、「詳細オプション (（ページ）)」を参照してください。

コンピューター設定の編集方法については、「コンピューターの設定の編集 (（ページ）)」を参照してください。

第 14 章. プラグイン・ポータル

プラグイン・ポータルは、BigFix 10 に導入された新しいコンポーネントであり、クラウド・デバイスや、BigFix に登録されている Windows 10 や MacOS のエンドポイントなどの最新デバイスを管理するのに役立ちます。最新のクライアント管理の詳細については、最新のクライアント管理の資料を参照してください。

プラグイン・ポータルは、ネットワーク上のクラウドまたは最新のデバイス用の個々のプラグインをインストールするための前提条件です。クラウド・プラグインが存在する場合にのみ機能します。

スケーラビリティーとパフォーマンスに関する考慮事項

プラグイン・ポータルは、専用インスタンスにインストールする必要があります。インスタンスは、物理または仮想にできます。メモリー、CPU、IO の要件は、エンドポイントの数と管理されるコンテンツの量に応じて変わります。

一言で言えば、BigFix 10.0.4 プラグイン・ポータルのスケールが高くなり、管理対象エンドポイントごとに消費されるリソースが少なくなります。

詳しくは、「BigFix Performance & Capacity Planning Resources」を参照してください。

BigFix 10.0.4 では、プラグイン・ポータル管理機能が、インスタンスあたり 10,000 から 50,000 のエンドポイントに増えました。これは、リソースの最適化とスケジュール管理の改善によって実現されました。

重要な最適化の 1 つに、コンテンツの削減があります。より具体的には、BigFix 10.0.4 では、不要なコンテンツのフィルタリングに役立つ以下の 2 つの機能がプラグイン・ポータルに追加されています。カスタム・サイト管理 (（ページ） 159) および サブスクライブしたサイトのコンテンツのフィルタリング (（ページ） 163)。

一般に、BigFix 10.0.4 プラグイン・ポータルのデフォルト・インストールは、優れたパフォーマンスと回復力のために最適化されています。微調整はオプションですが、パフォーマンスをさらに向上させることができます。古いバージョンからアップグレードする前に入念に『カスタム・サイト管理 (（ページ） 159)』を読むことをお勧めします。これは、特定のカスタム・サイトがサブスクライブ解除される可能性があるためです。

プラグイン・ポータルをインストールするための前提条件

プラグイン・ポータルをインストールする予定のターゲット・コンピューターに、以下をインストールする必要があります。

- BigFix エージェント、バージョン 10.0 以降。
- MongoDB バージョン 4.2 以降。

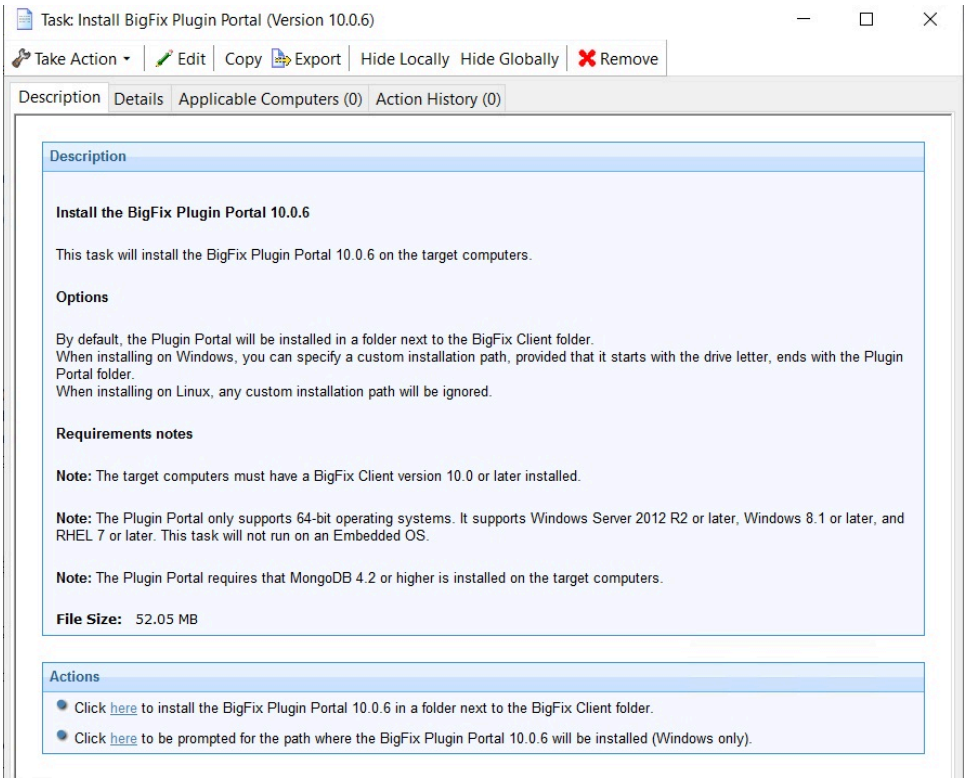
注: 強制モードで SELinux を使用した Red Hat Enterprise Linux に MongoDB をインストールする場合、追加の構成を行う必要があります。詳細については、「Red Hat または CentOS への MongoDB Community Edition のインストール」を参照してください。

注: プラグイン・ポータルをインストールしたターゲット・コンピューターに着信する HTTPS トラフィックを、TCP ポート 52311 で有効にする必要があります。この通信を妨げるルールがある場合、プラグイン・ポータルはその親リレーから更新を受信せず、プラグイン・ポータルが管理するデバイスへのアクションは完了しません。

これらのターゲット・コンピューターに BigFix リレーまたは BigFix サーバーがインストールされていないことを確認する必要があります。プラグイン・ポータルはこれらのコンポーネントと互換性がないためです。さらに、PeerNest (（ページ） 307) 機能が有効になっているコンピューターにプラグイン・ポータルをインストールすることはできません。

プラグイン・ポータルのインストール

プラグイン・ポータルを BigFix エージェントにインストールするには、BES サポート・サイトから**「BigFix プラグイン・ポータルをインストール (バージョン x)」**という名前のタスクを実行します。このタスクでは、選択したターゲットに BigFix プラグイン・ポータル・バージョン x.x をインストールします。



デフォルトでは、プラグイン・ポータルは次のディレクトリーにインストールされます。

- Windows:
  - C:\Program Files (x86)\BigFix Enterprise\BES Plugin Portal
- Linux:
  - /var/opt/BESPluginPortal
  - /opt/BESPluginPortal

注: 環境内に複数のプラグイン・ポータルがある場合がありますが、特定のターゲット・コンピューターにインストールできるプラグイン・ポータルは 1 つだけです。

BigFix 10.0.5 以降、Windows にプラグイン・ポータルをインストールする際に、ドライブ名で始まりプラグイン・ポータル・フォルダーで終わるカスタム・インストール・パスを指定できます。Linux にプラグイン・ポータルをインストールする場合、カスタム・インストール・パスは無視されます。

いくつかのクライアント設定を使用して、プラグイン・ポータルを構成できます。詳しくは、プラグイン・ポータル (（ページ）　　)を参照してください。

プラグイン・ポータルは、WebUI からもインストールできます。詳しくは、WebUI の資料を参照してください。

プラグイン・ポータルのアップグレード

前提条件:BigFix サーバー、リレー、コンソールをアップグレードします。

プラグイン・ポータルをアップグレードするには、BES サポート・サイトから「**更新済みプラグイン・ポータル - BigFix バージョン X**」という名前の Fixlet を実行します。Fixlet は、BigFix がサーバーがアップグレードされた後にのみ、エンドポイントに関連状態になります。クライアントは、BigFix サーバーがアップグレードされたことを確認してから、関連する他のアップグレード Fixlet のレポートを開始します (デフォルトでは、クライアントは登録されているサーバーのバージョンを 6 時間に 1 回チェックします)。

プラグイン・ポータルをアップグレードすると、プロキシー対象のエンドポイントが特定の期間、アクションの状態と取得されたプロパティーについてレポートできなくなる場合があります。アップグレードは、これによる IT 運用の中断が最小限のときに実行することをお勧めします。サポートが必要な場合は、HCL 技術サポートにお問い合わせください。

プラグイン・ポータルのアンインストール

エンドポイントからプラグイン・ポータルをアンインストールするには、**「TROUBLESHOOTING: BigFix プラグイン・ポータルをアンインストール」**という名前のタスクを実行します。このタスクにより、次のものが削除されます。

- プラグイン・ポータル
- 関連するクライアント設定
- プラグイン・ポータルのインストール・ディレクトリーとストレージ・ディレクトリー内のファイルとディレクトリー

また、このタスクにより、MongoDB 上のプラグイン・ポータル・データベースも削除されます。

カスタム・サイト管理

カスタム・サイトは通常、BigFix エージェントがインストールされているエンドポイントにコンテンツを配布するよう設計されています。

プラグイン・ポータルによって検出されたデバイスを、使用できないコンテンツを有するカスタム・サイトにサブスクライブすると、コンテンツを評価および管理するための不要な手間がかかり、無駄なリソース使用とパフォーマンスの低下を生み出します。

このため、BigFix 10.0.4 以降では、検出したデバイスをカスタム・サイトにサブスクライブするための追加フィルターがプラグイン・ポータルに適用されます。

BigFix 10.0.4 より前のバージョンでは、プラグイン・ポータルは、サイトのサブスクリプション条件に起因する適用性の関連度を評価した後、デバイスをカスタム・サイトにサブスクライブしていました。次のいずれかのインスペクターを使用する場合**のみ**、適用条件の関連度がプラグイン・ポータルによって評価されます。

- **エージェント・コンテキスト**
- **プロキシー・エージェント・コンテキスト**
- **プラグイン・ポータル・コンテキスト**

カスタム・サイトの適用条件が上記のインスペクターを使用 **しない** 場合、カスタム・サイトはプラグイン・ポータルによって評価されません。検出されたデバイスはカスタム・サイトをサブスクライブしません。

適用条件で少なくとも 1 つのインスペクターを使用する場合、サイトへのサブスクリプションは、以前のリリースで行われる **適用条件の関連度の評価によって異なります**。

BigFix 10.0.4 以前のバージョンについて、例を挙げて説明します。すべてのエージェントをカスタム・サイトにサブスクライブするには、次の関連式を使用します。

- **真**

「コンピューターのサブスクリプション」コンソール・タブの**「すべてのコンピューター」**選択項目に対応します。
BigFix 10.0.4 以降では、プラグイン・ポータル・エージェントも組み込む必要がある場合、上記のインスペクターのいずれかを明示的に使用する必要があります。例:

- **true またはプラグイン・ポータル・コンテキスト**

それ以外の場合、プラグイン・ポータルによって検出されたエージェントはサイトに**サブスクライブしません**。

この変更は、プラグイン・ポータルが表すエンドポイントにのみ適用され、プロキシー・エージェントによって表されるエンドポイントには適用されません。

プラグイン・ポータルによって検出されないエージェントでは、変更はありません。サイトの適用条件の関連度評価に応じてカスタム・サイトをサブスクライブします。

カスタム・サイトのコンテンツに**変更はありません**。デバイスがサイトをサブスクライブする場合、各コンテンツ・エレメント (Fixlet、タスク、分析など) の適用条件は、そのエレメントに指定された関連度にのみ依存します。

この機能は、設定で無効化できます

_BESPluginPortal_Performance_ExcludeCustomSitesSubscription = 0

プラグイン・ポータルがインストールされているクライアントにアクセスします。この設定では、プラグイン・ポータルは 10.0.4 より前のバージョンと同様に動作します。

この機能はデフォルトで使用可能です。その結果、10.0.4 より前のバージョンからプラグイン・ポータルをアップグレードすると、**プラグイン・ポータル・エージェントが特定のカスタム・サイトからサブスクライブを取り消される可能性があります**。この動作を回避する場合は、アップグレードする前にこの機能を無効にする必要があります。BigFix コンソールを使用してアップグレードする場合、Fixlet の**「更新されたプラグイン・ポータル - BigFix バージョン X」**にも、このプラグインを無効にするオプションが用意されています。このオプションは BigFix WebUI を使用した場合、利用できません。

カスタム・サイトの適用条件の関連度の例を以下に示します。

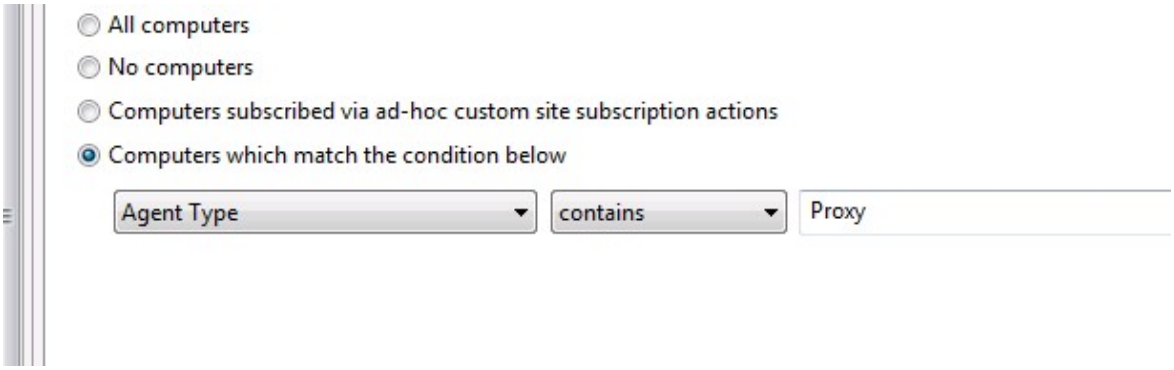
- プラグイン・ポータル・エージェントのみをサブスクライブするには:
  - **プラグイン・ポータル・コンテキスト**
- ネイティブ・エージェントのみをサブスクライブするには、次を行います。
  - **エージェント・コンテキスト**
- プラグイン・ポータル以外のすべてのエージェントをサブスクライブするには:
  - **真**

- すべてのエージェントをサブスクライブするには、次のいずれかを使用します。
    - **true またはプラグイン・ポータル・コンテキスト**
    - **true またはプロキシー・エージェント・コンテキスト**
- プラグイン・ポータルまたは**プロキシー済み**のエージェントによって検出されたプロキシー・エージェントのみをサブスクライブするには、以下のいずれかを使用します。
    - **プロキシー・エージェント・コンテキスト**
    - **非エージェント・コンテキスト**
- 名前に「dept1」が含まれる (大文字と小文字を区別しない) すべてのプラグイン・ポータル・エージェントをサブスクライブするには:
    - **プラグイン・ポータル・コンテキストで存在し (コンピューター名)、その (小文字のストリングとして「dept1」が含まれる)**
- Red Hat OS にインストールされているすべてのプロキシー・エージェントをサブスクライブするには:
    - **プロキシー・エージェント・コンテキスト内に存在する (オペレーティング・システム) (小文字の名前に「red hat」が含まれる)**
- Red Hat OS にインストールされているすべてのエージェント (プラグイン・ポータルを除く) をサブスクライブするには:
    - **存在する (オペレーティング・システム) その (小文字としての名前に「red hat」が含まれる)**

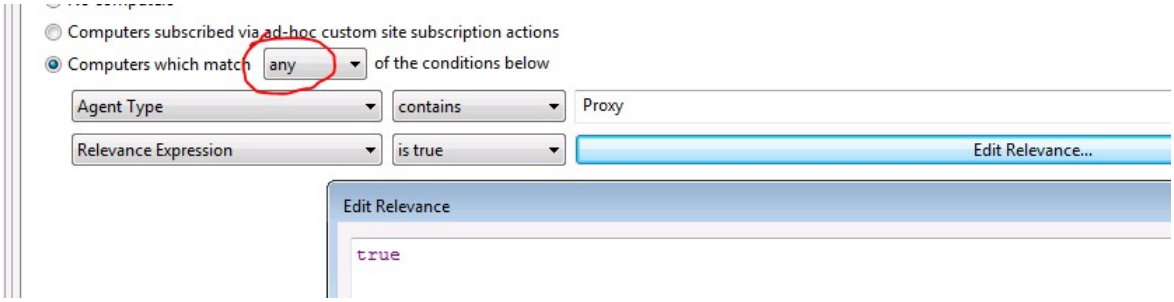
「コンピューターのサブスクリプション ・コンソール」タブの使用例をいくつか示します。

コンソールは**エージェント・タイプ**条件を**プロキシー・エージェント**・インスペクターを使用する式に分解するため、この条件を設定するだけで、プラグイン・ポータルに無視されなくなります。

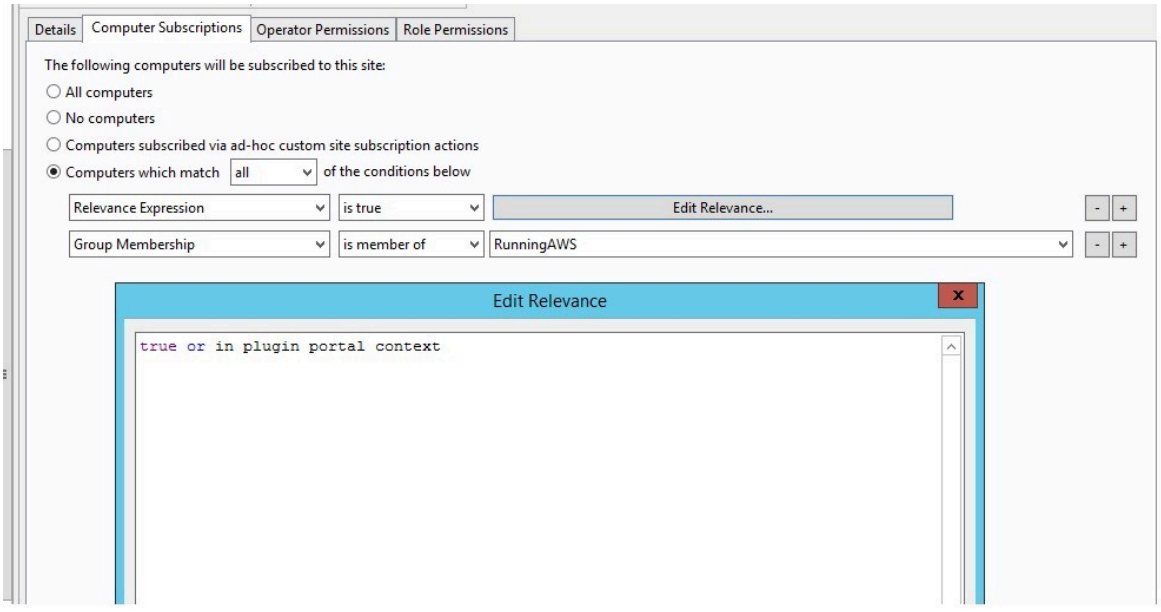
- プロキシー・エージェントのみをサブスクライブするには:



- すべてのエージェントをサブスクライブするには、次を行います。



- 指定されたサーバー・ベースのコンピューター・グループのすべてのエージェントをサブスクライブするには:



サブスクライブしたサイトのコンテンツのフィルタリング

BigFix 10.0.4 以降ではプラグイン・ポータルの負荷を軽減するために、プラグイン・ポータルに新しい機能を追加しました。検出したデバイスで不要なコンテンツを管理しないようにするためです。

コンテンツの削減は最初に実行し、パフォーマンスの向上、操作の高速化、コストの削減を行う必要があります。以下の手順に従って、完全に除外できないすべての外部サイトについて、プラグイン・ポータルが検出したデバイスに不要な Fixlet および分析を除外することを強くお勧めします。これを行うには、適切なサブスクリプション関連度を設定するか、カスタム・サイト管理 (（ページ） 159) で記述されているフィルターを渡してカスタム・サイトを設定します。

プラグイン・ポータルは不要な関連度の評価するよう強制されないため、以下で説明するコンテンツ・フィルタリング機能は、不要な Fixlet、アクション、または分析の関連度を変更するよりも簡単で効率的です。

サイトがダウンロードされた後、プラグイン・ポータルは収集時に一部の Fixlet または分析を評価プロセスから除外できます。除外されたコンテンツを `PluginPortalSubscriptionOptions.json` という名前の JSON ファイルに指定し、それをサイト自体に組み込むだけです。

JSON ファイルは、以下の 3 つの配列で構成されます。

- **ExcludedFiles**: 無視する **FXF 名** のリストを定義するストリングの配列。
- **ExcludedFixletIDs**: 無視される、**Fixlet/タスク ID** のリストを定義する整数の配列。
- **ExcludedAnalysisIDs**: 無視される、**分析 ID** のリストを定義する整数の配列。

JSON ファイルの例:

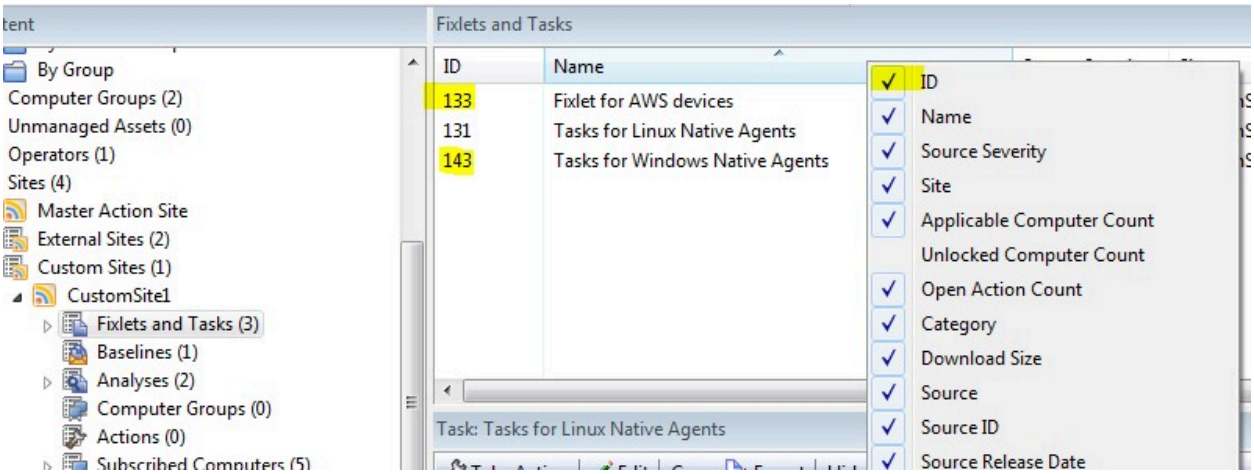
```
{
  "ExcludedFiles": [
   "My Windows analyses.fxf",
   "Department 121.fxf"
  ],
  "ExcludedFixletIDs": [
```

```
   209, 31, 405
  ],
  "ExcludedAnalysisIDs": [
   211, 33
  ]
 }
```

ファイル名を指定すると、そのファイルのすべてのコンテンツは、評価フェーズ中にポータル・プラグインに無視されます。プロキシー・デバイスによって消費される必要がある**カスタム・サイト**や FXF 定義コンテンツなど、ファイル名を指定できない場合は、無視する Fixlet、タスク、または分析 ID のリストを指定できます。

カスタム・サイトの使用例を以下に示します。

除外するコンテンツの ID を取得するには、カスタム・サイトの **「Fixlet とタスク」**BigFix コンソール・パネルを開き、「ID」列を確認します。列が表示されない場合は、列名を右クリックして ID項目を確認し、対応する列を表示します。



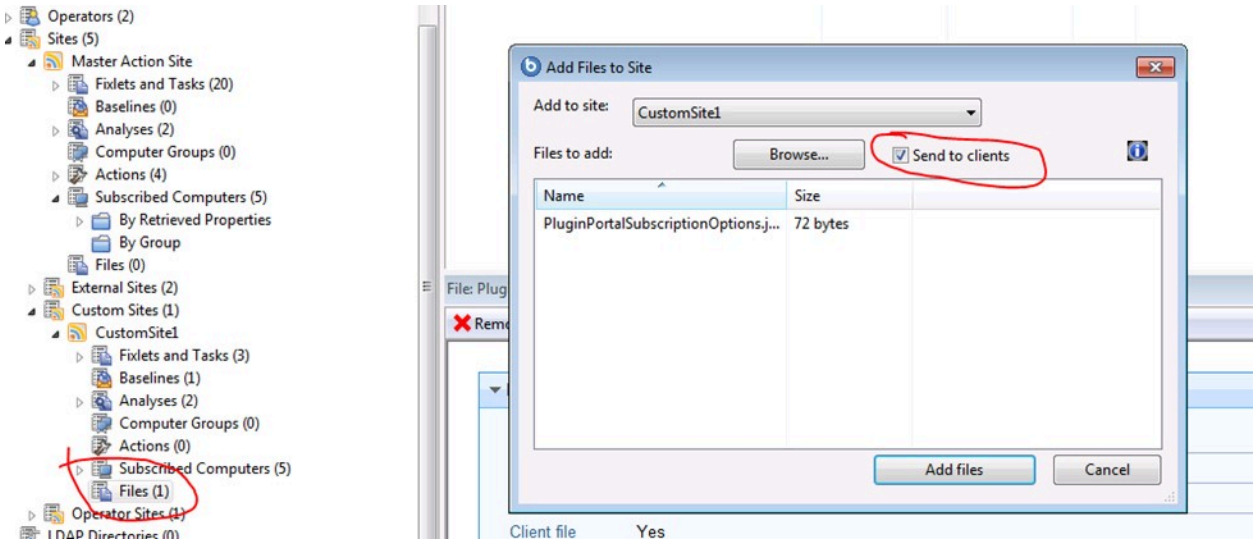
「分析」パネルでも同じ操作を行 います。

ID を取得したら、BigFix コンソール・マシン内の任意の場所に JSON ファイルを作成します。

```
{
  "ExcludedFixletIDs": [
   133, 143
  ],
```

```
 "ExcludedAnalysisIDs": [
  211, 33
 ]
}
```

その後、JSON ファイルをカスタム・サイトに追加できます。**「クライアントに送信」**オプションにチェック・マークを付ける必要があります。



ファイルを追加 するボタンをクリックすると、プラグイン・ポータルによって検出されたコンピューターを含むすべてのサブスクライブ済みコンピューターに、新規バージョンのカスタム・サイトが配布されます。

BES プラグイン・ポータルのプラグインを構成するためのアクション・コマンド

プラグイン・ポータルのプラグインは、PluginStore sqlite データベースに設定を保管することで構成できます。これは、タスク用に特別に設計されたいくつかのアクション・コマンドを使用して簡単に実行できます。

概要

アクション・スクリプトを使用して PluginStore データベースを操作するためのコマンド構文は、以下のとおりです。

```
plugin store "<plugin name>" set "<plugin store key>" value "<setting
 value>" on "{now}"
plugin store "<plugin name>" set encrypted "<plugin store key>" value
 "<setting value>" on "{now}"
plugin store "<plugin name>" delete "<plugin store key>"
plugin store "<plugin name>" multiple set "<percent encoded json>" on
 "{now}"
plugin store "<plugin name>" multiple set encrypted "<percent encoded
 json>" on "{now}"
```

plugin store コマンドは、`VMwareAssetDiscoveryPlugin` などのプラグインの名前を受け入れます。

set キーワードの後に続く **plugin store key** は、サブセクションと呼ばれる一連の語句のように下線で区切られた表示になっており、`Log_Path` のようになります。

value キーワードの後に続くパラメーターは、選択した設定に適した値にできます。

コマンドの表示例を以下に示します。

```
plugin store "AWSAssetDiscoveryPlugin" set "Log_Verbose" value "0" on
 "{parameter "action issue date" of action}"
```

参考までに、PluginStore データベース・スキーマを以下のとおりにすることを検討してください。

キー	「値」	開始日
TEXT NOT NULL PRIMARY KEY	KEY TEXT NOT NULL	DATE

plugin store コマンドについて詳しくは、『plugin store』を参照してください。

クラウド・プラグインの構成

Amazon Web Services、Microsoft Azure、Google Cloud Platform、および VMware クラウド環境を管理するために、いくつかのクラウド・プラグインをプラグイン・ポータルにインストールできます。

プラグインを構成するすべてのプラグイン・ストア・コマンドは、**plugin store** コマンドで開始する必要があります。PluginStore データベースを操作するためのコマンド構文について詳しくは、『概要 (（ページ） 165)』を参照してください。

クラウド・プラグインの構成の「plugin store」キーワードの後に必要なプラグイン名は、以下のとおりです。

- **AWSAssetDiscoveryPlugin**

- **AzureAssetDiscoveryPlugin**

- **GCPAssetDiscoveryPlugin**

- **VMwareAssetDiscoveryPlugin**

この情報はプラグイン・ストアの設定を構成する上で重要です。プラグイン・ストアのアクション・コマンドは、設定キーを正しく構築するためにこれらの名前に依存するためです。

plugin store コマンドで受け入れられるオプションは、**set**、**multiple set**、**delete** です。**set** キーワードと **multiple set** キーワードがキーワード **encrypted** で装飾されている場合、データベース内で暗号化された値になります。

set オプションの後には、設定するプラグイン・ストア・キーが続き、その後に **value** キーワードと、保存する値自体が続く必要があります。最後に、現在の日付が **on** キーワードの後に指定されます。

以下に例を示します。

plugin store "AWSAssetDiscoveryPlugin" **set** "Credentials_AccessKey_myLabel" **value** "myAccessKey" **on** "{parameter "action issue date" of action}"

pluginstore "AWSAssetDiscoveryPlugin" **set encrypted** "Credentials_SecretAccessKey_myLabel" **value** "mySecret" **on** "{parameter "action issue date" of action}"

プラグイン・ストアには、以下の設定が表示されます。

キー	「値」	開始日
_AWSAssetDiscoveryPlugin_Credentials_AccessKey_myLabel	myAccessKey	0123456789

_AWSAssetDiscoveryPlugin_Credentials_AccessKey_myLabel	{b_SecretAccessKey_myLabel	0123456789

multiple set オプションは、複数の設定を一度にすばやく構成するために使用されます。これに対して、**encrypted** オプションも使用できます。**multiple set** オプションの後には、データベースに追加する必要があるキーと値のペアのリストを含む、パーセント・エンコーディングされた JSON が続く必要があります。デコードされた JSON の例を以下に示します。

```
{
    "Credentials_AccessKey_myLabel" : "myAccessKey",
    "Credentials_Region_myLabel" : "myLabelRegion",
    "HTTP_ProxyURL" : "myProxyURL",
    "HTTP_ProxyUser" : "myProxyUser"
}
```

コマンドの出力例を以下に示します。

plugin store "AWSAssetDiscoveryPlugin" **multiple set** <example json> **on** "{parameter "action issue date" of action}"

プラグイン・ストアに以下の設定を追加する必要があります。

キー	「値」	開始日
_AWSAssetDiscoveryPlugin_Credentials_AccessKey_myLabel	myAccessKey	0123456789
_AWSAssetDiscoveryPlugin_Credentials_Region_myLabel	myLabelRegion	0123456789
_AWSAssetDiscoveryPlugin_HTTP_ProxyURL	myProxyURL	0123456789
_AWSAssetDiscoveryPlugin_HTTP_ProxyUser	myProxyUser	0123456789

set コマンドでは「on」キーワードが必要であり、その後に設定が発行される日付が続く必要があります。

[...] **on** "{parameter "action issue date" of action}"

delete オプションは、プラグイン・ストアから特定のキーを削除するだけです。

plugin store "AWSAssetDiscoveryPlugin" **delete** "Credentials_Region_myLabel"

キー	「値」	開始日
_AWSAssetDiscoveryPlugin_Credentials_AccessKey_myLabel	myAccessKey	0123456789
_AWSAssetDiscoveryPlugin_HTTP_Proxy_myLabel	myProxyURL	0123456789
_AWSAssetDiscoveryPlugin_HTTP_Proxy_myLabel	myProxyUser	0123456789

delete コマンドの後にキーワード **all** を発行すると、指定されたプラグインのすべてのプラグイン・ストア設定が削除されます。

plugin store "AWSAssetDiscoveryPlugin" **delete all**

プラグイン設定の多くは、特定の資格情報セットにバインドされています。この資格情報セットは、資格情報ラベルと呼ばれるものによって識別されます。そのため、アクション・スクリプトを作成する場合は、アクション・スクリプトでラベルをアクション・パラメーターに保管して、後で複数回再利用できるようにすることをお勧めします。

```
parameter "credentialsLabel" = "<my label>"
```

注: 置換する必要がある値は、<my label> のように、不等号括弧で囲まれた値のみです。

キーまたはキーの一部を複数回再解釈する必要がある場合は、パラメーターにも保管でき、後で連結することもできます。

```
parameter "accessKey" = "Credentials_AccessKey"
parameter "secretAccessKey" = "Credentials_SecretAccessKey"
```

例えば、新しいパラメーターは、前のパラメーターを連結して定義できます。

```
parameter "credentialsLabel" = "<my label>"
```

プラグインをプログラムで構成する場合は、パラメーターが存在するかどうか、またはパラメーターが空でないかどうかを確認すると便利です。これを実現するには、コードを if ブロックで囲みます。

```
if {(exists parameter "myParam")AND (parameter "myParam" != "")}
 // my code
endif
```

共通のプラグイン設定

以下の設定は、すべてのクラウド・プラグインに共通です。

Discovery_Region - プラグインのデフォルト・リージョン。このリージョンは、プラグインに保存されている資格情報に関連するすべての AWS アカウントで有効なリージョンのリストを取得するために使用されます。この設定は必須です。

Log_Path - プラグインのログのパス。

Log_Verbose - 1 に設定すると、デバッグ・ロギングが有効になります。0 に設定すると、情報ロギングのみが表示されます。

JSONファイル設定 - 一部の設定は、settings.json という JSON を使用してクラウド・プラグインに定義されます。このような JSON の例を以下に示します。

```
{
    "ID": <plugin name>,
    "ConfigurationOptions": "",
    "DeviceReportRefreshIntervalMinutes": <refresh interval in minutes>,
    "DeviceReportExpirationIntervalHours": 168,
    "CommandFormat": "JSON",
    "SendSettingsToPlugin": [],
    "ExecutablePath": <executable path>,
    "HandlePartialRefresh": false,
    "FullReportsInRefreshAll": true,
    "NoRefreshBeforeActionIntervalMinutes": 60
}
```

AWSAssetDiscoveryPlugin 構成

Amazon Web Services プラグインを完全に構成するために必要な設定を以下に示します。

IAM ユーザー固有の設定

Credentials_AccessKey_<label> - IAM ユーザーに関連付けられたアクセス・キー ID。この設定は必須です。

Credentials_SecretAccessKey_<label> - IAM ユーザーに関連付けられたシークレット・アクセス・キー。この設定の値は暗号化する必要があります。この設定は必須です。

Credentials_Region_<label> - ラベル <label> を持つ IAM ユーザー資格情報のデフォルト・リージョン。このリージョンは、Discovery Region をオーバーライドします。

Credentials_Roles_<label>_<arn> - ラベル <label> を持つ IAM ユーザーによって引き受ける ARN <arn> を持つロールのリージョン。このリージョンは、Credentials Region と Discovery Region の両方をオーバーライドします。この値は空にできます。

Credentials_Roles_ExternalId_<label>_<arn> - ラベル <label> を持つ IAM ユーザーによって引き受ける ARN <arn> を持つロールの外部ID。この値は暗号化する必要があります。IAM ロールに外部 ID が必要ない場合は、この設定を省略できます。

詳細設定

HTTP_ProxyURL - プラグインの HTTP プロキシーのURL。

HTTP_ProxyUser - プラグインの HTTP プロキシーのユーザー。

HTTP_ProxyPassword - プラグインの HTTP プロキシーのパスワード。この設定の値は暗号化する必要があります。

RegionAllowedList_<label> - ラベルが <label> のユーザーに対して、リストされたリージョンでのみプラグインが検出を実行するように強制します。個々のリージョンをセミコロン「;」で区切って指定します。

例: `eu-central-1;eu-west-1;us-east-1`

AWS プラグインのインストール時に、許可されるリージョンを指定できます。AWS リージョンを制限する方法の詳細については、『Limit AWS Regions to restrict the scope of device discovery』を参照してください。

AWSAssetDiscoveryPlugin 構成の例

一部のパラメーターの初期化:

```
parameter "firstLabel" = "foo"
parameter "secondLabel" = "bar"
```

```
parameter "accessKey" = "Credentials_AccessKey"
parameter "secretAccessKey" = "Credentials_SecretAccessKey"
```

注: foo と bar は、架空のラベルの名前です。ただし、**Credentials_AccessKey** と **Credentials_SecretAccessKey** は実際の設定名です。これらの 4 つのパラメーターを定義します。これらのパラメーターを組み合わせることにより、上記のようにユーザー・キーとパスワードを設定するために必要なキーを定義できるためです。

プラグインのデフォルト・リージョンの設定:

```
plugin store "AWSAssetDiscoveryPlugin" set "Discovery_Region" value
 "eu-west-1" on "{parameter "action issue date" of action}"
```

最初のユーザーの構成:

```
parameter "firstUserAccessKey" = "{parameter "accessKey"}_{parameter
 "firstLabel"}"
parameter "firstUserPassword" = "{parameter "secretAccessKey"}_{parameter
 "firstLabel"}"
plugin store "AWSAssetDiscoveryPlugin" set "{parameter
 "firstUserAccessKey"}" value "<myUserKey1>" on "{parameter "action issue
 date" of action}"
plugin store "AWSAssetDiscoveryPlugin" set encrypted "{parameter
 "firstUserPassword} value "<myUserPass1>" on "{parameter "action issue
 date" of action}"
plugin store "AWSAssetDiscoveryPlugin" set "Credentials_Region_{parameter
 "firstLabel"}" value "eu-central-1" on "{parameter "action issue date" of
 action}"
```

注: プラグイン・ストア・キーを完全に定義するために、最初の 4 つのパラメーターを組み合わせて 2 つの新しいパラメーターをペアで連結しました。パラメーターを連結するには、下線で区切られた 2 つのパラメーターで構成され

た文字列を新しいパラメーターに割り当てます。したがって、パラメーター「firstUserAccessKey」の内容は「Credentials_AccessKey_foo」になります。

2 番目のユーザーの構成:

```
parameter "secondUserAccessKey" = "{parameter "accessKey"}_{parameter
 "secondLabel"}"
parameter "secondUserPassword" = "{parameter "secretAccessKey"}_{parameter
 "secondLabel"}"
plugin store "AWSAssetDiscoveryPlugin" set "{parameter
 "secondUserAccessKey"}" value "<myUserKey2>" on "{parameter "action issue
 date" of action}"
plugin store "AWSAssetDiscoveryPlugin" set encrypted "{parameter
 "secondUserPassword} value "<myUserPass2>" on "{parameter "action issue
 date" of action}"
plugin store "AWSAssetDiscoveryPlugin" set "Credentials_Roles_{parameter
 "secondLabel"}_fakeRoleARN1" value "us-east-1" on "{parameter "action
 issue date" of action}"
plugin store "AWSAssetDiscoveryPlugin" set "Credentials_Roles_{parameter
 "secondLabel"}_fakeRoleARN2" value "us-west-1" on "{parameter "action
 issue date" of action}"
plugin store "AWSAssetDiscoveryPlugin" set encrypted
 "Credentials_Roles_ExternalId_{parameter
 "secondLabel"}_fakeRoleARN2" value "myExternalId" on "{parameter "action
 issue date" of action}"
```

ログを verbose に設定します。

```
plugin store "AWSAssetDiscoveryPlugin" set "Log_Verbose" value "1" on
 "{parameter "action issue date" of action}"
```

許可されるリージョンのリストの設定:

```
plugin store "AWSAssetDiscoveryPlugin" set "RegionAllowedList_{parameter
 "secondLabel"}" value "us-east-1;us-west-1" on "{parameter "action issue
 date" of action}"
```

AWS プラグインのインストール時に、許可されるリージョンを指定できます。AWS リージョンを制限する方法の詳細については、『Limit AWS Regions to restrict the scope of device discovery』を参照してください。

PluginStore で予期される出力の例を以下に示します。

キー	「値」
_AWSAssetDiscoveryPlugin_Credentials_AccessKey_foo	myUs
_AWSAssetDiscoveryPlugin_Credentials_SecretAccessKey_foo	{obf}A
_AWSAssetDiscoveryPlugin_Credentials_Region_foo	eu-cer
_AWSAssetDiscoveryPlugin_Credentials_AccessKey_bar	myUs
_AWSAssetDiscoveryPlugin_Credentials_SecretAccessKey_bar	{obf}A
_AWSAssetDiscoveryPlugin_Credentials_Roles_bar_fakeRoleARN1	us-eas
_AWSAssetDiscoveryPlugin_Credentials_Roles_bar_fakeRoleARN2	us-we
_AWSAssetDiscoveryPlugin_Credentials_Roles_ExternalId_bar_fakeRoleARN2	{obf}A
_AWSAssetDiscoveryPlugin_Discovery_Region	eu-we
_AWSAssetDiscoveryPlugin_Log_Verbose	1
_AWSAssetDiscoveryPlugin_RegionAllowedList_bar	us-eas

AzureAssetDiscoveryPlugin 構成

Microsoft Azure プラグインを完全に構成するために必要な設定を以下に示します。

IAM ユーザー固有の設定

Credentials_ClientID_<label> - ラベル <label> を持つユーザーのクライアント ID。

Credentials_ClientSecret_<label> - ラベル <label> を持つユーザーのクライアント秘密鍵。

Credentials_SubscriptionID_<label> - ラベル <label> を持つユーザーのサブスクリプション ID。

Credentials_TenantID_<label> - ラベル <label> を持つユーザーのテナント ID。

AzureAssetDiscoveryPlugin 構成の例

```
parameter "myLabel" = "foo"
plugin store "AzureAssetDiscoveryPlugin" set
 "Credentials_TenantID_{parameter "myLabel"}" value "myTenantID" on
 "{parameter "action issue date" of action}"
plugin store "AzureAssetDiscoveryPlugin" set
 "Credentials_ClientID_{parameter "myLabel"}" value "myClientID" on
 "{parameter "action issue date" of action}"
plugin store "AzureAssetDiscoveryPlugin" set encrypted
 "Credentials_ClientSecret_{parameter "myLabel"}" value "myClientSecret" on
 "{parameter "action issue date" of action}"
plugin store "AzureAssetDiscoveryPlugin" set
 "Credentials_SubscriptionID_{parameter "myLabel"}" value
 "mySubscriptionID" on "{parameter "action issue date" of action}"
```

各表記の意味は次のとおりです。

myTenantID

ユーザーのテナント ID。

myClientID

クライアントのユーザー ID。

myClientSecret

ユーザーのクライアント秘密鍵。

mySubscriptionID

ユーザーのサブスクリプション ID。

GCPAssetDiscoveryPlugin 構成

Google Cloud Platform プラグインを完全に構成するために必要な設定を以下に示します。

サービス・アカウント固有の設定

Credentials_JSON_<label> - GCP 上のプロジェクトのサービス・アカウント・メンバーに関連する暗号化された JSON キー。

GCP JSON キー・ファイルは、次のようになります。

```
{
  "type": "service_account",
  "project_id": "test-123456",
  "private_key_id": "0123456789abcdefghilmnopqrstuvz",
  "private_key": "-----BEGIN PRIVATE KEY-----\naVeryLongKey\n-----END
 PRIVATE KEY-----\n",
  "client_email": "testusersvc@test-123456.iam.gserviceaccount.com",
  "client_id": "01234567891011121314",
  "auth_uri": "https://accounts.google.com/o/oauth2/auth",
  "token_uri": "https://oauth2.googleapis.com/token",
  "auth_provider_x509_cert_url":
 "https://www.googleapis.com/oauth2/v1/certs",
  "client_x509_cert_url":
 "https://www.googleapis.com/robot/v1/metadata/x509/
  testusersvc%40test-123456.iam.gserviceaccount.com"
}
```

JSON キーは、plugin store アクション・コマンドに送る前にパーセント・エンコーディングされている必要があります。GCP プラグインへのサービス・アカウントの構成に必要なすべての情報は JSON に含まれているため、挿入や削除を行うための唯一の設定です。

JSON には秘密鍵が含まれているため、暗号化する必要があります。

GCPAssetDiscoveryPlugin 構成の例

```
parameter "jsonKey" = "<percent encoded json>"
plugin store "GCPAssetDiscoveryPlugin" set encrypted
 "Credentials_JSON_foo" value "{parameter "jsonKey"}" on "{parameter
 "action issue date" of action}"
```

VMWareAssetDiscoveryPlugin 構成

VMware プラグインを完全に構成するために必要な設定を以下に示します。

IAM ユーザー固有の設定

Credentials_Username_<label> - ラベル <label> を持つユーザーのユーザー名。

Credentials_Password_<label> - ラベル <label> を持つユーザーの暗号化されたパスワード。

Credentials_URL_<label> - ラベル <label> を持つユーザーの資格情報ラベル。

VMwareAssetDiscoveryPlugin 構成の例

```
parameter "myLabel" = "foo"
plugin store "VMWareAssetDiscoveryPlugin" set
 "Credentials_Username_{parameter "myLabel"}" value "myUsername" on
 "{parameter "action issue date" of action}"
plugin store "VMWareAssetDiscoveryPlugin" set "Credentials_URL_{parameter
 "myLabel"}" value "myURL" on "{parameter "action issue date" of action}"
plugin store "VMWareAssetDiscoveryPlugin" set encrypted
 "Credentials_Password_{parameter "myLabel"}" value "myPassword" on
 "{parameter "action issue date" of action}"
```

各表記の意味は次のとおりです。

myUsername

ユーザーのユーザー名。

myURL

ユーザーの資格情報ラベル。

myPassword

ユーザーの暗号化されたパスワード。

REST API を使用した BESPluginPortal プラグインの構成

プラグイン構成は、REST API を使用して BigFix アクションを発行することで実行できます。

残りの API に渡されるデータは BES XML ファイルでなければなりません。

必要な構成用のコマンドを含むアクション・スクリプトを作成し、それに応じて BES XML のフィールドにアクションを入力するだけで十分です。このような XML の例を以下に示します。

```
<?xml version="1.0" encoding="utf-8"?>
<BES xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd=
"http://www.w3.org/2001/XMLSchema" SkipUI="true">
    <SingleAction>
        <Title>test action</Title>
        <Relevance>true</Relevance>
        <ActionScript>
            plugin store "<plugin name>" set "<setting key>" value
 "<setting value>" on "{parameter "action issue date" of action}"
        </ActionScript>
        <SuccessCriteria />
        <Settings />
        <SettingsLocks />
        <Target>
            <ComputerID>0123456789</ComputerID>
        </Target>
    </SingleAction>
</BES>
```

ActionScript エレメントには、アクション・スクリプト・テキストが含まれます。

このアクションのターゲットは、構成するプラグインがインストールされている BESPluginPortal マシンのネイティブ表現である必要があります。したがって、Target

フィールド内の ComputerID エレメントを使用し、その値に適切なコンピューター IDを入力すると便利です。

XML ファイルまたは XML ストリングをコードで作成した後、XML に記述されているアクションを POST を実行して発行できます。

次のコマンドは、cURL を使用して POST を実行します。ファイル action.xml は、ポート <port> (通常、BigFix の場合は 52311) を介してサーバー <server> に通知され、action.xml に含まれるアクションを作成して実行します。

```
curl --request POST --data-binary @action.xml --user <username>:<password>
 https://<server>:<port>/api/actions
```

```
curl -X POST -d @action.xml -ku <username>:<password>
 https://<server>:<port>/api/actions
```

Action REST API リソースについて詳しくは、『Action 』または BES.xsd および BESAPI.xsd ファイルを参照してください。

または、 IEM CLI を使用して BigFix REST API に対する要求を簡単に発行することもできます。

これを行うには、まず IEM CLI でセッションを開きます。端末を開き、BigFix Server フォルダーで IEM CLI を見つけます。

その後、次のコマンドを実行します。

```
iem login --server=<server>:<port> --user=<user> --password=<password>
```

その後、セッションが開いたら、次のコマンドを使用して要求を発行するだけです。

```
iem POST <path to action.xml> /api/actions
```

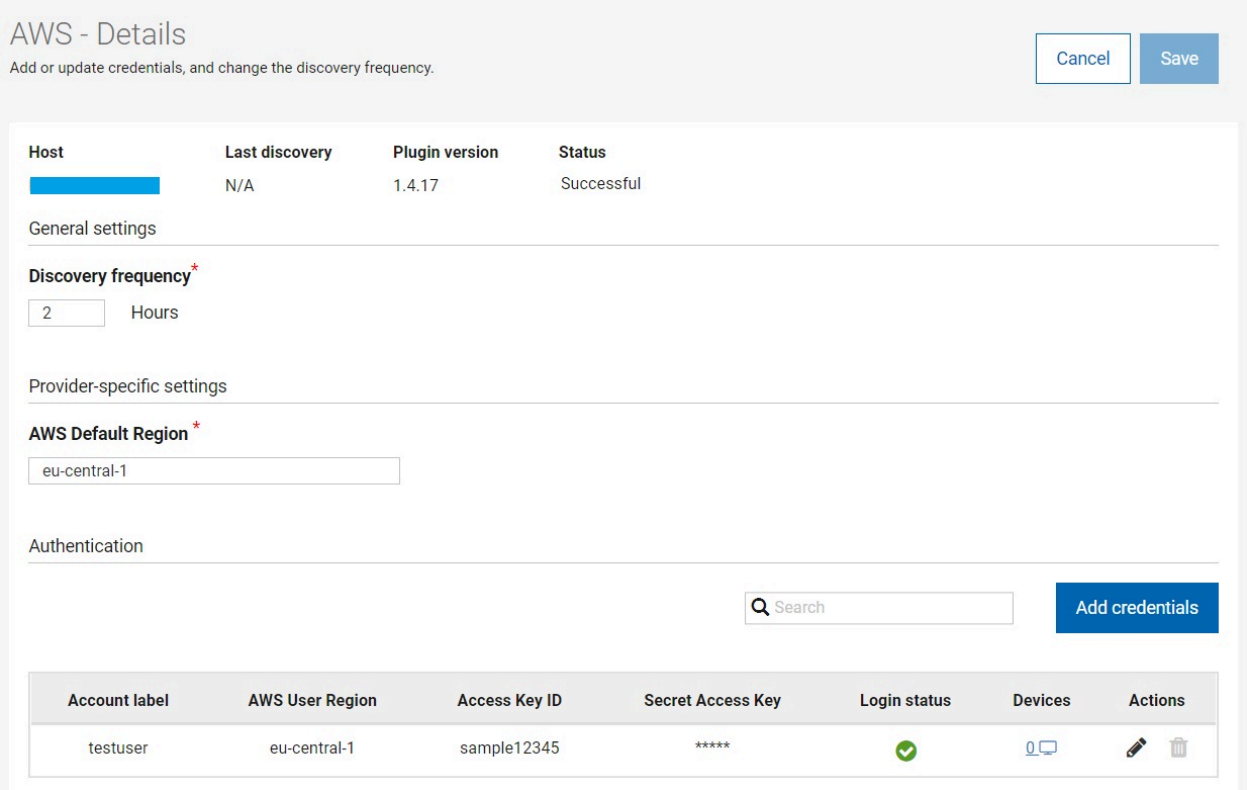
詳しくは、『 IEM CLI からの要求の実行 』を参照してください。

例 1: AWS プラグインの構成

この例では、BigFix REST API を介してディスパッチされた単一アクションを使用して AWS Asset Discovery プラグインを構成します。

AWS プラグインを既にインストールしており、ユーザーを既に構成済みであるが、多数の他のユーザーをすばやく構成したいとします。

以下の図に示すように、ログインを正しく実行した **testuser** を使用してプラグインがインストールされ、ステータスが「Successful」であることが分かります。



次に、構成の更新に使用される XML ファイルを作成します。このような XML ファイルは、REST API コマンドによってデータ・ファイルとして渡されます。

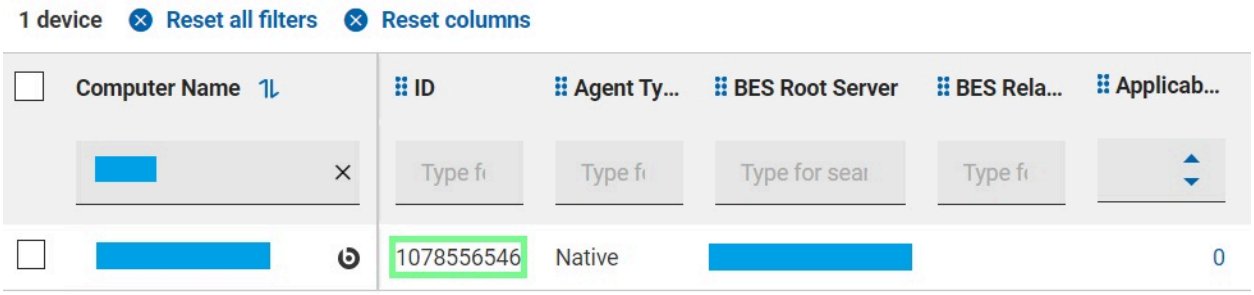
詳しくは、『 REST API を使用した BESPluginPortal プラグインの構成 (（ページ）178)』ページと『 BigFix REST API 』ページを参照してください。

単純なアクションには、以下のテンプレートを使用します。

```
<?xml version="1.0" encoding="utf-8"?>
<BES xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema" SkipUI="true">
    <SingleAction>
```

```
        <Title>AWS Plugin Config Template</Title>
        <Relevance>true</Relevance>
        <ActionScript>
            <!--Your action script-->
        </ActionScript>
        <SuccessCriteria />
        <Settings />
        <SettingsLocks />
        <Target>
            <ComputerID><!--Your ComputerID--></ComputerID>
        </Target>
    </SingleAction>
</BES>
```

XML ファイルのコメントから分かるように、少なくとも 2 つのフィールドに入力する必要があります。まず、 **ComputerID** (AWSAssetDiscoveryPlugin がインストールされているポータル・マシンの ComputerID) を見つけます。



次に、構成のアクション・スクリプトを作成します。2 人のユーザー (つまり **testuser2** と **testuser3**) を構成するとします。これらの 2 人のユーザーは、それぞれ **us-east-1** と **af-south-1** を自分のリージョンとして持っており、デフォルトのプラグイン・リージョンよりも優先されます。

『クラウド・プラグインの構成 (*（ページ） 166)*』で説明したように、便利なキーとデータをアクション・パラメーターに保管します。

```
parameter "firstLabel" = "testuser2"
parameter "secondLabel" = "testuser3"

parameter "accessKey" = "Credentials_AccessKey"
parameter "secretAccessKey" = "Credentials_SecretAccessKey"
parameter "region" = "Credentials_Region"
```

次に、plugin store 設定を構成するコマンドを記述します。

注: 置換する必要がある値は、<myUserKey2> のように、不等号括弧で囲まれた値のみです。

```
parameter "firstAccessKey" = "{parameter "accessKey"}_{parameter
 "firstLabel"}"
parameter "firstPassword" = "{parameter "secretAccessKey"}_{parameter
 "firstLabel"}"
plugin store "AWSAssetDiscoveryPlugin" set "{parameter
 "firstAccessKey"}" value "<myUserKey2>" on "{parameter "action issue date"
 of action}"
plugin store "AWSAssetDiscoveryPlugin" set encrypted "{parameter
 "firstPassword"}" value "<myUserPass2>" on "{parameter "action issue date"
 of action}"
plugin store "AWSAssetDiscoveryPlugin" set "Credentials_Region_{parameter
 "firstLabel"}" value "us-east-1" on "{parameter "action issue date" of
 action}"

parameter "secondAccessKey" = "{parameter "accessKey"}_{parameter
 "secondLabel"}"
parameter "secondPassword" = "{parameter "secretAccessKey"}_{parameter
 "secondLabel"}"
```

```
plugin store "AWSAssetDiscoveryPlugin" set "{parameter
 "secondAccessKey"}" value "<myUserKey3>" on "{parameter "action issue
 date" of action}"
plugin store "AWSAssetDiscoveryPlugin" set encrypted "{parameter
 "secondPassword"}" value "<myUserPass3>" on "{parameter "action issue
 date" of action}"
plugin store "AWSAssetDiscoveryPlugin" set "Credentials_Region_{parameter
 "secondLabel"}" value "af-south-1" on "{parameter "action issue date" of
 action}"
```

必要なものが揃ったので、埋め込まれた XML ファイルを **action.xml** などのカスタム名で保存するだけです。

```
<?xml version="1.0" encoding="utf-8"?>
<BES xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema" SkipUI="true">
    <SingleAction>
        <Title>AWS Plugin Config Template</Title>
        <Relevance>true</Relevance>
        <ActionScript MIMEType="application/x-Fixlet-Windows-Shell">
parameter "firstLabel" = "testuser2"
parameter "secondLabel" = "testuser3"

parameter "accessKey" = "Credentials_AccessKey"
parameter "secretAccessKey" = "Credentials_SecretAccessKey"
parameter "region" = "Credentials_Region"

parameter "firstAccessKey" = "{parameter "accessKey"}_{parameter
 "firstLabel"}"
parameter "firstPassword" = "{parameter "secretAccessKey"}_{parameter
 "firstLabel"}"
```

```
plugin store "AWSAssetDiscoveryPlugin" set "{parameter
 "firstAccessKey"}" value "<myUserKey2>" on "{parameter "action issue date"
 of action}"
plugin store "AWSAssetDiscoveryPlugin" set encrypted "{parameter
 "firstPassword"}" value "<myUserPass2>" on "{parameter "action issue date"
 of action}"
plugin store "AWSAssetDiscoveryPlugin" set "Credentials_Region_{parameter
 "firstLabel"}" value "us-east-1" on "{parameter "action issue date" of
 action}"

parameter "secondAccessKey" = "{parameter "accessKey"}_{parameter
 "secondLabel"}"
parameter "secondPassword" = "{parameter "secretAccessKey"}_{parameter
 "secondLabel"}"
plugin store "AWSAssetDiscoveryPlugin" set "{parameter
 "secondAccessKey"}" value "<myUserKey3>" on "{parameter "action issue
 date" of action}"
plugin store "AWSAssetDiscoveryPlugin" set encrypted "{parameter
 "secondPassword"}" value "<myUserPass3>" on "{parameter "action issue
 date" of action}"
plugin store "AWSAssetDiscoveryPlugin" set "Credentials_Region_{parameter
 "secondLabel"}" value "af-south-1" on "{parameter "action issue date" of
 action}"
       </ActionScript>
       <SuccessCriteria />
       <Settings />
       <SettingsLocks />
       <Target>
           <ComputerID>1078556546</ComputerID>
       </Target>
   </SingleAction>
```

注: testuser2 と testuser3 は、架空のラベル名です。ただし、**Credentials_AccessKey**、**Credentials_SecretAccessKey**、**Credentials_Region**は実際の設定名です。これらのパラメーターを定義します。これらのパラメーターを組み合わせることにより、上記のようにユーザー・キーとパスワードを設定するために必要なキーを定義できるためです。

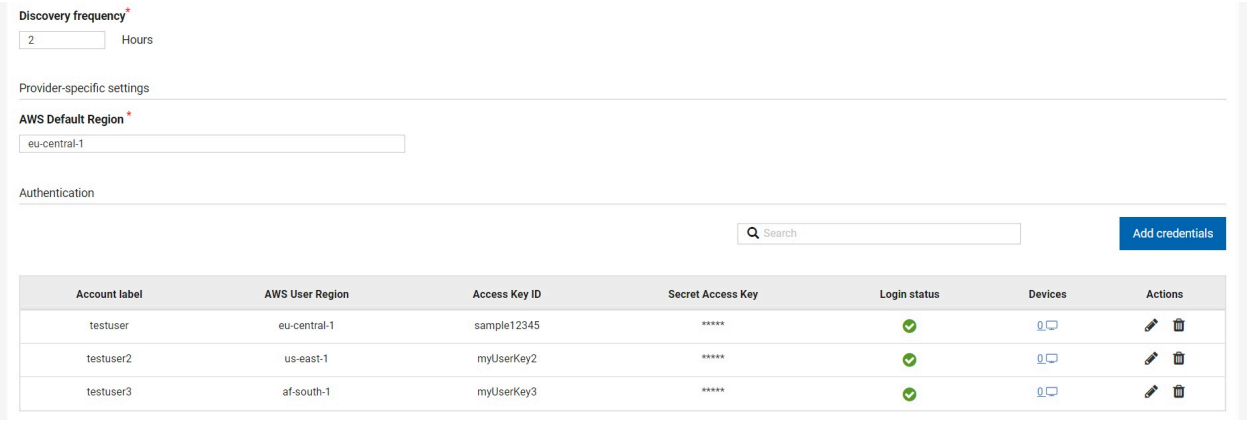
REST API を使用して **action.xml** に含まれるアクションを起動するには、IEM CLI を使用して以下のコマンドを発行します。

```
iem POST <path to action.xml> /api/actions
```

出力は次のようになります。

```
C:\Program Files (x86)\BigFix Enterprise\BES Server\IEM CLI>iem POST C:\Users\Administrator\Desktop\action.xml /api/actions
<?xml version="1.0" encoding="UTF-8"?>
<BESAPI xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="BESAPI.xsd">
        <Action Resource="https://[redacted]:52311/api/action/149" LastModified="Fri, 16 Jul 2021 10:30:37 +0000">
                <Name>AWS Plugin Config Template</Name>
                <ID>149</ID>
        </Action>
</BESAPI>
```

アクションが正常に完了したかどうかを WebUI で確認します。この場合、2 人の新規ユーザーが即時に使用可能になります。

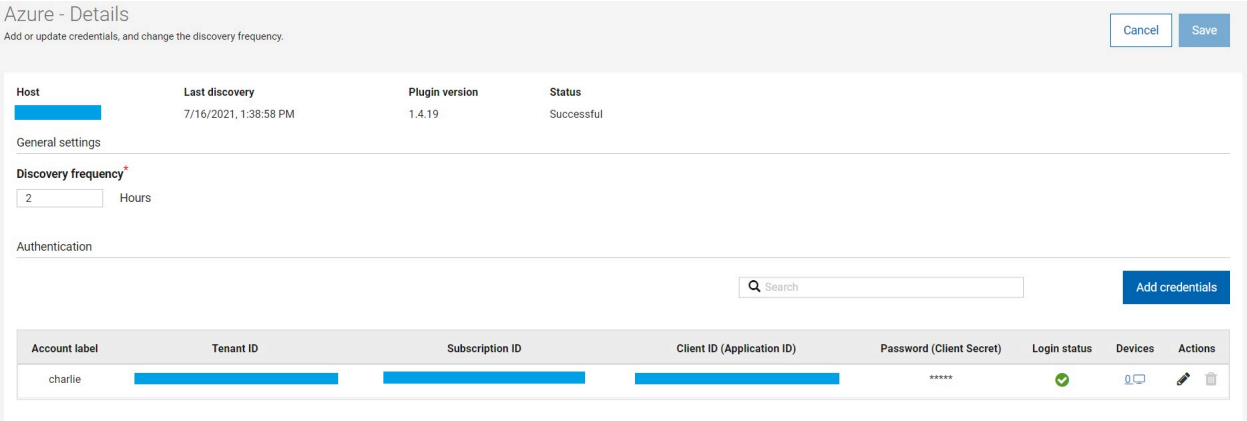


例 2: カスタム・アクションを使用した Azure プラグインの構成

この例では、BigFix WebUI を使用してカスタム・アクションを作成して実行することにより、Azure Asset Discovery プラグインを構成します。

『例 1: AWS プラグインの構成 (（ページ） 179)』で説明されている AWS の例と同様に、Azure プラグインが既にインストールされていると仮定します。

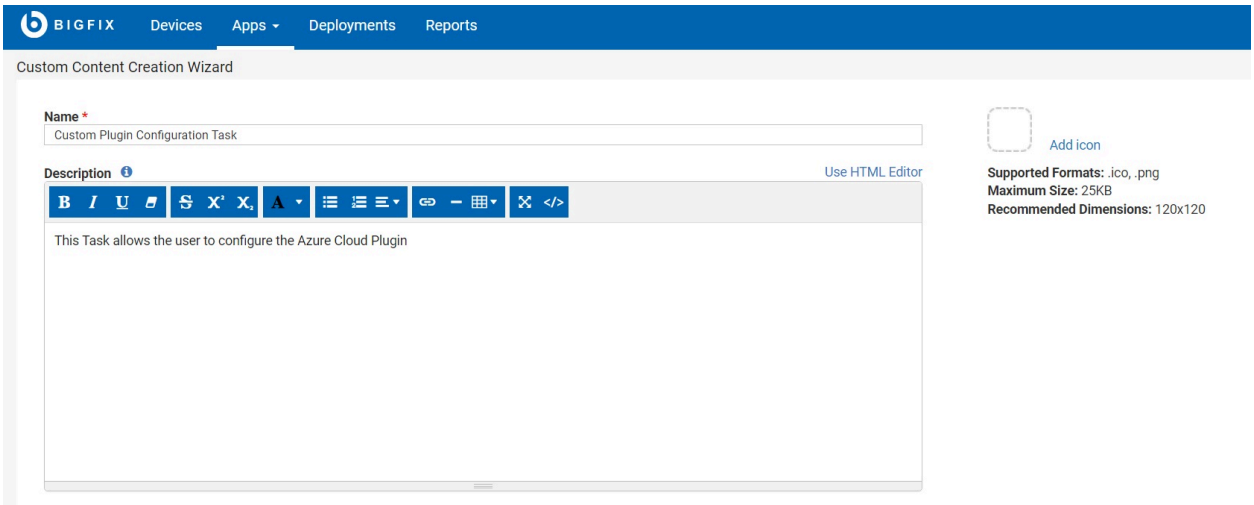
Azure プラグインについて WebUI によって報告される内容の詳細については、以下を参照してください。



2 人のユーザー (**foo** と **bar**) を構成するためのカスタム・アクションを作成します。このユーザーには、**TenantID**、**SubscriptionID**、**ClientID**、**ClientSecret** が使用可能です。

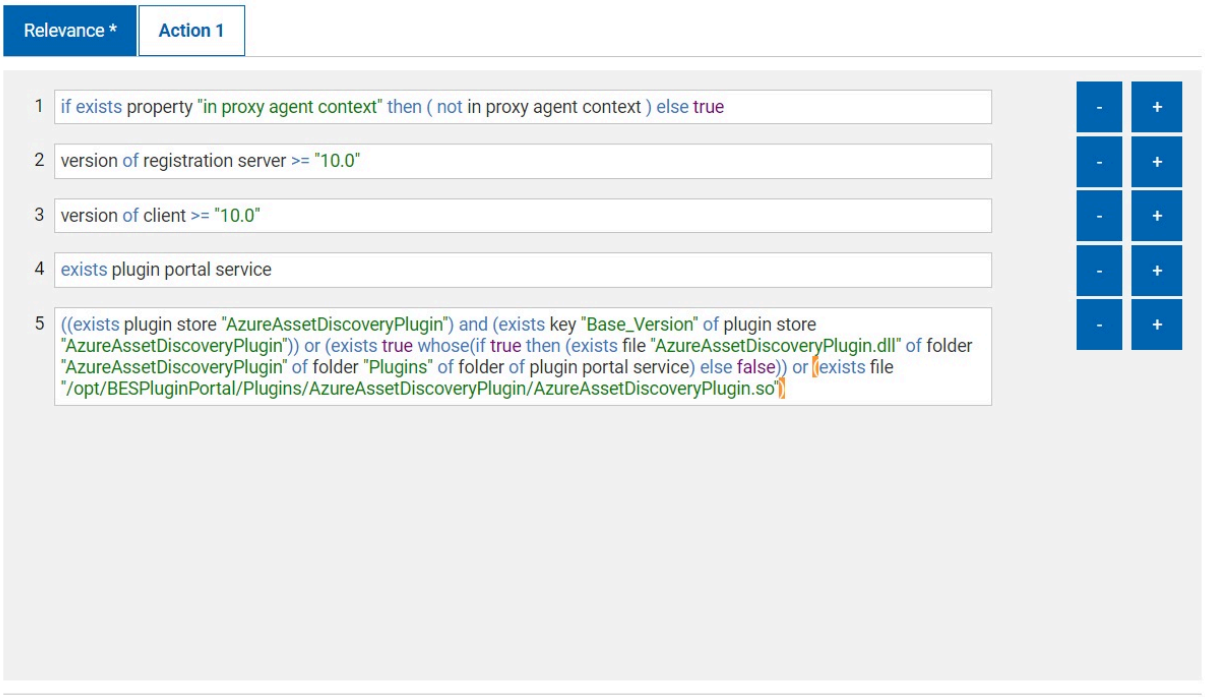
「WebUI」>「アプリケーション」>「カスタム」で、**「カスタム・コンテンツの作成」**をクリックします。

必要に応じて、アイテムの名前と説明を入力します。



その後、以下の関連度を追加して、適切なデバイスを確実にターゲットにします (Azure プラグインがインストールされているプラグイン・ポータルである必要があります)。

```
if exists property "in proxy agent context" then ( not in proxy agent
 context ) else true

version of registration server >= "10.0"

version of client >= "10.0"

exists plugin portal service

((exists plugin store "AzureAssetDiscoveryPlugin") and (exists key
 "Base_Version" of plugin store
"AzureAssetDiscoveryPlugin")) or (exists true whose(if true then (exists
 file "AzureAssetDiscoveryPlugin.dll" of folder
"AzureAssetDiscoveryPlugin" of folder "Plugins" of folder of plugin portal
 service) else false)) or (exists file
"/opt/BESPluginPortal/Plugins/AzureAssetDiscoveryPlugin/AzureAssetDiscovery
Plugin.so")
```



ここで、AWS プラグインの実行と同様に、構成のアクション・スクリプトを作成します。

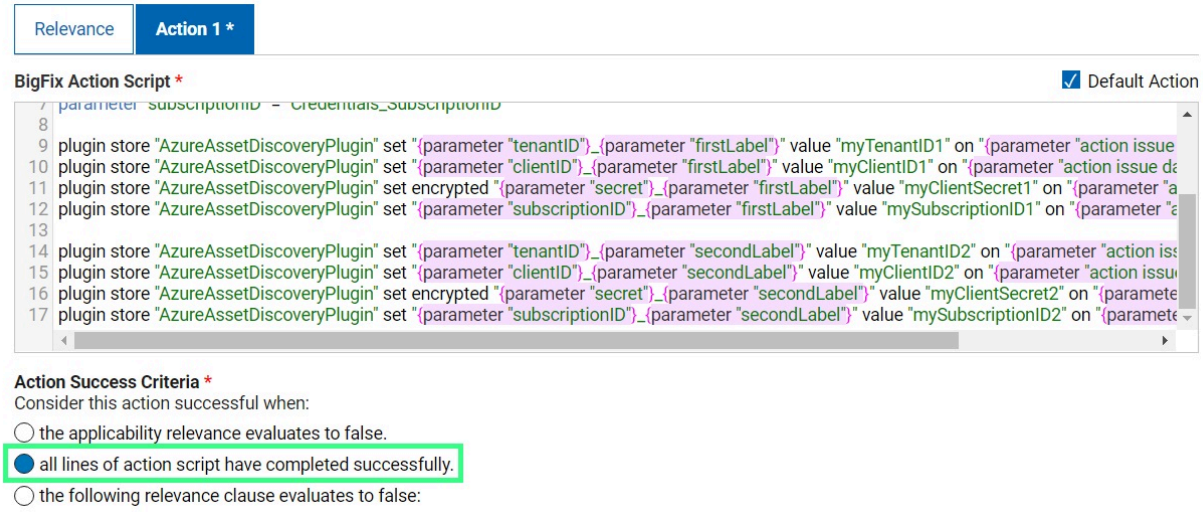
注: 置換する必要がある値は、<myTenantID1> のように不等号括弧で囲まれた値のみです。

```
parameter "firstLabel" = "foo"
parameter "secondLabel" = "bar"

parameter "tenantID" = "Credentials_TenantID"
parameter "clientID" = "Credentials_ClientID"
parameter "secret" = "Credentials_ClientSecret"
parameter "subscriptionID" = "Credentials_SubscriptionID"

plugin store "AzureAssetDiscoveryPlugin" set "{parameter
 "tenantID"}_{parameter "firstLabel"}" value "<myTenantID1>" on "{parameter
 "action issue date" of action}"
```

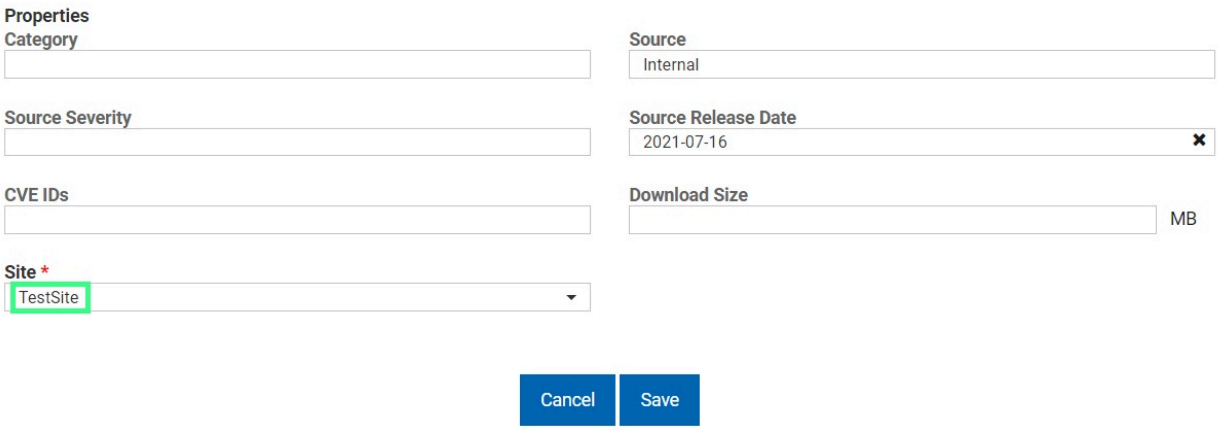
```
plugin store "AzureAssetDiscoveryPlugin" set "{parameter
 "clientID"}_{parameter "firstLabel"}" value "<myClientID1>" on "{parameter
 "action issue date" of action}"
plugin store "AzureAssetDiscoveryPlugin" set encrypted "{parameter
 "secret"}_{parameter "firstLabel"}" value "<myClientSecret1>" on
 "{parameter "action issue date" of action}"
plugin store "AzureAssetDiscoveryPlugin" set "{parameter
 "subscriptionID"}_{parameter "firstLabel"}" value "<mySubscriptionID1>" on
 "{parameter "action issue date" of action}"

plugin store "AzureAssetDiscoveryPlugin" set "{parameter
 "tenantID"}_{parameter "secondLabel"}" value "<myTenantID2>" on
 "{parameter "action issue date" of action}"
plugin store "AzureAssetDiscoveryPlugin" set "{parameter
 "clientID"}_{parameter "secondLabel"}" value "<myClientID2>" on
 "{parameter "action issue date" of action}"
plugin store "AzureAssetDiscoveryPlugin" set encrypted "{parameter
 "secret"}_{parameter "secondLabel"}" value "<myClientSecret2>" on
 "{parameter "action issue date" of action}"
plugin store "AzureAssetDiscoveryPlugin" set "{parameter
 "subscriptionID"}_{parameter "secondLabel"}" value
 "<mySubscriptionID2>" on "{parameter "action issue date" of action}"
```

注: foo と bar は、架空のラベルの名前です。ただし、**Credentials_TenantID**、**Credentials_ClientID**、**Credentials_ClientSecret**、**Credentials_Subscri**は実際の設定名です。これらの 4 つのパラメーターを定義します。これらのパラメーターを組み合わせることにより、上記のようにユーザー・キーとパスワードを設定するために必要なキーを定義できるためです。

次に、スクリプト全体をコピーして「アクション」ボックスに貼り付け、**「アクション・スクリプトのすべての行が正常に完了した場合。」**を選択していることを確認します。

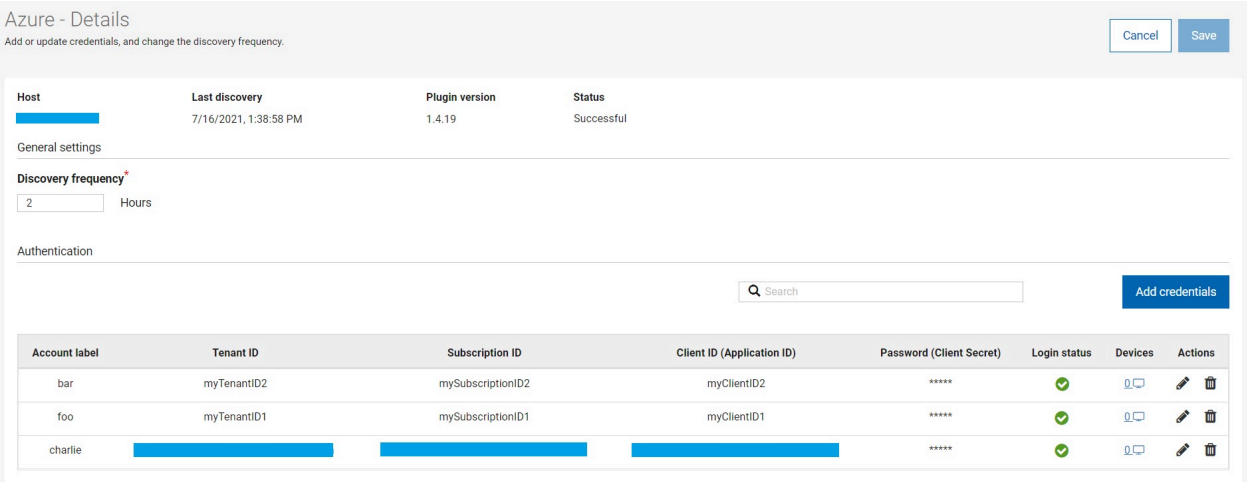


必要な**「サイト」**を選択し、**「保存」**をクリックします。



これで、**「WebUI」>「アプリケーション」>「カスタム」**アプリケーションに戻って、新しく作成されたタスクをデプロイしてプラグインを構成できます。

この操作を完了すると、「プラグイン管理」パネルに新規ユーザーが表示されます。



例 3: 複数設定の一括構成

この例では、プラグインの大規模な構成を迅速にディスパッチするために、**plugin store** コマンドの **multiple set** オプションを使用して、一度に多くの設定を構成します。

plugin store のコマンドおよびオプションについて詳しくは、『plugin store』と『概要（（ページ） 165)』を参照してください。

このタイプのコマンドの構文は次のとおりです。

```
plugin store “<plugin name>” multiple set “<percent encoded json>” on
 “<now>”
```

次のような形式の JSON ファイルに、プラグインの設定が既に多数収集されているとします。

```
{
    “settingKey1”: “settingValue1”,
    “settingKey2”: “settingValue2”,
    “settingKey3”: “settingValue3”,
    …
    “settingKeyN”: “settingValueN”,
}
```

『例 1: AWS プラグインの構成 (（ページ） 179)』で説明されているように、新しいアクションを作成し、アクション・スクリプトに必要なコマンドを入力するだけです。この場合、適切なセクションでパーセント・エンコーディングされた JSON をコピーして貼り付けた後に、multiple set を発行します。

```
<?xml version="1.0" encoding="utf-8"?>
<BES xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema" SkipUI="true">
    <SingleAction>
        <Title>AWS Plugin Config Template</Title>
        <Relevance>true</Relevance>
        <ActionScript MIMEType="application/x-Fixlet-Windows-Shell">
            plugin store "AWSAssetDiscoveryPlugin" multiple set "<percent
 encoded json>" on "{parameter "action issue date" of action}"
        </ActionScript>
        <SuccessCriteria />
        <Settings />
        <SettingsLocks />
        <Target>
            <ComputerID>1078556546</ComputerID>
        </Target>
    </SingleAction>
</BES>
```

REST API を使用してこのアクションを発行すると、データベース内の構成 JSON のすべてのキー値が追加されます。

下の図に示すように、1 回の操作で多数の設定をすばやく追加できます。

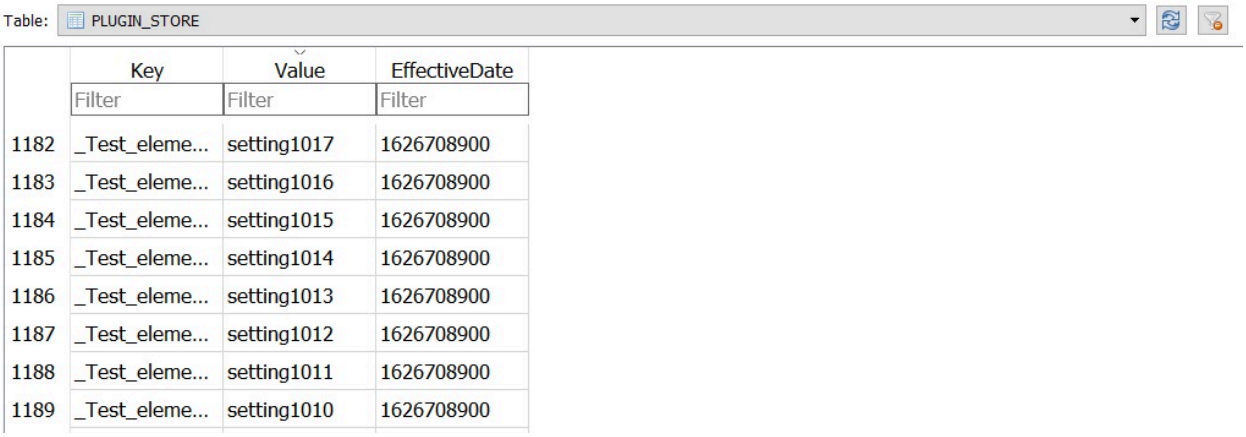


Table: PLUGIN_STORE

	Key	Value	EffectiveDate
	Filter	Filter	Filter
1182	_Test_eleme...	setting1017	1626708900
1183	_Test_eleme...	setting1016	1626708900
1184	_Test_eleme...	setting1015	1626708900
1185	_Test_eleme...	setting1014	1626708900
1186	_Test_eleme...	setting1013	1626708900
1187	_Test_eleme...	setting1012	1626708900
1188	_Test_eleme...	setting1011	1626708900
1189	_Test_eleme...	setting1010	1626708900

「encrypted」キーワードは追加されていないため、追加された設定は平文であることに注意してください。

このタイプのアプローチは、コードから実行することを意図しています。このため、より簡単な手順は、JSON ファイルをエンコードするための専用スクリプトを作成し、アクション・スクリプトを作成し、REST API を使用して新規アクションを発行することです。

第 15 章. BigFix 管理機能の拡張

BigFix 10 は、デバイスが物理デバイスか仮想デバイスかに関係なく、ネットワーク上のデバイスの可視性と管理を強化するいくつかの重要な新機能を備えています。

最新の IT インフラストラクチャーの管理で直面する課題

インフラストラクチャーの管理は、IT 組織にとってますます困難で複雑になっています。複数の種類のサーバー、さまざまなオペレーティング・システム、ソフトウェア、クラウド・コンピューティングとサービス、刻々と変化するテクノロジーの出現により、IT 環境の追跡、制御、管理が難しくなります。

- クラウド・コンピューティングやモビリティなどのテクノロジーは、IT ランドスケープを急速に変化させ、最新の状態を維持することが困難になります。
- 従来のコンプライアンスを遵守しながら、新しいコンプライアンスと規制の要件に対応するために、コスト効率の高いソリューションの必要性が高まっています。
- IT 組織が最新のテクノロジーを中心に運用を拡大し続けると、セキュリティーが大きな関心事になります。
- 高度なコンピューティングとデータ分析をサポートする高度な IT インフラストラクチャーには、効率的で費用対効果の高いデータ抽出とデータ・ストレージ技術が必要です。

BigFix 10 の機能

異機種 IT 環境全体の透明性を実現するには、BigFix 10 のようなより自動化された包括的で堅牢なソリューションが必要です。このまったく新しいバージョンの BigFix は、ネットワーク内のリソースの正確なビュー、主要な分析、詳細な分析情報を提供するため、意思決定者は、IT 管理に関する情報に基づいたより迅速な意思決定を行うことができます。

関連情報

クラウド・リソースの管理 (（ページ） *195)*

WebUI でのクラウド・プラグインの管理 (（ページ）)

最新のクライアント管理

Insights

クラウド・リソースの管理

各組織の IT エコシステムは拡大し続けており、サーバー、Mac、Linux デスクトップ、Linux ラップトップなどのさまざまなエンドポイントが組み込まれています。クラウド・コンピューティングの出現により、データ・センターなどの自社のオンプレミス・インフラストラクチャー全体を、Amazon Web Services、Microsoft Azure、VMware、Google Cloud Platform (GCP) などのクラウドに移行しようと考える IT 組織が増えています。したがって、エンドポイントの管理はますます複雑になり、クロス・ホスティング環境をサポートするエンドポイント管理ソリューションが必要とされています。

BigFix 10 クラウド・システム上のリソースも管理する機能を備えています。新機能により、BigFix の管理者およびオペレーターは、クラウド上 (パブリック、プライベート、ハイブリッド) の Windows、Mac、Linux のエンドポイントを完全に視覚化し、エンドポイントが物理的であるか仮想的であるかにかかわらず安全かつコスト管理された方法で管理することができます。

この機能により BigFix の機能が拡張され、単一のウィンドウで重複情報なく仮想デバイス (AWS®、VMware、Azure、GCP 対応) の検出、管理、保護できるようになります。

- プライベート・クラウド、パブリック・クラウド、ハイブリッド・クラウドを含む複数のクラウド・プロバイダー間のすべてのクラウド・インスタンスに BigFix の可視性を拡張します。
- クラウド・ネイティブな API と BigFix エージェントを組み合わせて活用し、仮想エンドポイントから情報を取得してクラウド・インスタンスを完全に可視化します。
- クラウドの管理を簡素化し、BigFix の自動化をクラウド・インスタンスに拡張します。
- クラウド・インスタンス・パッチについて詳細な情報を得た上で決断できます。

- エンドポイントのタイプまたは場所にかかわらず、単一のソリューションですべてのエンドポイントでの継続的なコンプライアンスを確保します。
- オンプレミス・チームとクラウド・チームがより効率的に作業するのに役立ちます。

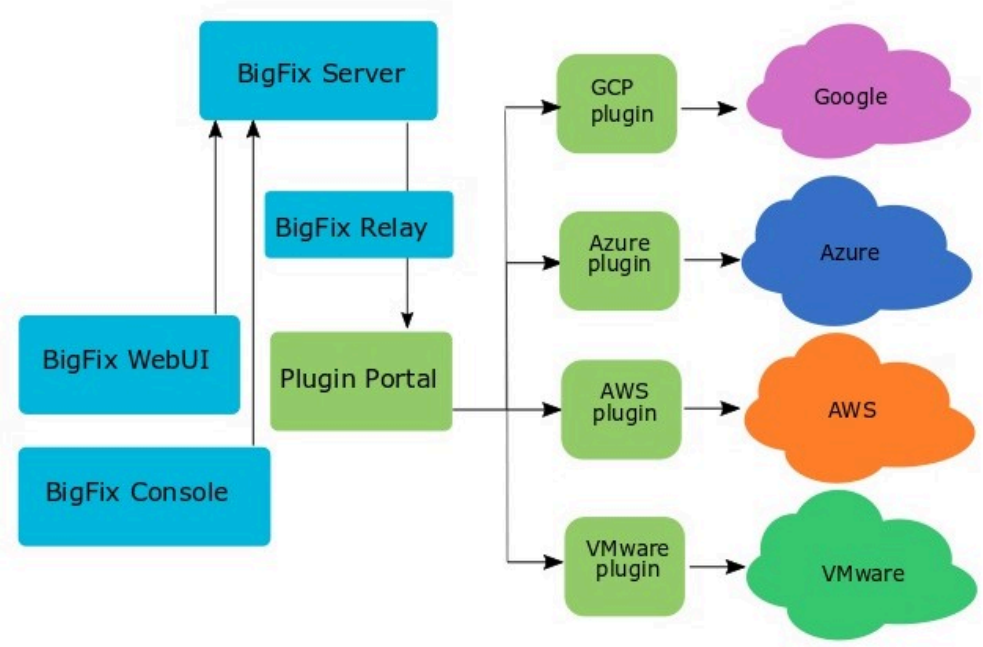
クラウド設定の理解

BigFix コンソールおよび WebUI は、BigFix エージェントによって管理されているか、クラウド・プラグインを介して プラグイン・ポータル (（ページ） 155) によって管理されているか、またはその両方によって管理されているかに関係なく、オペレーターにクラウド・リソースの単一の表現を提供します。これにより、オペレーターはすべてのクラウド・インスタンスを完全に可視化して管理することができます。

BigFix 10 は、Amazon Web Services (AWS)、VMware、Microsoft Azure、および Google のクラウド・プラットフォームについて以下のユース・ケースをサポートしています。

- BigFix のマスター・オペレーターは以下を実行することができます。
  - 安全かつコストを管理しながら環境を維持しながら、すべてのリソースの制御、実行中のリソースの監視ができます。
  - IT エンタープライズのサービスを確実に維持し、安全に管理できます。
  - クラウド・インスタンスを監視できます。
- BigFix オペレーターは、クラウド・インスタンスの整理および管理ができ、そのインベントリーを管理できます。
- ビジネスに不可欠なアプリケーションを実行するサーバーの所有者は、クラウド・インスタンスを円滑に機能させることができます。
- コンテンツ作成者は、カスタム・コンテンツを作成することなく、分析を実行してクラウド・リソースとプロパティーを検査することができます。

マルチクラウド管理機能により、BigFix のコンソールおよび WebUI の両方を使用できます。



プラグイン・ポータルのデプロイメント・トポロジーに関しては、以下のオプションを使用できます。

- ポータル・インスタンスは、BigFix ルート・サーバーからリモートでデプロイできます。これは、ポータルが管理している大規模なリモート・クラウド・インスタンスがある場合に便利な場合があります。
- プラグイン・ポータルは、大量のレポートを生成することがあります。したがって、上位レベルのリレー・インフラストラクチャー (存在する場合) では、高帯域幅を提供する必要があります。例えば、プラグイン・ポータルが多数のデバイスを管理する場合、最上位リレー、偽のルート・リレー、またはルート・サーバー自体に対してプラグイン・ポータル・レポートを作成することをお勧めします。

クラウド・リソースの検出

BigFix 10 は、サポートされている任意のクラウド・プロバイダーで所有しているリソースを検出する機能を備えています。

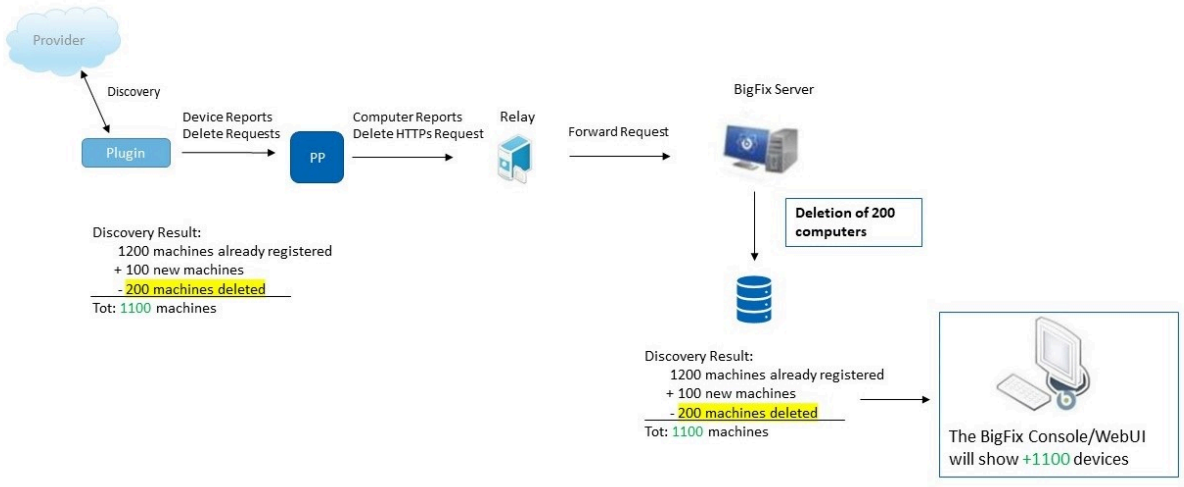
検出はクラウド・プラグインによって実行されます。各プラグインは、対応するクラウド・プロバイダーに定期的に照会を行い、使用可能なクラウド・リソース・データを取得します。取得されたデータはプラグイン・ポータルによって処理され、最終的に BigFix サーバーに送信されます。BigFix エージェントがインストールされているクラウド・リソースの場合、新しいクラウド・プロバイダーのインスペクターにより、コンピューターの相関 (（ページ） 200)の目的で使用されるデータを取得できます。

BigFix バージョン 10.0.8 では、BigFix データベースで検出されなくなったコンピューター・インスタンスを削除する新機能が導入されました。クラウド・プラグインは検出中にデバイスが存在しないことを認識すると、プラグイン・ポータルにレポートを送信します。このレポートは、該当デバイスをデータベースから削除し、削除 HTTPS Post 要求をサーバーに送信します。これにより、データベースからの物理的削除または論理削除のいずれかが実行されます。

デフォルトでは、BigFix サーバーは論理削除を実行します。すべてのデータは引き続きデータベースに残りますが、デバイスは削除済みとしてマークされます。誤った資格情報などのエラーが原因でデバイスは検出されなくなりますが、削除されません。新しい設定がクラウド・プラグインに追加され、データベースからの物理的削除を実行できます。このオプションを使用すると、BigFix サーバーは、BES コンピューター・リムーバー・ツールと同様に、デバイスとそれに関連するすべてのデータをデータベースから物理的に削除します。クラウド・プラグインに別の設定が追加され、すべての相関関係にあるインスタンスの削除も可能になりました。これらの設定はどちらも WebUI で変更できます。これらの設定について詳しくは、クラウド・プラグインの構成 (（ページ） 219) を参照してください。

次の例では、プラグインは 100 台の新しいマシンを検出しますが、別の 200 台を検出しないため、プラグイン・ポータルは 100 台の新規マシンの新規レポートと、その他の 200 台のマシンの空のレポートを受け取ります。プラグイン・ポータルでは、新規デバイスを登録し、検出されないデバイスを削除する必要があります (デバイス・アラート/削除要求の矢印)。この新機能により、プラグイン・ポータルは検出されなくなったデバイスに対しても HTTP 削除要求を送信します (HTTP 削除要求矢印)。リレーは BigFix サーバーに要求を

転送します。このサーバーは、BigFix データベースから検出されない 200 個のデバイスの物理削除または論理削除 (上記にて紹介し、すでに説明されている新しいプラグイン設定に応じる) のいずれかを実行します。削除されたデバイスは BigFix コンソールと WebUI コンピューター・リストに表示されなくなります。BigFix コンソールと WebUI には、実際に検出されたデバイスのリストが常に表示されます。



プラグイン・ポータルは、削除するデバイスのセットを異なるリストに編成して BigFix サーバーに送信します。これらのリストは、すでに記述されている新しいプラグイン設定で設定された値の組み合わせによって定義されます。新しい `_BESPluginPortal_DeleteDevices_MaxNumberOfDevicesPerBlock` 設定では、各リストに含めることができるデバイスの最大数を定義できます。デフォルトで、各リストには最大 500 個のデバイスを含めることができます。『`_BESPluginPortal_DeleteDevices_MaxNumberOfDevicesPerBlock` 設定』について詳しくは、プラグイン・ポータル (（ページ）) を参照してください。

関連情報

プラグイン・ポータル (（ページ） *155)*

クラウド設定の理解 (（ページ） *196)*

相関関係にあるデバイス

BigFix 10 は、同一コンピューターの複数の表現を相関させる機能を備えているため、オペレーターは単一のエンティティーとして操作できる (本文書では相関関係にあるデバイスとも呼ばれる) ほか、必要に応じて特定の表現を管理することもできます。

例えば、Microsoft Azure のクラウド・プラグインが Microsoft Azure 上に作成された VM を検出し、同時にその VM が BigFix エージェントを実行していることを検出すると、同一コンピューター上にある 2 つの別の表現が BigFix サーバーに報告されます。この場合、BigFix 10 は 2 つの表現を相関させ、BigFix コンソールはこれをグループ化して展開可能な形で表示します。

本書では、クラウド・プラグインが検出したコンピューター表現を表すのに「プロキシー」という用語を使用します。また、「ネイティブ」という用語は、BigFix エージェントに関連付けられたコンピューター表現を表します。

注: 相関関係にあるデバイスでは、BigFix エージェントのバージョン 10 以上が必要です。

クラウド・プラグインによる相関関係にあるデバイスの有効化

クラウド・プラグインがインストールされると、そのプラグインが検出したデバイスの相関機能が自動的に有効化されます。

相関関係にあるデバイスの表示

相関関係にあるデバイスは論理エンティティーであり、BigFix コンソールの**「コンピューター」**ビューで展開可能なオブジェクトとして表示されます。オブジェクトのルート・エレメントは相関関係にあるデバイスを示し、固有のコンピューター ID を有します。相関関係にあるデバイスの ID は相関の時点で作成され、通常は単一表記の ID より大きい値が付けられます。相関関係にあるデバイスを展開すると、相関表現はわずかにインデントされて表示されます。

Computers

Computer Name	Agent Type	Device Type	OS	ID
NC148399	Native	Server	Win10 10.0.18363.720 (1909)	1088897586
nc926099	Native	Server	Mac OS X 10.15.4 (19E266)	1626695561
ip-172-31-16-130.eu-west-3.compute.internal	Native	Server	Linux Amazon 2 (4.14.165-131.185.amzn2.x86_64)	2154271525
ip-172-31-16-130.eu-west-3.compute.internal	Native	Server	Linux Amazon 2 (4.14.165-131.185.amzn2.x86_64)	6787877
ip-172-31-16-130	Proxy - Amazon Web Services	Cloud	N/A	1626805836
NC926057	Native	Server	Win10 10.0.18363.720 (1909)	2154568203
nc926163.prod.hclpnp.com	Native	Server	Linux Red Hat Enterprise Linux 8.1 (4.18.0-147.el8.x86_64)	2691200772
nc926163.prod.hclpnp.com	Native	Server	Linux Red Hat Enterprise Linux 8.1 (4.18.0-147.el8.x86_64)	539137562
NC926163-RHEL8	Proxy - VMware	Cloud	Red Hat Enterprise Linux 8 (64-bit)	543717124
azure-sles-system	Native	Server	Linux SuSE Enterprise Server 15.1 (4.12.14-8.22-azure)	2692268216
azure-sles-system	Native	Server	Linux SuSE Enterprise Server 15.1 (4.12.14-8.22-azure)	544784568
azure-sles-system	Proxy - Microsoft Azure	Cloud	Linux	1078763439
NC926171	Native	Server	Win2019 10.0.17763.1098 (1809)	2699955519
ip-172-31-43-76	Proxy - Amazon Web Services	Cloud	N/A	15218897
ip-172-31-47-141	Proxy - Amazon Web Services	Cloud	windows	542673993
ip-172-31-31-4	Proxy - Amazon Web Services	Cloud	N/A	1080191625
ip-172-31-37-184	Proxy - Amazon Web Services	Cloud	N/A	1613760714
azure-system-A	Proxy - Microsoft Azure	Cloud	Windows	2935830
win10Pro1809img	Proxy - Microsoft Azure	Cloud	Windows	13514690
RHEL76-auto	Proxy - Microsoft Azure	Cloud	Linux	546703723
SystemA	Proxy - Microsoft Azure	Cloud	Linux	549953433
Linux_Server_Machine	Proxy - VMware	Cloud	Red Hat Enterprise Linux 7 (64-bit)	294345
nc132125-win10	Proxy - VMware	Cloud	Microsoft Windows 10 (64-bit)	2153091
nc1474121-RHEL8-(EFI)	Proxy - VMware	Cloud	Red Hat Enterprise Linux 8 (64-bit)	5346946
nc141067-sol11	Proxy - VMware	Cloud	Oracle Solaris 11 (64-bit)	6422506
NC1474160	Proxy - VMware	Cloud	Apple macOS 10.14 (64-bit)	1089635679

相関関係にあるデバイスは、相関関係にあるデバイスからプロパティーを継承します。デバイスが同一プロパティーに対して異なる値を報告した場合、相関関係にあるデバイスは、最も正確かつ意味のあるデータ・ソースであるネイティブの値を継承します。

注:

- 相関表現のうち 1 つが BigFix コンソールのプリファレンスに指定された時間内にチェックインされず、さらにオフラインである場合は、相関関係にあるデバイスもオフラインになります。
- プロキシー・コンピューターに関連付けられているエージェントのバージョンは、それらを管理しているプラグイン・ポータルのバージョンに対応しています。
- AWS プラグインによって検出されたプロキシー・コンピューターのコンピューター名は、プライベート DNS 名から取得されたホストの名前に対応します。

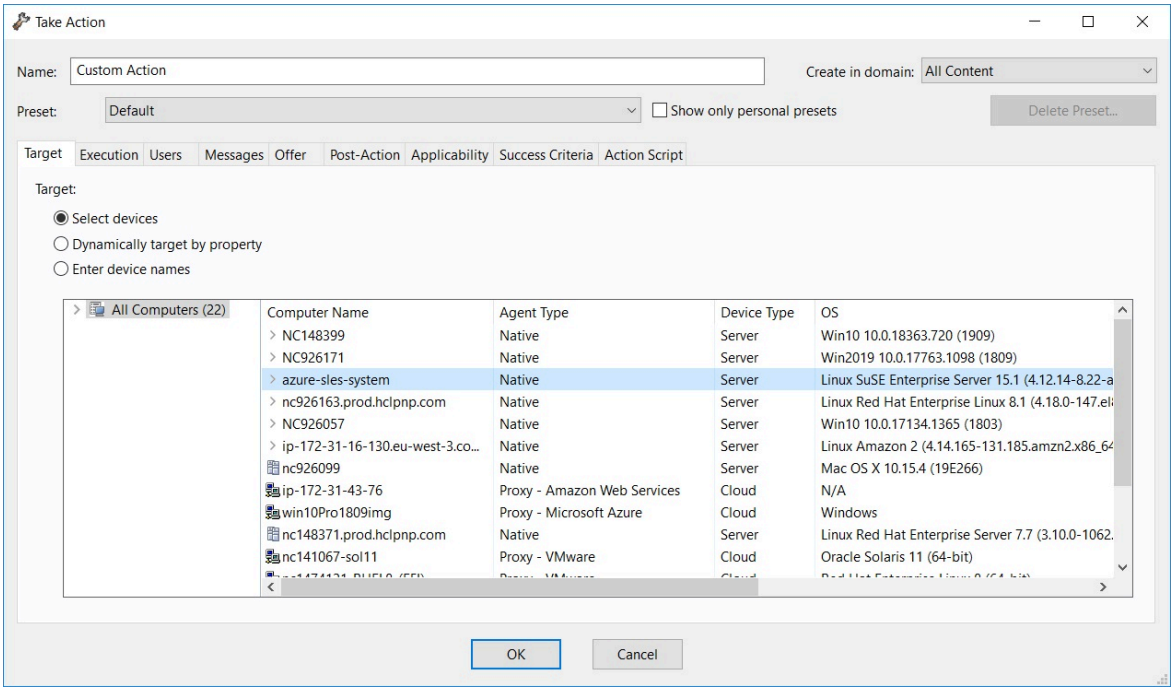
相関関係にあるデバイスの管理

相関関係にあるデバイスはマスター・オペレーターと、2 つ以上の相関表現を管理するマスター・オペレーターではない管理者に対して表示されます。オペレーターは、相関表現を相互に独立して管理できます。いかなる継承メカニズムも、管理権限をある相関表現から相関表現に伝播することはありません。

相関関係にあるデバイスでの操作の実行

オペレーターは相関関係にあるデバイスをターゲットに設定することができ、操作のタイプに応じて BigFix サーバーが適切な相関表現にディスパッチします。以下にいくつかの例を示します。

- 下の図に示すように、オペレーターがカスタム・アクションを実行し、**「ターゲット」**タブで**「デバイスの選択」**を選択して、相関関係にあるデバイスを選択します。



この場合、BigFix サーバーは actionscrip コマンドの解析結果に基づいて、このアクションを相関表現の 1 つにディスパッチします。すべての actionscript コマンドを実行できる相関表現が 1 つしかない場合、BigFix サーバーはアクションをその表現にディスパッチします。actionscript 全体が 1 つ以上の表現に適用できる場合、BigFix サーバーは常にアクションをネイティブ・コンピューターにディスパッチします。

注:

- オペレーターが特定のコンピューター表現を対象にする場合、**「ターゲット」**タブの 相関関係にあるデバイスを展開して該当の表現を選択します。この場合、BigFix サーバーはアクションをこの表現に直接送信します。
- 1 つの表現のみに適用できる Fixlet からアクションが実行された場合 (このシナリオでは、**「ターゲット」**タブの相関関係にあるデバイスを展開して確認可能)、アクションはその表現に送信されます。

- オペレーターは相関関係にあるデバイスをターゲットに設定し、BigFix クエリを実行します。この場合、BigFix Query は常にネイティブ・コンピューターに送信されます。これは、BigFix エージェントが BigFix Query を実行できる唯一のコンポーネントであるためです。
- オペレーターは**「コンピューター」**ビューで相関関係にあるデバイスを選択し、マニュアル・コンピューター・グループに追加します。この場合、すべての相関表現がマニュアル・コンピューター・グループに追加されます。
- オペレーターは**「コンピューター」**ビューで相関関係にあるデバイスを選択し、**「データベースから削除...」**を選択します。この場合、すべての相関表現を削除するよう設定され、BigFix コンソールでは相関コンピューターとしてもスタンドアロン・コンピューターとしても表示されなくなります。
- オペレーターは「コンピューター」ビューで相関関係にあるデバイスを選択し、**「更新の送信」**を実行します。この場合、すべての相関表現に更新の通知が送信されます。

注: 相関関係にあるデバイスの ID を参照するクライアントの関連式は、いずれのネイティブ・コンピューターまたはプロキシー・コンピューターとも一致しません。これらの ID が BigFix サーバーのみに知られている論理エンティティーを表しているためです。例えば、相関関係にあるデバイスの ID を参照して、自動コンピューター・グループを定義するかコンピューターをサイトにサブスクライブしても、コンピューターはコンピューター・グループには含まれず、サイトにもサブスクライブされません。

相関関係にあるデバイスでの REST API の使用

BigFix 10 の REST API は、以前の BigFix リリースの XML スキーマ定義と引き続き互換性がある一方で、メソッドおよび適用可能なリソースを使用して相関関係にあるデバイスをサポートし、処理することもできます。

例えば、相関関係にあるデバイスの ID はアクションのターゲットとして使用できます。この場合、オペレーターの権限および ActionScript に含まれるコマンドに応じて BigFix サーバーがアクションを適切なターゲットにディスパッチします。

同様に、相関関係にあるデバイスの ID を使用して BigFix サーバーから情報を取得することもできます。例えば、オペレーターが相関関係にあるデバイスの設定を取得する場合、REST API はネイティブ・コンピューターの設定が記載されたメイン・セクションと、プロキシー・コンピューターの設定を含む `ManagementExtension` と命名されたサブセクションから構成される XML を返します。詳細については、『Computer REST APIs』を参照してください。

相関関係にあるデバイスでのセッション関連度の使用

セッション関連度を使用する場合、インスペクター `bes computers` および `bes computers set` (相関関係にあるデバイスの場合) は、CorrelationID に関連する BES Computer オブジェクト**のみ**を返します。相関関係にあるデバイスを表す BES コンピューター上の Fixlet のプロパティーまたは適用条件の値を求めるのは、優先順位に基づき表現 (ネイティブや Azure など) を照会し、最初に使用可能な値を返すのと同じです。

2 つの新しいセッション関連インスペクター、`bes computers with extensions` と `bes computers with extensions set` が導入され、返されるコンピューターのセットには、相関関係にあるデバイスと表現の**両方**を表す BES コンピューターが含まれます。

BES コンピューター・オブジェクトが相関関係にあるデバイスまたは相関関係にあるデバイスの表現を表しているかどうかを確認するために、BES コンピューター・オブジェクトに次の 2 つの新しいプロパティーが追加されました。

- `correlation flag of <bes_computer> : boolean` が相関関係にあるデバイスを表す BES コンピューター・オブジェクトに対して true を返します。
- `extension flag of <bes_computer> : boolean` が相関関係にあるデバイスの拡張を表す BES Computer オブジェクトに対して true を返します。

2 つのプロパティーも追加され、その相関表現の拡張を照会しました。

- `correlation of <bes_computer> : bes_computer` は、拡張から相関関係にあるデバイスの BES コンピューター・オブジェクトを返します。
- `correlation id of <bes_computer> : integer` は、拡張から相関関係にあるデバイスのコンピューター ID のみを返します。

例えば、次のような関連度があります。

```
(name of it, agent type of it, correlation id of it) of bes computer with
 extensions
whose (extension flag of it)
```

相関関係にあるデバイスで切り分けるすべてのデバイス、その名前、エージェント・タイプ、相関関係にあるデバイスの ID を返す。
上記の『**相関関係にあるデバイスの表示**』セクションのスクリーンショットのようなデプロイメントでは、次のタプルが返されます。

```
ip-172-31-16-130.eu-west-3.compute.internal, Native, 2154271525
ip-172-31-16-130, Proxy - Amazon Web Services, 2154271525
NC926057, Native, 2154568203
NC926057-Win10, Proxy - VMware, 2154568203
nc926163.prod.hclpnp.com, Native, 2691200772
NC926163-RHEL8, Proxy - VMware, 2691200772
azure-sles-system, Native, 2692268216
azure-sles-system, Proxy - Microsoft Azure, 2692268216
NC926171, Native, 2699955519
nc926171-Win2019-Srv, Proxy - VMware, 2699955519
```

詳しくは、『Computer Inspectors』を参照してください。

相関表現の削除

例えば、相関関係にあるデバイスがネイティブ表現およびプロキシー表現と相関しており、ある時点で 2 つの表現のうちどちらかが削除と設定された場合 (リソースがクラウド・プラグインで見つけられなくなった際、BigFixコンソールまたは BigFix コンピューター・

リムーバー・ツールから手動で削除するか、BigFix サーバーで自動的に削除します。詳しくは クラウド・リソースの検出 (（ページ） 198) をご覧ください)、相関関係にあるデバイスも削除するよう設定され、BigFix コンソールに表示されなくなります。残りの表現はスタンドアロン・コンピューターとして表示されます。

BigFix コンピューター・リムーバー・ツールがデータベースから表現を完全に削除すると、相関関係にあるデバイスもデータベースから削除されます。

クラウド・プラグインの概要

バージョン 10 以降、BigFix は Amazon Web Services、Microsoft Azure、Google Cloud Platform、VMware のクラウド環境を管理するために、プラグイン・ポータルに添付できるクラウド・プラグインのセットを提供しています。各プラグインは類似した機能のセットを有していますが、対応するクラウド・プロバイダーの性質上、一部の機能のサポートが異なります。

AWS Asset Discovery プラグイン

AWS Asset Discovery プラグインは、Amazon Web Services EC2 インスタンスに関するデータを検出して報告できます。さらに、AWS クラウド環境全体に BigFix エージェントをデプロイすることもできます。

構成

現時点では、AWS Asset Discovery プラグインは IAM ユーザーとロールを設定でサポートしています。AWS のベストプラクティスに沿って、シングルアカウントとクロスアカウントのシナリオがサポートされます。

詳しくは、『クラウド・プラグインのインストール (（ページ） 213)』および『BES プラグイン・ポータルのプラグインを構成するためのアクション・コマンド (（ページ） 165)』を参照してください。

BigFix エージェントのインストール

パッチ 2 以降の AWS プラグインではネイティブ API を使用した BigFix エージェントのデプロイメントをサポートしています。

詳しくは、『BigFix クラウド・リソースへのエージェントのインストール (（ページ） 285)』を参照してください。

インスペクター

AWS プラグインは、検出したいくつかの EC2 インスタンスのプロパティーを報告します。**パッチ 2 以降**では、追加の構成時に他のいくつかの AWS System Manager プロパティーを使用できます。また、各インスタンスは資格情報ラベルも報告するため、どの資格情報セットが情報を報告するかを識別することもできます。

詳しくは、『AWS Asset Discovery プラグイン・インスペクター (（ページ） 226)』を参照してください。

分析

各プラグインで分析を使用できます。詳しくは、クラウド分析のアクティブ化 (（ページ） 278) を参照してください。

リソースに関するデータについては、クラウド分析データ (（ページ） 278) を参照してください 。

Azure Asset Discovery プラグイン

Azure Asset Discovery プラグインでは、Microsoft Azure 仮想マシンに関するデータを検出して報告できます。さらに、Azure クラウド環境全体に BigFix エージェントをデプロイすることもできます。

構成

Azure Asset Discovery プラグインでは、**クライアント ID**、**パスワード**、**サブスクリプション ID** 、**テナント ID** を設定する必要があります。

詳しくは、『クラウド・プラグインのインストール (（ページ） 213)』および『BES プラグイン・ポータルのプラグインを構成するためのアクション・コマンド (（ページ） 165)』を参照してください。

BigFix エージェントのインストール

パッチ 2 以降では、Azure プラグインはネイティブ API を使用した BigFix エージェントのデプロイメントをサポートしています。

詳しくは、『BigFix クラウド・リソースへのエージェントのインストール (（ページ） 285)』を参照してください。

インスペクター

Azure プラグインは、検出されたインスタンスのいくつかのプロパティーを報告します。また、各インスタンスは資格情報ラベルも報告するため、どの資格情報セットが情報を報告したかを識別することもできます。

詳しくは、『Azure Asset Discovery プラグイン・インスペクター (（ページ） 237)』を参照してください。

分析

分析は各プラグインで使用できます。詳しくは、クラウド分析のアクティブ化 (（ページ） 278) を参照してください。

リソースに関するデータについては、クラウド分析データ (（ページ） 278) を参照してください。

GCP Asset Discovery プラグイン

パッチ 2 以降では、GCP Asset Discovery プラグインをインストールできます。プラグインは、GCP Compute Engine インスタンスに関するデータを検出して報告できます。

構成

GCP Asset Discovery プラグインを構成するには、資格情報 JSON が必要です。

パッチ 4 以降では、マルチプロジェクト・ディスカバリーも使用可能です。

詳しくは、『クラウド・プラグインのインストール (（ページ） 213)』および『BES プラグイン・ポータルのプラグインを構成するためのアクション・コマンド (（ページ） 165)』を参照してください。

インスペクター

GCP プラグインは、検出されたインスタンスのいくつかのプロパティーを報告します。また、各インスタンスは資格情報ラベルも報告するため、どの資格情報セットが情報を報告するかを識別することもできます。

詳しくは、『GCP Asset Discovery プラグイン・インスペクター (（ページ） 243)』を参照してください。

分析

各プラグインで分析を使用できます。詳しくは、クラウド分析のアクティブ化 (（ページ） 278) を参照してください。

リソースに関するデータについては、クラウド分析データ (（ページ） 278) を参照してください。

VMware Asset Discovery プラグイン

VMware Asset Discovery は、VMware ゲスト仮想マシンに関するデータを発見して報告できます。

パッチ 6 以降では、VMware ホストも発見に使用できるようになります。さらに、VMware プラグインはゲストとホストの両方に対してより多くのインスペクターとプロパティーを報告し、ユーザーが複数のアクション・コマンドを利用できるようにします。VMware プラグインの機能は、旧型の ESXi Management Extender の機能と一致します。

構成

VMware Asset Discovery プラグインを構成するには、**vCenter Server URL**とともに**ユーザー名**および**パスワード**を設定する必要があります。

詳しくは、『クラウド・プラグインのインストール (（ページ） 213)』および『BES プラグイン・ポータルのプラグインを構成するためのアクション・コマンド (（ページ） 165)』を参照してください。

アクション・コマンド

次の表に、使用可能なすべての新規アクション・コマンドを示します。このアクション・コマンドを使用するには、カスタム・アクションを作成するのではなく、Patch for ESXi、Virtual Endpoint Manager 、Server Automation サイトのコンテンツを使用することをお勧めします。

詳しくは、『VMware プラグイン・コマンド (（ページ） 274)』を参照してください。

ホスト	ゲスト・スナップショット	ゲスト電源	ゲスト VLAN	ゲスト・ツール	Patch	サーバー自動化
ホストのシャットダウン	スナップショットの作成	中断 (ハード)	VLAN の変更	マウント ツール	パッチ・アクションの初期化	ISO からの VM の作成

保守モードの開始	スナップショットの復元	中断 (ソフト)	アップグレード・ツール	vm の変更
保守モードの終了	スナップショットに移動	リセット		VM の削除
ホストの再起動	スナップショットの名前変更	再開		ISO 更新
転送スケジュール	すべてのスナップショットの削除	電源 ON		テンプレートからの Windows VM の作成
VLAN 電源オン	スナップショットの削除	電源オフ (ハード)		テンプレートからの Linux VM の作成
		電源オフ (ソフト)		VM をテンプレートに変換
				VM をテンプレートに複製

インスペクター

VMware プラグインは、VMware 仮想マシンのいくつかのプロパティーを報告します。**パッチ 6 以降**では、VMware ホストでも新しいインスペクターとプロパティーを使用できます。各インスタンスは資格情報ラベルを報告するため、どの資格情報セットが情報を報告するかを識別できます。

詳しくは、『VMware Asset Discovery プラグイン・インスペクター (（ページ） 250)』を参照してください。

分析

各プラグインで分析を使用できます。詳しくは、クラウド分析のアクティブ化 (（ページ） 278) を参照してください。

リソースに関するデータについては、クラウド分析データ (（ページ） 278) を参照してください 。

クラウド・プラグインの管理

BigFix 10 プラットフォームでは、Amazon Web Services (AWS)、Microsoft Azure、VMware、Google Cloud Platform (GCP) などの各クラウド・プロバイダーのプラグインがサポートされています。各クラウド・プロバイダーには独自の機能や外部プログラムとの接続方法があり、データへのアクセスや機能をさまざまな方法で処理しています。

プラグイン・ポータルおよびクラウド・プラグインをインストールできるようにするには、マスター・オペレーター (MO) 権限が必要です。

マルチクラウド管理機能により、BigFix のコンソールおよび WebUI の両方を使用できます。

関連情報

プラグイン・ポータルのインストール (（ページ）)

クラウド・プラグインのインストール (（ページ）)

クラウド・プラグインのインストールの計画

クラウド・プラグインは、プラグイン・ポータル (（ページ） 155)を実行しているコンピューターにインストールできます。オプションで、それらをすべて同じプラグイン・ポータル・インスタンスにインストールできます。

ただし、このオプションは、『BigFix キャパシティー・プランニング・ガイド』に含まれているプラグイン・ポータルのキャパシティー・プランニング要件を満たしている場合にのみ考慮に入れる必要があります。

BigFix のデプロイメントでは、タイプ (AWS、Microsoft Azure、VMware、GCP) ごとにクラウド・プラグインのインスタンスを 1 つだけ使用することを強くお勧めします。この推

奨事項の目的は、同じクラウド・リソースが複数回検出されるのを回避することです。そうしないと、1 つの相関コンピューターに同じプロキシー・コンピューターの複数のインスタンスが相関し、相関コンピューターの表示と管理に不必要な複雑さと負荷が増えることになります。

すべてのクラウド・プラグインのインストールには、サポートされるクラウド・プラットフォームに依存する特定の構成情報が必要ですが、さらに資格情報の入力が必要です。これにより、組み込みの SDK は対応するクラウド・プラットフォームの API を呼び出して、関連するクラウド・サービスにアクセスすることができます。すべてのクラウド・プラグインは資格情報ベースの検出を実行するため、仮想マシンを検出して関連データを取得できるかどうかは、クラウド・プラットフォーム側で資格情報を所有するユーザーに対して付与される権限に依存します。必要な権限の詳細については、「クラウド・プラグインのインストール (（ページ） 213)」を参照してください。

注: 各クラウド・プラグインは、複数の資格情報セットをサポートしています。ただし、インストール時に指定できるセットは 1 つだけです。追加の資格情報セットは、後で BigFix WebUI を使用して追加できます。

AWS、Microsoft Azure および Google Cloud プラグインをインストールするときは、これらのプラグインがインターネット経由で HTTPS を使用して関連するクラウド・サービスにアクセスできることを確認する必要があります。

クラウド・プラグインは、次のデフォルト・パスにインストールされます。

Windows:

- C:\Program Files (x86)\BigFix Enterprise\BES Plugin Portal\Plugins*Plugin_name*

Windows 以外:

- /opt/BESPluginPortal/Plugins/*Plugin_name*
- /var/opt/BESPluginPortal/Plugins/*Plugin_name*

クラウド・プラグインのインストール

各クラウド・プラグインには、BES サポートに関する特定のインストール・タスクがあり、プラグイン・ポータルがインストールされているコンピューターに関連します。

注: クラウド・プラグインの最終公開バージョンでは、最新バージョンのプラグイン・ポータルが必要な場合があります。古いプラグインは、新しいプラグイン・ポータルで引き続き正しく機能します。

タスクは、**「説明」**タブのすべての必須フィールドに入力した後でのみ実行できます。

クラウド・プラグインをインストールできるのは、マスター・オペレーター (MO) のみです。

すべてのクラウド・プラグインの拡張構成は、WebUI を使用して実行できます。

Amazon Web Services プラグイン

アカウント・ラベル

指定された `Access Key ID` / `Secret Access Key` ペアのわかりやすい名前。英数字のみを含める必要があります。

デフォルト・リージョン名

これは、プラグインが検出を実行するときに最初に接続する必要がある AWS リージョンの名前です。例えば、プラグインにヨーロッパ (フランクフルト) リージョンへの接続の検出を開始させる場合、指定する値は `eu-central-1` です。プラグインは、指定した `Access Key ID/Secret Access Key` ペアがアクセスできる他のすべてのリージョンに接続することで、その検出を自動的に完了します。

注:

- フィールドでは大文字と小文字が区別されます。AWS で文書化されているように、大文字と小文字を正しく入力してください。
- 新しい `Access Key ID` / `Secret Access Key` ペアを追加する場合、BigFix WebUI では、オプションでユーザー・リージョンの値を指定できます。指定されたキー・ペアのこの値がデフォルト・リージョンで優先されます。

使用可能なリージョンの詳細については、AWS の資料を参照してください。

AWS プラグインのインストール時に、許可されるリージョンを指定できます。AWS リージョンを制限する方法の詳細については、『Limit AWS Regions to restrict the scope of device discovery』を参照してください。

アクセス・キー ID とシークレット・アクセス・キー

IAM ユーザーに関連付けられた `Access Key ID` / `Secret Access Key` ペア。

IAM ユーザーの要件は次のとおりです。

- MFA を有効にしてはなりません
- プログラムによるアクセス・タイプが必要です
- 少なくとも次の権限が必要です。リソース "*" に対するアクション "**ec2:Describe***" が許可されている
  - 事前定義された適切な AWS ポリシーは *AmazonEC2ReadOnlyAccess* です

AWS アクセス・キーの詳細については、AWS の資料を参照してください。

IAM ロール

IAM ロールに関連付けられた ARN/リージョン/外部 ID。

BigFix v10.0 パッチ 4 と同時にリリースされたクラウド・プラグイン・バージョン 1.4 以降では、IAM ロールがサポートされています。IAM ロールは一連の割り当て権限を持つ識別情報です。ビジネス・ニーズに応じて、管理ユーザーを含む任意の信頼できるユーザーが、一時的に引き受けることができます。ロールには資格情報がないため、パスワードの有効期限の対象ではありません。役割を引き受ける場合、ログイン中のユーザーは一時的な資格

情報を要求します。期間は、管理ユーザーが割り当てた最大時間を超えることはできません。

注: IAM ロールを使用する場合は、ロールを引き受けるユーザーがロールに対して `sts:AssumeRole` を実行する権限を持っていることを確認してください。

注: 役割は、引き受けるユーザーを完全に置き換えます。つまり、ユーザーが管理する各操作は役割によって実行され、役割はクラウド・プラグインを管理するユーザーに必要な同じ権限を持っている必要があります。

注: AWS ロールが挿入されると、AWS プラグインは、取得元の資格情報ではなく、検出時に AWS ロールを使用します。クラウド環境で検出するすべての AWS デバイスがこれらの役割に含まれるようにする必要があります。そうしない場合、一部のマシンが検出されない可能性があります。

次の情報を指定する必要があります。

IAM ロールに関連付けられた `ARN` / `Region` / `External ID`。

各表記の意味は次のとおりです。

ARN

ロールの Amazon リソース名。AWS 上のリソースの固有 ID です。

AWN について詳しくは、AWS の資料を参照してください。

領域

(オプション): IAM ロールのデフォルト AWS リージョンです。詳しくは、『**デフォルトのリージョン名**』セクションを参照してください。新しい IAM ロールを追加する場合、BigFix ではオプションでロール・リージョンの値を指定できます。デフォルト・リージョンとユーザー・リージョンの両方で、指定されたロールが有効になります。

外部 ID

(オプション): AWS リソースへのアクセスをサード・パーティーに委任する必要がある場合は、IAM ロールを外部 ID と併用できます。これは、サード・パーティーがクラウド環境のリソースおよびサービスにアクセスして使用することを想定した設定です。**外部ID**は、環境を所有する組織によってサード・パーティーに提供される必要があります。また、**GUID** である必要があります。

外部 ID について詳しくは、AWS の資料を参照してください。

HTTP プロキシー

AWS プラグインがインストールされるシステムがインターネットに直接接続されていない場合は、オプションで HTTP プロキシーを指定できます。サポートされているプロキシーの認証方法については、AWS の資料を参照してください。

Microsoft Azure プラグイン

アカウント・ラベル

指定されたサービス・プリンシパル・カルテットのわかりやすい名前。英数字のみを含める必要があります。

クライアント ID、パスワード、サブスクリプション ID、テナント ID

サービス・プリンシパル・カルテット。サービス・プリンシパルの要件:

- 組み込みの リーダー の役割が割り当てられている必要があります。
- MFA を有効にしてはなりません。

Microsoft Azure サービス・プリンシパルの詳細については、Microsoft Azure の資料を参照してください。

VMware プラグイン

アカウント・ラベル

指定されたユーザー名/パスワード・ペアのわかりやすい名前。英数字のみを含める必要があります。

vCenter サーバー

vCenter サーバーの IP アドレスのホスト名。

ユーザー名とパスワード

vCenter サーバーにアクセスするための資格情報。

注: VMware プラグインは govmomi ライブラリーを使用し、その互換性が、プラグインが準拠する vCenter のバージョンを定義します。詳しくは、govmomi の資料を参照してください。

Google Cloud Platform プラグイン

アカウント・ラベル

指定したサービス・アカウント資格情報のわかりやすい名前。英数字のみを含める必要があります。

サービス・アカウント資格情報

サービス・アカウントの鍵を含む、Google が提供する .json ファイルの内容をコピーして貼り付けます。必要な IAM 権限は次のとおりです。

- compute.zones.list
- compute.regions.list
- compute.instances.list
- compute.images.list
- compute.disks.list
- compute.machineTypes.list
- compute.subnetworks.list

これらの権限はすべて必須です。

Google Cloud Platform サービス・アカウントについて詳しくは、Google の文書を参照してください。

注: Google Cloud Platform プラグインは、プラグインが構成された .json ファイルで指定されているプロジェクトに属する VM インスタンスを検出します。Google Cloud Platform プラグインのバージョン 1.4 以降では、サービス・アカウントがリスト可能かつ権限をを有するすべての追加プロジェクトを管理できます。プロジェクトをリストできるようにするには、リソース・マネージャー API を有効にして、サービス・アカウントがリソース・マネージャーに対する要求を発行できるようにする必要があります。つまり、*resourcemanager.projects.get* 権限を持つようにします。プロジェクトのリスティングに関して詳しくは、Google の資料を参照してください。

関連情報

プラグイン・ポータル (（ページ） *155*)

クラウド設定の理解 (（ページ） *196*)

クラウド・プラグインのインストールの確認

クラウド・プラグインのデプロイが完了すると、最初の検出の試行がすぐに開始され、検出されたリソースが BigFix コンソールの**「コンピューター」**ビューまたは BigFixWebUI の**「デバイス」** ページに数分以内に表示されます。

インストール後の正常性チェックとして、次のことを確認できます。

1. 次のデフォルト・パスにクラウド・プラグイン・ログ・ファイルが存在すること:
   - **Windows**: C:\Program Files (x86)\BigFix Enterprise\BES Plugin Portal\Plugins*Plugin_name*\Logs\log.txt
   - **Linux**: /var/opt/BESPluginPortal/Plugins/*Plugin_name*/Logs/log.txt
2. 機能中のプラグインがインストールされたクラウド・プラグイン・ログ・ファイルの内容は次のようになります。

```
2020/03/31 19:33:48 - [info] <Plugin_name> 1.1.59 starts on <OS
 platform>
2020/03/31 19:33:48 - [info] Plugin Portal with API version 1
```

```
2020/03/31 19:33:55 - [info] Refresh all: Attempting discovery
2020/03/31 19:35:53 - [info] Refresh all: Discovery returned <number>
 unique devices
```

3. BESPluginPortal プロセスの実行ステータス。
4. 次のデフォルト・パスにあるプラグイン・ポータル・ログ・ファイルの内容:
   - **Windows**: C:\Program Files (x86)\BigFix Enterprise\BES Plugin Portal \BESPluginPortal.log
   - **Linux**: /var/log/BESPluginPortal.log

   最新のプラグイン・ポータルの再起動後、ログ・ファイルには次のような行が含まれているはずです。

```
Tue, 31 Mar 2020 19:33:49 +0200 - 1222616832 - Running
 plugin'<Plugin_name>'
```

注: 検出されたリソースが存在しない限り、プラグイン・ポータル・ログ・ファイルには次のメッセージがあることが予想されます。

```
Tue, 31 Mar 2020 19:33:49 +0200 - 2383846272 - Error reading
 records from the database.
```

予期しない結果が発生した場合は、トラブルシューティング (（ページ） 295)セクションを参照してください。

クラウド・プラグインの構成

クラウド・プラグインは、主に BigFix WebUI を介して後で更新可能な基本構成でインストールされます。

検出の頻度

クラウド・プラグインが関連するクラウド・リソースの検出を実行する頻度です。クラウド・プラグインは、デフォルトの検出の頻度 120 分でインストールされます。

検出の頻度は、BigFix WebUI の**「プラグイン管理」**セクションから更新できます。クラウド・プラグインの検出の頻度を更新できる BES サポート・タスクもあります。

プロキシー・インスタンスの削除

このオプションは、WebUI から変更可能なクラウド・プラグインのブール設定で設定されます。プラグイン・ポータルが削除要求を BigFix サーバーに送信し、この設定が有効になっていれば、BigFix サーバーは物理的削除を実行します。これは、デバイスと関連するすべてのデータをデータベースから物理的に削除することを意味します。この設定が無効になっている場合 (デフォルトでは無効)、BigFix サーバーは論理削除を実行します。これは、デバイスは削除済みとしてマークされているものの、BigFix データベースには引き続き含まれていることを意味します。

この機能について詳しくは、クラウド・リソースの検出 (（ページ） 198) を参照してください。

相関インスタンスの削除

このオプションは、WebUI から変更可能なクラウド・プラグインのブール設定で設定されます。プラグイン・ポータルが削除要求を BigFix サーバーに送信し、この設定が有効になっていれば、BigFix サーバーは、プロキシー・インスタンスに関連付けられたネイティブ・インスタンスもプラグイン・ポータル削除要求から削除します。この設定はデフォルトでは無効になっています。

この機能について詳しくは、クラウド・リソースの検出 (（ページ） 198) を参照してください。

ログ

クラウド・プラグインは、標準のログ・レベルと、次のデフォルトのログ・パスでインストールされます。

- **Windows**: C:\Program Files (x86)\BigFix Enterprise\BES Plugin Portal\Plugins*Plugin_name*\Logs\log.txt
- **Linux**: /var/opt/BESPluginPortal/Plugins/*Plugin_name*/Logs/log.txt

ログ・ファイルは、サイズが 10 MB に達するとローテーションします。ローテーションされた最新の 10 個のログは、ログ・ディレクトリーにあります。

BigFix WebUI の**「プラグイン管理」**セクションでログ・パスと冗長度を更新できます。

複数の資格情報セットのサポート

クラウド・プラグインは、複数の資格情報セットを使用するように構成されている場合があります。最初のセットはインストール時に指定され、追加のセットは BigFix WebUI の**「プラグイン管理」**セクションからいつでも追加できます。

検出のたびに、クラウド・プラグインは使用可能なすべての資格情報セットを調べ、各セットで使用可能なクラウド・リソースを取得します。資格情報セットが検出試行失敗の最大回数に達すると、次回の検出時にそれが考慮されなくなります。

検出試行失敗の最大回数

これは、スキップされる前に資格情報セットが連続して検出を失敗した最大回数です。どんな場合でも検出が失敗すると回数が増えるわけではなく、誤ったパスワードや期限切れのパスワードが原因で失敗したログイン試行に関連する場合にのみ、回数が増えます。検出が成功すると、カウンターがリセットされます。プラグイン・ポータルを再起動すると、カウンターもリセットされます。

3 回連続して失敗した後、失敗した資格情報セットをスキップすると、クラウド・プラットフォーム側で設定されているアカウント・ロックアウト・ポリシーによって資格情報セットが影響を受ける可能性が低くなります。

資格情報セットが試行失敗の最大回数に達すると、次のメッセージが標準ログに書き込まれます。

```
[error] Refresh all: user 'Account Label' reached the maximum attempts (3)
 and it will be skipped
```

これが発生した場合、オペレーターは BigFix WebUI を利用してパスワードを更新する必要があります。更新された資格情報セットは、次の検出から再び考慮に入れられます。

プロキシー・サポート

AWS プラグインおよび Microsoft Azure プラグインは、インターネット経由で HTTPS を使用して関連するクラウド・サービスにアクセスできる必要があります。サポートされているプロキシー構成は異なりますが、どちらの場合もプロキシーを使用してそれを行うことができます。

AWS プラグインのプロキシーを構成する方法

AWS プラグインをインストールするとき、BigFix WebUI の Fixlet によるインストールまたは「クラウド・プラグインのインストール」ページの適切なフィールドに入力することで、プロキシーを介して検出を実行するようにプラグインを構成できます。WebUI では、インストールしたプラグインの構成を編集することで、後でプロキシーを指定することもできます。

HTTPS プロキシーの場合、カスタム CA 証明書ファイルを使用してプロキシーの SSL 証明書を検証するように AWS プラグインを設定することができます。BES サポート・タスクで、この構成を実行できます。

Microsoft Azure プラグインのプロキシーを構成する方法

Microsoft Azure プラグインがプロキシーを経由するようにするには、次のようにプロキシーを構成する必要があります。

Windows

http_proxy および *https_proxy* 環境変数を使用して、システム・レベルでプロキシーを設定する必要があります。

Linux

/etc/opt/BESPluginPortal/custom.config ファイル (このファイルが存在しない場合は手動で作成してください) で、次のキーと値のペアを指定する必要があります。

- https_proxy=*http://proxyHost:proxyPort/*
- http_proxy=*http://proxyHost:proxyPort/*

Windows と Linux の両方で、BigFix プラグイン・ポータル・サービスを再起動してプロキシー構成を有効にします。

注: Azure クラウド・プラグインに組み込まれている Azure SDK では、プロキシーをシステム・レベルで設定する必要があるため、ローカルのプラグイン・ポータルからその親のリレーへの通信も影響を受けます。直接通信を復元するには、プラグイン・ポータルで関連する BigFix 構成設定を使用してプロキシーを定義し、

関連する例外リストに親リレーを含める必要があります。詳細については、「サーバー/リレー・プロキシーの設定」を参照してください。

関連情報

プラグイン・ポータル (（ページ） 155)

クラウド設定の理解 (（ページ） 196)

クラウド・プラグイン・インスペクター

クラウド・プラグインが報告するリソースには、それらが表すインスタンスに関連する多くの情報が含まれています。このようなデータは、多数のインスペクターを介して使用可能となります。

こうしたインスペクターは、すべてのプラグインが共有する共通セクションと、単一のプラグインに限られた各プラグインのより具体的なセクションに分けられます。このトピックでは、 **共通インスペクター**について説明します。

コンピューター名

```
computer name: string
```

デバイスの表示名。

CPU

```
main processor: processor
```

メイン・プロセッサーに対応するプロセッサー・オブジェクトを返します。

```
speed of <processor>: hertz
```

Hertz のプロセッサーの速度を戻します。

```
family name of <processor>: string
```

CPU のファミリー名を返します。

データ・ソース

```
data source: string
```

データ・ソースは、データの送信元システムの名前です。

デバイス ID

```
device id: string
```

デバイス ID は、各デバイス固有の ID です。

デバイス・タイプ

```
device type: string
```

デバイスのタイプ。クラウド・プラグイン・リソースの場合、デフォルトで「クラウド」に設定されています。

DNS 名

```
dns name: string
```

DNS の名前。

ネットワーク

```
network: network
```

ネットワーク・インスペクターは、ネットワーク構成に関するデータを報告します。

```
ip interfaces of <network>: plural ip interface
```

ネットワークのすべての IP インターフェースを返します。

```
address of <ip interface>: ipv4or6 address
```

IP インターフェースの IP アドレスを返します。

```
subnet address of <ip interface>: ipv4or6 address
```

指定されたインターフェースが属するサブネット・アドレスを返します。

```
loopback of <ip interface>: boolean
```

特定のネットワーク IP インターフェイスがループバック・インターフェイスかどうかを示します。

```
adapters of <network>: plural network adapter
```

ネットワークの 1 つ以上のネットワーク・アダプター・オブジェクトを返します。

```
up of <network adapter>: boolean
```

指定されたネットワーク・アダプターが現在動作している場合は True を返します。

```
loopback of <network adapter>: boolean
```

指定されたネットワーク・アダプターがループバック・インターフェースの場合は True を返します。

```
ipv6 interfaces of <network adapter>: plural network adapter interface
```

指定されたネットワーク・アダプターの IPv6 インターフェースをネットワーク・アダプター・インターフェース・タイプとして返します。

```
address of <network adapter interface>: ipv4or6 address
```

指定されたネットワーク・アダプター・インターフェースの IP アドレスを ipv4or6 アドレス・タイプとして返します。

OS

```
operating system: operating system
```

OS インスペクターは、オペレーティング・システムに関するデータを返します。デフォルトでは名前とバージョンが報告されますが、クラウド・プラグインに応じて追加のプロパティーを使用できます。

```
name of <operating system>: string
```

OS の名前。

```
version of <operating system>: version
```

OS のバージョン。

RAM

```
ram: ram
```

マシンのランダム・アクセス・メモリーのプロパティーを調べるための ram オブジェクトを返します。クラウド・プラグインに応じて、追加のプロパティーを使用できます。

```
size of <ram>: integer
```

現在のマシン上のランダム・アクセス・メモリーのバイト数を返します。

AWS Asset Discovery プラグイン・インスペクター

AMI 起動インデックス

```
ami launch index: integer
```

AMI 起動インデックス。

関連付けの概要

```
association overview: association overview
```

関連付けの概要インスペクターは、集約された関連付けに関する状況情報を返します。

```
detailed status of <association overview>: string
```

集約された関連付けに関する詳細な状況情報。

```
status count of <association overview>: string
```

インスタンスの関連付けの数。

AWS イメージ

```
aws image: aws image
```

AWS イメージ・インスペクターは、AMI とそのプロパティーを表します。

```
id of <aws image>: string
```

AMI のID。

```
name of <aws image>: string
```

AMI の名前。

```
architecture of <aws image>: string
```

AMI のアーキテクチャー。

```
description of <aws image>: string
```

AMI 作成時に入力された説明。

```
creation date of <aws image>: string
```

AMI を作成した日時。

```
hypervisor of <aws image>: string
```

AMI のハイパーバイザー。

```
location of <aws image>: string
```

AMI の場所。

```
owner alias of <aws image>: string
```

AMI 所有者の AWS アカウントの別名または AWS アカウントID。

```
type of <aws image>: string
```

AMI のタイプ。

```
kernel id of <aws image>: string
```

イメージに関連付けられたカーネル (存在する場合)。マシン・イメージにのみ適用されます。

```
public of <aws image>: boolean
```

イメージに公開起動許可がある場合は true を返し、暗黙的および明示的な起動許可しかない場合は false を返します。

```
tags of <aws image>: plural tag
```

AMI に割り当てられたタグの配列。

```
product codes of <aws image>: plural product code
```

製品コードの配列。

```
virtualization type of <aws image>: string
```

AMI の仮想化タイプ。

```
platform details of <aws image>: string
```

AMI の請求コードに関連付けられたプラットフォームの詳細。

デバイス・マッピングのブロック

```
block device mapping: block device mapping
```

```
[block device mapping, block device mappings]: plural block device mapping
```

デバイス・マッピングのブロック・インスペクターは、ブロック・デバイス・マッピングについて説明します。各ブロック・デバイス・マッピングのデータ配列を返します。

```
device name of <block device mapping>: string
```

ブロック・デバイス・マッピングのデバイス名。

```
ebs of <block device mapping>: ebs
```

インスタンスの起動時に EBS ボリュームを自動的にセットアップするため使用されるパラメーター。

CPU パッケージ

```
cpupackage: cpupackage
```

CPU パッケージ・インスペクターは、CPU オプションに関するデータを返します。

```
core of <cpupackage>: integer
```

インスタンスのコアの数。

```
thread of <cpupackage>: integer
```

コアあたりのスレッド数。

資格情報ラベル

```
credentials label: string
```

資格情報ラベルは、プラグインのインストール時にユーザーによって定義されたカスタム・ストリングです。ユーザーの資格情報セットを一意で識別します。

資格情報のロール

```
credentials role: string
```

インスタンスが検出された IAM ロールのロール ARN。資格情報にロールが指定されていない場合、値は空のままです。

EBS

```
ebs: ebs
```

```
ebses: plural ebs
```

EBS インスペクターは、ブロック・デバイス・マッピングで EBS ボリュームをセットアップするために使用されるパラメーターを記述します。

```
attach time of <ebs>: string
```

添付が開始された時点のタイム・スタンプ。

```
delete on termination of <ebs>: boolean
```

インスタンス終了時にボリュームが削除されるかどうかを示します。

```
status of <ebs>: string
```

添付の状態。

```
volume id of <ebs>: string
```

EBS ボリュームのID。

ENA 有効化

```
ena enabled: boolean
```

ENA を使用した拡張ネットワーキングを使用可能にするかどうかを指定します。

ハイパーバイザー

```
hypervisor: string
```

インスタンスのハイパーバイザー。

イメージ ID

```
image id: string
```

インスタンスの起動に使用される AMI イメージのID。

インスタンス ID

```
instance id: string
```

インスタンスの固有 ID。

インスタンス・タイプ

```
instance type: string
```

インスタンスのタイプ。

キー名

```
key name: string
```

インスタンスが起動された鍵ペアの名前 (存在する場合)。

起動時刻

```
launch time: string
```

インスタンスが起動した時刻。

所有者 ID

```
owner id: integer
```

インスタンス・オーナーの ID。

Placement (配置)

```
placement: placement
```

インスタンスが起動した場所 (該当する場合)。

```
affinity of <placement>: string
```

専用ホスト上のインスタンスの親和性設定。

```
availability zone of <placement>: string
```

インスタンスのアベイラビリティー・ゾーン。

```
group name of <placement>: string
```

インスタンスの配置グループの名前。

```
host id of <placement>: string
```

インスタンスが存在する専用ホストのID。

```
partition number of <placement>: string
```

インスタンスが含まれるパーティションの番号。配置グループ戦略がパーティションに設定されている場合にのみ有効です。

```
spread domain of <placement>: string
```

インスタンスのスプレッド・ドメイン。

```
tenancy of <placement>: string
```

インスタンスのテナンシー (インスタンスが VPC で実行されている場合)。

プライベート DSN 名

```
private dns name: string
```

インスタンスに割り当てられたプライベート DNS ホスト名。

プライベート IP アドレス

```
private ip address: string
```

インスタンスに割り当てられたプライベート IPv4 アドレス。

製品コード

```
product codes: plural product codes
```

製品コード・インスペクターは、製品コードを記述するオブジェクトを返します。

```
id of <product code>: string
```

製品コード。

```
type of <product code>: string
```

製品コードのタイプ。

パブリック IP アドレス

```
public ip address: string
```

インスタンスに割り当てられたパブリック IPv4 アドレスまたはキャリア IP アドレス。

パブリック DNS 名

```
public dns name: string
```

インスタンスに割り当てられたパブリック DNS ホスト名。

領域

```
region: string
```

インスタンスが存在する AWS リージョン。

要求 ID

```
requester id: integer
```

代理でインスタンスを起動した要求元の ID。

ルート・デバイス・タイプ

```
root device type: string
```

AMI で使用されるルート・デバイス・タイプ。

セキュリティ・グループ

```
security groups: plural security group
```

セキュリティ・グループ・インスペクターは、インスタンスに割り当てられたセキュリティ・グループに関するデータを返します。

```
id of <security group>: string
```

セキュリティー・グループのID。

```
name of <security group>: string
```

セキュリティー・グループの名前。

状態

```
state: string
```

インスタンスの現在の状態。

サブネット ID

```
subnet id: string
```

インスタンスが実行されているサブネットの ID。

システム管理者

```
system manager: system manager
```

システム管理者インスペクターは、インスタンスで使用可能な場合、AWS System Manager に関する情報を返します。システム管理者インスペクターは、ネイティブ API を介した AWS での BigFix エージェント・デプロイメントに必要です。詳細については、BigFix クラウド・リソースへのエージェントのインストール (（ページ） 285)を参照してください。

```
activation id of <system manager>: string
```

サーバーまたは VM の登録時に Systems Manager によって作成された有効化 ID。

```
agent version of <system manager>: string
```

インスタンスに存在する SSM エージェントのバージョン (存在する場合)。BigFix エージェントのインストールの場合、存在します。

```
association overview of <system manager>: association overview
```

集約された関連付けの概要を返します。

```
association status of <system manager>: string
```

関連付けの状況。

```
computer name of <system manager>: string
```

インスタンスの完全修飾ドメイン名。

```
ip address of <system manager>: string
```

インスタンスの IP アドレス。

```
iam role of <system manager>: string
```

オンプレミス Systems Manager 管理対象インスタンスに割り当てられた IAM ロール。

```
instance id of <system manager>: string
```

インスタンス ID。

```
latest version of <system manager>: boolean
```

インストールされている SSM エージェントのバージョン。

```
last association execution of <system manager>: string
```

関連付けが最後に実行された日付。

```
last ping of <system manager>: string
```

SSM エージェントが System Manager サービスに ping した日時。

```
last successful association of <system manager>: string
```

関連付けが正常に実行された最後の日付。

```
name of <system manager>: string
```

オンプレミス・サーバーまたは仮想マシンが Systems Manager 管理対象インスタンスとして有効化されるときに、そのサーバーまたは仮想マシンに割り当てられる名前。

```
ping status of <system manager>: string
```

SSM エージェントの接続状況。BigFix エージェントをインストールする場合は「オンライン」にする必要があります。

```
platform name of <system manager>: string
```

インスタンスのオペレーティング・システム・プラットフォームの名前。

```
platform type of <system manager>: string
```

インスタンスのオペレーティング・システム・プラットフォームのタイプ。

```
platform version of <system manager>: string
```

インスタンスのオペレーティング・システム・プラットフォームのバージョン。

```
registration date of <system manager>: string
```

サーバーまたは VM が AWS に管理対象インスタンスとして登録された日付。

```
resource type of <system manager>: string
```

リソース・タイプ (EC2 または管理対象インスタンスのいずれか)。

タグ

```
tags: plural tag
```

タグ・インスペクターは、インスタンスに関連付けられたタグの配列を返します。

```
key of <tag>: string
```

タグのキー。

```
value of <tag>: string
```

タグの値。

稼働時間

```
uptime: integer
```

インスタンスが起動されてからの時間 (判明している場合)。

仮想化タイプ

```
virtualization type: string
```

インスタンスで使用される仮想化のタイプ。

VPC ID

```
vpc id: string
```

インスタンスが実行されている VPC のID。

Azure Asset Discovery プラグイン・インスペクター

Azure イメージ

```
azure image: azure image
```

Azure イメージ・インスペクターは、インスタンスの Azure イメージに関するデータを返します。

```
publisher of <azure image>: string
```

イメージ・パブリッシャーを指定します。

```
offer of <azure image>: string
```

仮想マシンの作成に使用されるプラットフォーム・イメージまたはマーケットプレイス・イメージのオファーを指定します。

```
sku of <azure image>: string
```

イメージ SKU。

```
version of <azure image>: string
```

仮想マシンの作成に使用されるプラットフォーム・イメージまたはマーケットプレイス・イメージのバージョンを指定します。使用できる形式は Major.Minor.Build または 'latest' です。

資格情報ラベル

```
credentials label: string
```

資格情報ラベルは、プラグインのインストール時にユーザーによって定義されたカスタム・ストリングです。ユーザーの資格情報セットを一意で識別します。

インスタンス ID

```
[instance id, instances id]: string
```

```
instance id: string
```

インスタンス ID はインスタンス固有の識別子で、VMID (128 ビット識別子) に対応します。

インスタンス・タイプ

```
instance type: string
```

仮想マシンのサイズを指定します。

ライセンス・タイプ

```
license type: string
```

使用されているイメージまたはディスクがライセンス取得済みでオンプレミスであることを指定します。この要素は、Windows Server オペレーティング・システムを含むイメージにのみ使用されます。

Linux 構成

```
linux configuration of <operating system>: linux configuration
```

仮想マシンのLinuxオペレーティング・システム設定を指定します。

```
disable password authentication of <linux configuration>: boolean
```

パスワード認証を無効にするかどうかを指定します。

```
provision vm agent of <linux configuration>: boolean
```

仮想マシン・エージェントを仮想マシンでプロビジョンするかどうかを示します。

ネットワーク

ネットワーク・インスペクターは、共通セクションで定義されます。Microsoft Azure プラグインの追加の IP インターフェース・プロパティーを次に示します。

```
public address of <ip interface>: ipv4or6 address
```

IP 構成にバインドされているパブリック IP アドレス。

オペレーティング・システム

オペレーティング・システム・インスペクターは、共通セクションで定義されます。Microsoft Azure プラグインの追加プロパティーを次に示します。

```
admin username of <operating system>: string
```

管理者アカウントの名前を指定します。

```
windows configuration of <operating system>: windows configuration
```

仮想マシンの Windows オペレーティング・システム設定を指定します。

```
linux configuration of <operating system>: linux configuration
```

仮想マシンの Linux オペレーティング・システム設定を指定します。

OS ディスク

```
os disk: os disk
```

OS ディスク・インスペクターは、仮想マシンが使用するオペレーティング・システム・ディスクに関する情報を報告します。

```
caching of <os disk>: string
```

キャッシュ要件を指定します。

```
create option of <os disk>: string
```

インスタンスの作成方法を指定します。

```
disk size gb of <os disk>: string
```

報告されるディスクのサイズ (ギガバイト単位)。

```
storage account type of <os disk>: string
```

管理対象ディスクのストレージ・アカウント・タイプを指定します。

```
name of <os disk>: string
```

ディスク名。

```
os type of <os disk>: string
```

ディスクに含まれる OS タイプ。

プロビジョニング状態

```
provisioning state: provisioning state
```

プロビジョニング状態インスペクターは、プロビジョニングの状態に関するデータを返します。

```
status of <provisioning state>: string
```

状況に関するローカライズ可能な短いラベル。

```
time of <provisioning state>: string
```

状況の時刻。

領域

```
region: string
```

リソースの場所。

リソース・グループ

```
resource group: string
```

仮想マシンのリソース・グループ。

状況

```
status: string
```

仮想マシンの電源状態。

タグ

```
tags: plural tag
```

タグ・インスペクターは、インスタンスに関連付けられたタグの配列を返します。

```
key of <tag>: string
```

タグのキー。

```
value of <tag>: string
```

タグの値。

ウルトラ SSD 有効化

```
ultra ssd enabled: boolean
```

VM または VMSS でストレージ・アカウント・タイプが UltraSSD_LRS の 1 つ以上の管理対象データ・ディスクを有する機能を使用可能または使用不可にするフラグ。

Windows 構成

```
windows configuration of <operating system>: windows configuration
```

仮想マシンの Windows オペレーティング・システム設定を指定します。

```
provision vm agent of <windows configuration>: boolean
```

仮想マシン・エージェントを仮想マシンでプロビジョンするかどうかを示します。

```
enable automatic updates of <windows configuration>: boolean
```

Windows 仮想マシンで自動更新を有効にするかどうかを指定します。

```
timezone of <windows configuration>: string
```

仮想マシンのタイム・ゾーンの名前を指定します。

VM エージェント

```
vm agent: vm agent
```

VM エージェント・インスペクターは、仮想マシンに存在する VM エージェントに関するデータを返します (使用可能な場合)。ネイティブ API を使用して Azure に BigFix エージェントをデプロイするには、VM エージェント・インスペクターが必要です。詳細については、BigFix クラウド・リソースへのエージェントのインストール (（ページ） 285)を参照してください。

```
version of <vm agent>: string
```

VM エージェントのバージョン。BigFix エージェントのインストールに存在します。

```
status of <vm agent>: string
```

VM エージェントの状況。BigFix エージェントのインストールでは「Ready (作動可能)」である必要があります。

GCP Asset Discovery プラグイン・インスペクター

IP 転送が可能

```
can ip forward: boolean
```

インスタンスが、一致しない宛先 IP またはソース IP を持つパケットの送受信を実行できるかどうかを示します。

コンピューター名

```
computer name: string
```

リソースの最初の作成時に、クライアントによって提供されるリソースの名前。

CPU プラットフォーム

```
cpu platform: string
```

現在のインスタンスによって使用される CPU プラットフォーム。

作成時のタイムスタンプ

```
creation timestamp: string
```

作成時のタイム・スタンプ。

資格情報ラベル

```
credentials label: string
```

資格情報ラベルは、プラグインのインストール時にユーザーによって定義されたカスタム・ストリングです。ユーザーの資格情報セットを一意で識別します。

データ・ソース

```
data source: string
```

データ・ソースは、データの送信元システム名です。VMware プラグインの場合、デフォルトで「プロキシー - Google Cloud Platform」に設定されます。

削除保護

```
deletion protection: boolean
```

リソースを削除から保護するかどうか。

説明

```
description: string
```

このリソースのオプションの説明。

ディスク

```
disks: plural gcp disk
```

ディスク・インスペクターには、インスタンス・ストレージに関する情報が含まれています。

```
type of <gcp disk>: string
```

ディスクのタイプ (SCRATCH または PERSISTENT) を指定します。

```
mode of <gcp disk>: string
```

このディスクを接続するモード (READ_WRITE または READ_ONLY)。

```
source of <gcp disk>: string
```

既存の永続ディスク・リソースに対する有効な部分 URL または完全な URL を指定します。

```
device name of <gcp disk>: string
```

一意のデバイス名を指定します。指定しない場合、サーバーは persistent-disk-x という形式のデフォルト名を選択します。x は Google Compute Engine によって割り当てられた番号です。このフィールドは永続ディスクにのみ適用されます。

```
index of <gcp disk>: integer
```

このディスクに対するゼロ・ベースのインデックス。0 はブート・ディスク用に予約されています。

```
boot of <gcp disk>: boolean
```

これがブート・ディスクであることを示します。

```
auto delete of <gcp disk>: boolean
```

インスタンスの削除時にディスクを自動削除するかどうかを指定しますが、ディスクの削除時には削除しません。

```
interface of <gcp disk>: string
```

このディスク (SCSI または NVME) の接続に使用するディスク・インターフェースを指定します。

```
size gb of <gcp disk>: integer
```

永続ディスクのサイズ (GB)。

```
description of <gcp disk>: string
```

リソースに関する任意の説明。

```
zone of <gcp disk>: string
```

ディスクが存在するゾーンの URL。

```
status of <gcp disk>: string
```

ディスク作成の状況。可能な値
は、CREATING、DELETING、READY、FAILED、RESTORING です。

```
source image of <gcp disk>: gcp image
```

このディスクの作成に使用されるソース・イメージ。

```
creation timestamp of <gcp disk>: string
```

作成時のタイム・スタンプ。

ホスト名

```
hostname: string
```

インスタンスの RFC1035 準拠のホスト名を指定します。

イメージ

```
archive size bytes of <gcp image>: integer
```

Google Cloud Storage に保存されているイメージ tar.gz アーカイブのサイズ (バイト単位)。

```
creation timestamp of <gcp image>: string
```

作成時のタイム・スタンプ。

```
source disk of <gcp image>: string
```

このイメージの作成に使用されるソース・ディスクの URL。

```
size gb of <gcp image>: integer
```

永続ディスクに復元したときのイメージのサイズ (GB 単位)。

```
family of <gcp image>: string
```

このイメージが属するイメージ・ファミリーの名前。

```
id of <gcp image>: integer
```

リソースの固有 ID 。

```
name of <gcp image>: string
```

リソースの名前。リソースの作成時にクライアントによって提供されます。

```
status of <gcp image>: string
```

イメージの状況。可能な値は、DELETING、FAILED、PENDING、READY です。

```
source type of <gcp image>: string
```

このディスクの作成に使用されるイメージのタイプ。

```
description of <gcp image>: string
```

リソースに関する任意の説明。

インスタンス ID

```
[instance id, instances id]: string
```

リソースの固有 ID 。この ID はサーバーによって定義されます。

ラベル

```
labels: plural tag
```

インスタンスに適用するラベル。

```
key of <tag>: string
```

ラベルのキー。

```
value of <tag>: string
```

ラベルの値。

マシン・タイプ

```
machine type: string
```

このインスタンスに使用するマシン・タイプ・リソースの完全または部分的なURL (`zones/zone/machineTypes/machine-type` 形式)。これは、インスタンスの作成時にクライアントによって提供されます。

最小 CPU プラットフォーム

```
minimum cpu platform: string
```

VM インスタンスの最小CPUプラットフォームを指定します。

ネットワーク

```
network tags: plural string
```

ネットワーク・インスペクターは、共通セクションで定義されます。タグの配列。

```
name of <ip interface>: string
```

ネットワーク・インターフェースの名前。

```
gateway address of <ip interface>: ipv4or6 address
```

サブネットワークの外部の宛先アドレスに到達するためのデフォルト・ルートのゲートウェイ・アドレス。

```
external address of <ip interface>: ipv4or6 address
```

このインスタンスに関連付けられた外部 IP アドレス。

```
access configurations of <ip interface>: plural access configuration
```

インスタンス・ネットワーク・インターフェースに添付されるアクセス構成。

```
type of <access configuration>: string
```

構成のタイプ。

```
name of <access configuration>: string
```

このアクセス構成の名前。

```
nat ip of <access configuration>: ipv4or6 address
```

このインスタンスに関連付けられた外部 IP アドレス。

```
network tier of <access configuration>: string
```

これは、対応するアクセス構成の設定に使用されるネットワーキング層を示します。

```
domain name of <access configuration>: string
```

パブリック PTR レコードの DNS ドメイン名。

プロジェクト ID

```
project id: string
```

インスタンスのプロジェクトの ID。

自己リンク

```
self link: string
```

リソースのサーバー定義 URL。

制限付き開始

```
start restricted: boolean
```

Compute Engine が疑わしいアクティビティーを検出したために VM の開始が制限されているかどうか。

状況

```
status: string
```

インスタンスの状況。

状況メッセージ

```
status message: string
```

状況を説明する、可読のメッセージ。

ゾーン

```
zone: string
```

インスタンスが存在するゾーンのURL。

VMware Asset Discovery プラグイン・インスペクター

Bios

```
vm bios: bios
```

bios プロパティーの uuid を含んで返される、vm bios インスペクター。

```
uuid of <bios>: string
```

`12345678-abcd-1234-cdef-123456789abc` 形式の 16 進数ストリングで表現される、仮想マシンの 128bit SMBIOS UUID。

Bios (ゲスト)

```
version of <vm bios>: string
```

物理シャーシの現在の BIOS バージョン。

```
release date of <vm bios>: time
```

BIOS のリリース日。

Bios (ホスト)

```
host bios : <host bios>
```

ホストのシステム BIOS に関する情報を返す、Host BIOS インスペクター。

```
model of <host bios>: string
```

ホスト BIOS モデル識別。

```
platform of <host bios>: platform
```

ホスト BIOS のプラットフォーム・オブジェクトを返します。

```
serial number of <platform>: string
```

プラットフォーム・オブジェクトから取得された、ホスト BIOS のシリアル番号。

コンピューター名

```
computer name: string
```

デバイスの表示名。ゲスト・デバイスの仮想マシンの名前とホスト・デバイスのホスト名。

資格情報ラベル

```
credentials label: string
```

資格情報ラベルは、プラグインのインストール時にユーザーによって定義されたカスタム・ストリングです。ユーザーの資格情報セットを一意で識別します。

カスタム属性

```
custom attributes: plural custom attribute
```

このインスペクターは、カスタム属性を返します。値を格納するための基本タイプの配列。

```
attribute of <custom attribute>: string
```

カスタム属性の属性名。

```
value of <custom attribute>: string
```

属性値のストリング値。

データ・ソース

```
data source: string
```

データ・ソースは、データの送信元システム名です。VMware プラグインの場合、デフォルトでは「プロキシー - VMware」に設定されます。

デバイス

```
device: device
```

このノードの全体的なアラーム状況。vSphere API 5.0 より後のリリースでは、vSphere Server は、このプロパティーのプロパティー・コレクター更新通知を生成しない場合があります。

デバイス (ゲスト)

```
status of <device>: string
```

仮想マシンの現在の電源状態。

```
state of <device>: string
```

仮想マシンの現在の状態。

```
vm uuid of <device>: string
```

仮想マシンの VC 固有 ID。

```
bios uuid of <device>: string
```

仮想マシンの BIOS 識別。

```
description of <device>: string
```

仮想マシンの説明。

```
vm path name of <device>: string
```

仮想マシンの構成ファイルへのパス名。

```
clean power off of <device>: string
```

電源がオフになっている仮想マシンの場合、仮想マシンの最後のシャットダウンが正常な電源オフだったかどうかを示します。仮想マシンが実行中か中断中かを未設定にします。

```
virtual disk count of <device>: integer
```

仮想マシンに接続されている仮想ディスクの数。

```
ethernet card count of <device>: integer
```

仮想ネットワーク・アダプターの数。

```
modification time of <device>: time
```

最後に実行されたタスク。

```
template of <device>: boolean
```

このデータ・オブジェクト・タイプは、テンプレート仮想マシン構成ファイルを記述します。

デバイス (ホスト)

```
bios release date of <device>: time
```

bios のリリース日。

```
bios version of <device>: string
```

物理シャーシの現在の bios バージョン。

```
bios uuid of <device>: string
```

ハードウェア BIOS の識別情報。

```
file system type of <device>: string
```

サポートされるファイル・システム・ボリューム・タイプのリストを表します。

```
maintenance mode of <device>: boolean
```

ホストが保守モードであるかどうかを示すフラグ。このフラグは、ホストが保守モードに入ったときに設定されます。メンテナンス・モードの開始フェーズでは設定されません。

```
model of <device>: string
```

システム・モデル の識別情報。

```
network adapters of <device>: string
```

ネットワーク・アダプターの数。

```
pending restart of <device>: boolean
```

構成の変更が原因でホストがリブートを必要とするかどうかを示します。

```
port number of <device>: string
```

ポート番号。

```
status of <device>: string
```

ホストの全体的なアラーム状況。

```
vendor of <device>: string
```

ベンダー・タイプ。

```
vmotion of <device>: boolean
```

このホストで VMotion が有効かどうかを示すフラグ。

```
vswitches of <device>: plural vswitches
```

仮想スイッチ。

```
portgroups of <vswitches>: plural string
```

仮想スイッチのポート・グループ。

```
name of <vswitches>: string
```

仮想スイッチの名前。

```
hostd log enabled of <device>: boolean
```

hostd ログ・ファイルが存在するかどうかを示すフラグ。

```
messages log enabled of <device>: boolean
```

messages ログ・ファイルが存在するかどうかを示すフラグ。

```
vmkernel log enabled of <device>: boolean
```

vmkernel ログ・ファイルが存在するかどうかを示すフラグ。

デバイス ID

```
device id: string
```

デバイス ID は、各デバイス固有の ID です。VMware ゲスト・デバイスの場合、これは仮想マシンの VC 固有 ID であるインスタンス UUID に対応します。VMware ホストの場合、接頭部「Host - 」が付加されたホスト名に対応します。

ディスク

```
disk: disk
```

マシンが使用するストレージに関するデータを返すディスク・インスペクター。

```
provisioned storage of <disk> : string
```

すべてのデータストア上のこの仮想マシンにコミットされたストレージ・スペースとコミットされていないストレージ・スペースの合計 (バイト単位)。

```
used storage of <disk> : string
```

すべてのデータストアにわたってこの仮想マシンにコミットされた合計ストレージ・スペース (バイト単位)。

```
unshared storage of <disk> : string
```

他の仮想マシンと共有されていない、すべてのデータストアにわたって仮想マシンが占めるストレージ・スペースの合計 (バイト単位)。

ドライブ (ゲスト)

```
drive: drive
```

```
drives: plural drive
```

ファイル・システムに関するデータを返すドライブ・インスペクター。

```
name of <drive>: string
```

ゲスト・オペレーティング・システム内の仮想ディスクの名前。例: C:\

```
free space of <drive>: integer
```

ディスクの空きスペース (バイト単位)。これは、VMware ツールによって取得されます。

```
total space of <drive>: integer
```

ディスクの合計容量 (バイト単位)。これは、仮想マシン構成の一部です。

ゲスト VM

```
 guest vms: plural guest vm
```

これらのインスペクターは、ゲスト仮想マシンのプロパティーを取得します。

```
name of <guest vm>: string
```

仮想マシンで構成されたゲスト名。

```
network of <guest vm>: network
```

仮想マシンで構成されたネットワーク。

```
operating system of <guest vm>: operating system
```

仮想マシン上のオペレーティング・システム。

```
host of <guest vm>: host
```

ホスト名。

```
 tool of <guest vm>: tool
```

仮想マシン上のツール。

```
processor of <guest vm>: processor
```

仮想マシン上のプロセッサー。

```
ram of <guest vm>: ram
```

仮想マシン上の RAM。

```
usage statistic of <guest vm>: usage statistic
```

仮想マシンの使用状況統計。

```
drives of <guest vm>: drive
```

仮想マシンのドライブ。

```
device of <guest vm>: device
```

仮想マシンで構成されたデバイス。

```
snapshots of <guest vm>: snapshot
```

仮想マシンのスナップショット。

ハードウェア

```
hardware: hardware
```

このインスペクターは、インスタンスの VMware 表現のハードウェア・タイプに関するプロパティーを保持します。デフォルトでは、仮想プロパティーとプロキシー・プロパティーは、クラウド・プラグイン表現で true を返します。

```
virtual of <hardware>: boolean
```

クライアントが仮想マシンで実行されている場合は true を返します。

```
proxied of <hardware>: boolean
```

デバイスが BES プラグイン・ポータルにレポートする場合、true を返します。それ以外の場合、false を返します。

ホスト (ゲスト)

```
host: host
```

現在のゲストのホストの名前を報告するホスト・インスペクター。

```
name of <host>: string
```

ホストの名前。

サービス (ホスト)

```
[host service, host services]: plural host service
```

ホスト上で実行される構成済みの各サービスに関する情報を返すホスト・サービス・インスペクター。

```
key name of <host service> : string
```

サービスの簡易 ID。

```
label of <host service> : string
```

サービスのラベルを表示します。

```
policy of <host service> : string
```

サービス・アクティベーション・ポリシー。

```
required of <host service> : string
```

サービスが必須であり、使用不可にできないかどうかを示すフラグ。

```
running status of <host service> : string
```

サービスが現在実行中であるかどうかを示すフラグ。

ネットワーク

ネットワーク・インスペクターは、共通セクションで定義されます。VMware クラウド・プラグインでは、以下のプロパティーを使用できます。

```
ip interfaces of <network>: plural ip interface
```

ネットワークのすべての IP インターフェースを返します。

```
address of <ip interface>: string
```

ゲスト・オペレーティング・システムに割り当てられたプライマリー IP アドレス (既知の場合)。それ以外の場合は、優先されるアドレスです。

```
loopback of <ip interface>: boolean
```

意図的な処理や変更を行わずに、電子信号またはデジタル・データ・ストリームをソースに戻すルーティング。

```
mac address of <ip interface>: string
```

アダプターの MAC アドレス。

ネットワーク (ゲスト)

VNIC インスペクターは、ネットワーク・アダプターに関するゲスト情報を提供します (既知の場合)。

```
vnics of <network>: plural vnic
```

ネットワークの VNIC を返します。

```
vnic counter of <network>: integer
```

仮想ネットワーク ID カード カウンターを表します。

```
name of <vnic>: string
```

デバイスのラベル。

```
up of <vnic>: boolean
```

仮想デバイスが接続されているかどうかを示すフラグ。

```
mac address of <vnic>: string
```

アダプターの MAC アドレス。

```
[address, addresses] of <vnic>: plural string
```

1 つ以上の手動 (静的) 割り当て済み IP アドレスで、指定されたインターフェースで設定する 0 個以上の IP アドレス。

```
dhcp enabled of <vnic>: boolean
```

DNS 構成の構成に動的ホスト制御プロトコル (DHCP) を使用するかどうかを示します。

```
connected of <vnic>: boolean
```

仮想デバイスが接続されているかどうかを示すフラグ。

```
device config id of <vnic>: integer
```

対応する仮想デバイスへのリンク。

```
label of <vnic>: string
```

VNIC の表示ラベル。

```
address counter of <vnic>: string
```

検出されたアドレスの数。

```
vnic counter of <network>: string
```

検出された VNIC の数。

OS

オペレーティング・システム・インスペクターは、共通セクションで定義されます。

```
name of <operating system>: string
```

バージョン情報を含む完全な製品名 (既知の場合)。それ以外の場合は、製品名の短縮形。

```
version of <operating system>: string
```

ドット区切りのバージョン。

OS (ゲスト)

```
[family, families] of <operating system>: string
```

ゲスト・オペレーティング・システム・ファミリー (既知の場合)。

```
id of <operating system>: string
```

ゲスト・オペレーティング・システム ID の短縮名 (既知の場合)。

OS (ホスト)

```
api version of <operating system>: version
```

ドット区切りの API のバージョン。

```
api type of <operating system>: string
```

サービス・インスタンスがスタンドアロン・ホストを表すかどうかを示します。サービス・インスタンスがスタンドアロン・ホストを表す場合、そのサービス・インスタンスの物理インベントリーはその単一ホストに固定されます。VirtualCenter サーバーは、単一ホスト上で追加機能を提供します。

```
boot time of <operating system>: time
```

ホストがブートされた時刻を表します。

```
build number of <operating system>: integer
```

OS ホストのビルド番号。

```
license product name of <operating system>: string
```

ライセンス製品の名前。

```
license product version of <operating system>: string
```

ライセンス製品のバージョン。

```
locale build of <operating system>: string
```

現在のセッションのロケールのビルド番号。通常、これは通常の製品ビルドからのローカライズの変更を反映した小さな数字です。

```
locale version of <operating system>: string
```

現行セッションのロケールのメッセージ・カタログのバージョン。

```
major version of <operating system>: version
```

ドット区切りの OS バージョン・プロパティーの最初の番号。

```
minor version of <operating system>: version
```

ドット区切りの OS バージョン・プロパティーの 2 番目の番号。

```
os type of <operating system>: string
```

オペレーティング・システム・アーキテクチャーのバージョン。

```
product line id of <operating system>: string
```

製品ラインの固有 ID である製品 ID。

```
vendor of <operating system>: string
```

この製品のベンダーの名前。

パッチ状況 (ホスト)

```
[patch status, patch statuses]: plural patch status
```

このインスペクターは、ESXi のパッチ適用に関する情報を収集します。

```
package name of <patch status>: string
```

ID は、一意的にページを識別します。

```
package install date of <patch status>: string
```

パッケージがインストールされた時刻。ステートレス・インストールの自動デプロイでは、値が設定されていません。

```
package version of <patch status>: string
```

ソフトウェア・パッケージを一意的に識別するバージョン文字列。

```
package vendor of <patch status>: string
```

ソフトウェア・パッケージを作成した法人。

```
package acceptance of <patch status>: string
```

受け入れレベルの定義。詳しくは、こちらを参照してください。

プロセッサー

```
processor: processor
```

プロセッサー・インスペクターは、共通セクションで定義されます。マシンで定義されたプロセッサー・オブジェクトを返します。

```
speed of <processor>: hertz
```

Hertz のプロセッサーの速度を戻します。デフォルトは -1 です。

```
family name of <processor>: string
```

CPU のファミリー名を返します。VMware クラウド・プラグインの場合、空のままです。

プロセッサー (ゲスト)

```
count of <processor>: integer
```

仮想マシン内のプロセッサーの数。

```
resource settings of <processor>: resource setting
```

リソース設定は、指定された種類のリソースに対する予約済みリソース要件と制限 (最大許容使用量) を指定します。CPU割り振りは MHz で指定されます。

プロセッサー (ホスト)

```
logical processor count of <processor>: integer
```

ホスト上の物理 CPU スレッドの数。

```
physical processor count of <processor>: integer
```

ホスト上の物理 CPU パッケージの数。

```
power management currenty policy of <processor>: string
```

現在の CPU 電源管理ポリシーに関する情報。

```
power management hardware support of <processor>: string
```

サポートされる CPU 電源管理に関する情報。

```
total processor core count of <processor>: integer
```

ホスト上の物理 CPU コアの数。

```
cpu packages of <processor>: plural cpu package
```

プロセッサーの CPU パッケージ。

```
cpu pkg bus hz of <cpu package>: integer
```

HZ 単位のバス速度。

```
cpu pkg description of <cpu package>: string
```

CPU のストリング要約の説明。

```
cpu pkg hz of <cpu package>: integer
```

HZ 単位の CPU 速度。

```
cpu pkg vendor of <cpu package>: string
```

CPU のベンダー名。現在可能な名前は、「Intel」、「AMD」、「arm」、「hygon」、または「unknown」です。

RAM

RAM インスペクターは、共通セクションで定義されます。

```
size of <ram>: integer
```

現在のマシン上のランダム・アクセス・メモリーのバイト数を返します。

RAM (ゲスト)

ゲスト・メモリー要件を超える仮想マシンによって使用されるメモリー・リソースの量 (バイト単位)。この値は、仮想マシンがメモリー・リソース割り振り機能をサポートするホストに登録されている場合にのみ設定されます。電源がオフの VM の場合、これは登録済みホスト上の VM の電源をオンにするために必要な最小オーバーヘッドです。電源がオンの VM の場合、これは現在のオーバーヘッド予約であり、ほとんどの場合、最小オーバーヘッドよりも大きく、時間とともに増加する値です。

```
overhead of <ram>: integer
```

ゲスト・メモリー要件を超える仮想マシンによって使用されるメモリーのバイト数。

```
resource settings of <ram>: resource setting
```

リソース設定は、特定の種類のリソースに対する予約済みリソース要件と、制限 (最大許容使用量) を指定します。メモリー割り振りは MB 単位で指定されます。

リソース設定

以下のプロパティーは、RAM およびプロセッサーに共通です。

```
reservation of <resource setting>: integer
```

仮想マシンまたはリソース・プールで使用可能であることが保証されているリソースの量。予約済みリソースは、使用されない場合は無駄になりません。使用率が予約より少ない場合は、実行中の他の仮想マシンがリソースを使用できます。単位はメモリーの場合は MB、CPUの場合は MHz です。

```
limit of <resource setting>: string
```

使用可能なリソースがある場合でも、仮想マシン/リソース・プールの使用率はこの制限を超えません。これは通常、使用可能なリソースに関係なく、仮想マシン/リソース・プールの一貫性のあるパフォーマンスを確保するために使用されます。-1 に設定すると、リソース使用量に固定の制限はありません (使用可能なリソースと共有によってのみ制限されます)。単位はメモリーの場合は MB、CPUの場合は MHz です。

```
overhead limit of <resource setting>: string
```

許可される最大オーバーヘッド・メモリー。電源がオンになっている仮想マシンの場合、オーバーヘッド・メモリー予約は、その overheadLimit より大きい値にはできません。このプロパティーは、電源オンの仮想マシンにのみ適用でき、リブートしても永続化されません。このプロパティーは、リソース・プールには適用されません。-1 に設定すると、予約に制限はありません。単位は MB です。

```
shares of <resource setting>: string
```

割り振りレベル。このレベルは、単純な共有ビューです。レベルは、あらかじめ決められた共有用の数値に対応しています。共有値が事前定義サイズにマップされない場合、レベルはカスタムとして設定されます。

スナップショット (ゲスト)

```
snapshots: plural snapshot
```

スナップショット・インスペクターは、スナップショットによって生成されたすべての読み取り専用データをレポートします。

```
creation time of <snapshot>: time
```

スナップショットが最後に更新された日付と時刻。

```
description of <snapshot>: string
```

スナップショットの説明。

```
name of <snapshot>: string
```

スナップショットの名前。

```
id of <snapshot>: integer
```

このスナップショットを仮想マシンの他のスナップショットと区別する固有 ID。

```
quiesced of <snapshot>: boolean
```

スナップショットが「静止」オプションを使用して作成されたかどうかを示すフラグ。「静止」オプションでは、ファイル・システムの一貫性のある状態が確保されます。

```
replay supported of <snapshot>: boolean
```

このスナップショットが、再生可能な仮想マシン上の記録セッションに関連付けられているかどうかを示すフラグ。

```
type of <snapshot>: string
```

このスナップショットの管理対象オブジェクト・タイプの名前。

```
value of <snapshot>: string
```

このスナップショットの管理対象オブジェクトの特定のインスタンス。

```
vm state of <snapshot>: string
```

このスナップショットが取得されたときの仮想マシンの電源状態。

```
children of <snapshot>: snapshots
```

このスナップショットが親であるすべてのスナップショットのスナップショット・データ。

ツール (ゲスト)

```
tool : <tool>
```

ツール・インスペクターは、VMware ツールに関するデータをレポートします。

```
status of <tool>: string
```

VMware ツールの現在のバージョン状況 (既知の場合)。

```
running status of <tool>: string
```

ゲスト・オペレーティング・システムでの VMware ツールの現在の実行状況 (既知の場合)。

```
version of <tool>: string
```

VMware ツールの現行バージョン (既知の場合)。

```
version status of <tool>: string
```

ゲスト・オペレーティング・システムでの VMware ツールの現在のバージョン状況 (既知の場合)。

使用状況統計 (ゲスト)

```
usage statistic: usage statistic
```

使用状況統計インスペクターは、仮想マシンの統計セットを返します。

```
overall cpu of <usage statistic>: hertz
```

基本の CPU パフォーマンス統計 (MHz 単位)。仮想マシンの実行中に有効です。

```
memory of <usage statistic>: integer
```

ゲスト・メモリー使用率の統計 (MB 単位)。これはアクティブ・ゲスト・メモリーとも呼ばれます。この数は、0 から仮想マシンの構成済みメモリー・サイズまでの範囲にすることができます。仮想マシンの実行中に有効です。

```
maximum cpu of <usage statistic>: hertz
```

現在の CPU 使用率の上限値。

```
[maximum memory, maximum memories] of <usage statistic>: integer
```

現在のメモリー使用量の上限。

```
[host memory, host memories] of <usage statistic>: integer
```

MV のオーバーヘッド・メモリーを含む、ホストのメモリー使用量 (バイト単位)。値の範囲は 0 から構成済みのリソース制限までであり、仮想マシンの実行中は有効です。コンシュームされたホスト・メモリーとも呼ばれます。

USB デバイス (ホスト)

```
[usb device, usb devices] : plural usb device
```

USB デバイス・インスペクターは、プロパティーのセットを返します。

```
name of <usb device>: string
```

USB デバイスの名前。

```
description of <usb device>: string
```

ユーザーに表示される USB デバイスの名前。

```
family of <usb device>: string
```

デバイス・クラス・ファミリー。可能な値については、『VirtualMachineUsbInfoFamily』を参照してください。

```
physicalpath of <usb device>: string
```

デバイスの物理パスを記述する自動接続パターン。これは、USB デバイスが接続されているホスト上の特定のポートへのパスです。

```
product of <usb device>: integer
```

USB デバイスの製品 ID。

```
speed of <usb device>: string
```

サーバーによって検出される可能性のあるデバイス速度。可能な値については、『VirtualMachineUsbInfoSpeed』を参照してください。

```
vendor of <usb device>: integer
```

USB デバイスのベンダー ID。

```
usage summary of <usb device>: usage summary
```

このデバイスを現在使用している仮想マシンに関するサマリー情報 (存在する場合)。

```
vm name of <usage summary>: string
```

仮想マシンの名前。

```
vm path of <usage summary>: string
```

仮想マシンの構成ファイルへのパス名。

仮想化インスペクター

```
datacenters: plural datacenter
```

仮想化インスペクターは、データセンター・プロパティーのセットを返します。

データセンターの仮想化インスペクター

```
name of <datacenter>: string
```

データセンターの名前。

```
datastores of <datacenter>: plural datastore
```

このデータセンターで使用可能なデータストア・オブジェクトへの参照の集合。

```
name of <datastore>: string
```

データストアの名前。

```
free space of <datastore>: string
```

このデータストアの空きスペース (バイト単位)。サーバーは定期的にこの値を更新します。これは、「更新」操作を使用して明示的に更新できます。

```
max file size of <datastore>: string
```

このファイル・システム・ボリュームに常駐できるファイルの最大サイズ。

```
timestamp of <datastore>: string
```

DatastoreInfo および DatastoreSummary の空きスペース値とキャパシティー値が更新された時刻。

```
url of <datastore>: string
```

データストアの固有ロケーター。

```
iso images of <datastore>: plural iso image
```

データストアの ISO イメージ。

```
path of <iso image>: string
```

検索結果内のフォルダー・パスに対する相対パス。

ハード・ドライバーの仮想化インスペクター

```
disk of <guest vm>: disk
```

ディスク。

```
hard driver counter of <guest vm>: string
```

ハード・ドライバーの数。

```
hard drivers of <guest vm>: plural hard driver
```

ハード・ドライバー。

```
hard drivers: plural hard driver
```

配列。

```
unitnumber of <hard driver>: string
```

コントローラー上のこのデバイスのユニット番号。コントローラー・プロパティーが null の場合 (例えば、デバイスが特定のコントローラー・オブジェクトに接続されていない場合など) は、このプロパティーは null です。

```
datastore name of <hard driver>: string
```

データストアの名前。

```
controllerkey of <hard driver>: string
```

このデバイスのコントローラー・オブジェクトのオブジェクト・キー。このプロパティーには、コントローラー・デバイス・オブジェクトのキー・プロパティー値が含まれます。

```
label of <hard driver>: string
```

ディスクのラベル。

```
capacity of <hard driver>: string
```

ディスクの合計容量 (バイト単位)。これは、仮想マシン構成の一部です。

```
key of <hard driver>: string
```

このデバイスを同じ仮想マシン内の他のデバイスと区別する固有キー。キーは不変ですが、再利用される可能性があります。つまり、デバイスが特定の仮想マシンに関連付けられている限り、キーは変更されません。ただし、デバイスが削除されると、別のデバイス

が追加されたときにそのキーが使用されることがあります。このプロパティーは読み取り専用ではありませんが、クライアントはその値を制御できません。永続的なデバイス・キーは、常にサーバーによって割り当ておよび管理されるため、すべてのデバイスに負ではないキー値が設定されます。新しいデバイスを追加する場合、仮想マシンを構成する際にコントローラーをデバイスに関連付けるために、クライアントが一時的にキーを割り当てる必要がある場合があります。ただし、サーバーがクライアントにデバイス・キーの再割り当てを許可せず、サーバーは設定時に渡された値とは異なる値を割り当てる場合があります。クライアントは、追加する新規デバイスの一時キー値として既存のデバイス・キーが再利用されないようにする必要があります (例えば、固有の負の整数を一時キーとして使用するなど)。デバイスを編集または削除する場合、クライアントはサーバー提供のキーを使用して既存のデバイスを参照する必要があります。

前回更新されたディスクの仮想化インスペクター

```
last time refreshed: string
```

最後に更新された時刻は、単にディスク・ディスカバリーのレポート実動時間を指します。レイアウトは次のとおりです。`<year> <month> <day> <hour>:<min>:<sec>`。

```
last time refreshed of <guest vm>: string
```

最後に更新された時刻は、単にディスク・ディスカバリーのレポート実動時間を指します。レイアウトは次のとおりです。`<year> <month> <day> <hour>:<min>:<sec>`。

VM 名の仮想化インスペクター

```
name: string
```

親に対する固有の、ゲストまたはホストの名前。エスケープ文字を含む可能性があります。

クラウド・プラグイン・コマンド

クラウド・プラグインは、幅広いアクションをサポートする機能を備えています。

クラウド・プラグインのコマンドは、従来のアクション・スクリプト・コマンドと同様に機能します。主な違いは、ターゲットが一部のクラウド・プロバイダー・インスタンスの表現であるということです。

単一のスクリプトでは、関連する表現に対して順番に実行される、追加のコマンドを組み込むことができます。

インスタンスの適用条件または関連度は、対応するクラウド・プラグインが報告するプロパティーに基づきます。同様に、クラウド・プラグインがコマンドを実行できるかどうかは、プラグインがサポートするコマンドに依存します。

クラウド・プラグインが適切なインスペクターをサポートしていることを確認してください。詳しくは、クラウド・プラグイン・インスペクター (（ページ） 223) を参照してください。

さらに、このセクションの子ページで、サポートされているコマンドを確認してください。

アクション・スクリプトの概要については、次を参照してください。https://developer.bigfix.com/action-script/

アクション・スクリプト・ガイドについては、次を参照してください。https://developer.bigfix.com/action-script/guide/

VMware プラグイン・コマンド

ホスト・コマンド

次の表は、ホスト・コマンドを説明しています。

Name (名前)	構文	Description
ホストのシャットダウン	shutdown host [force=<force>]	このコマンドは、ホストをシャットダウンします。

ホストの再起動	reboot host [force=<force>]	このコマンドはホストをリブートします。
保守モードの開始	enter maintenance mode [action=<action> timeout=<timeout> evacuate=<evacuate>]	このコマンドは、ホストを保守モードにします。このタスクの実行中、およびホストが保守モードの場合、仮想マシンの電源をオンにすることはできず、ホストに対してプロビジョニング操作を実行することもできません。
保守モードの終了	exit maintenance mode [timeout=<timeout>]	このコマンドは、保守専用ホスト構成操作を同時に実行している場合を除き、ホストを保守モードから解放します。

必要な VMware 特権:

- *Host.Config.Maintenance*

ゲスト電源コマンド

次の表では、ゲスト電源コマンドについて説明します。

Name (名前)	**構文**	**Description**
電源 ON	電源 ON	このコマンドは、対象のゲスト仮想マシンの電源をオンにします。
電源オフ (ハード)	電源オフ (ハード)	このコマンドは、対象のゲスト仮想マシンの電源をオフにします。
電源オフ (ソフト)	電源オフ (ソフト)	このコマンドは、対象のゲスト仮想マシンをシャットダウンします。
リセット	リセット	このコマンドは、対象のゲスト仮想マシンをリセットします (最初に電源をオフにしてから電源をオンにします)。
再開	再開	このコマンドは、対象のゲスト仮想マシンをリブートします。

中断 (ソフト)	中断 (ソフト)	このコマンドは、対象のゲスト仮想マシン・オペレーティング・システムに対して、中断操作の準備を要求します。
中断 (ハード)	中断 (ハード)	このコマンドは、多少のゲスト仮想マシンでの実行を中断します。

必要な VMware 特権:

- *VirtualMachine.Interact.PowerOn*
- *VirtualMachine.Interact.PowerOff* ソフトとハードの電源オフの両方
- *VirtualMachine.Interact.Suspend* ソフトとハードの一時停止の両方
- *VirtualMachine.Interact.Reset* リセットと再始動の両方

ゲスト・スナップショット・コマンド

次の表では、ゲスト・スナップショット・コマンドについて説明します。

Name (名前)	構文	Description
スナップショットの作成	create snapshot <snapshot-name>	このコマンドは、指定された名前のスナップショットを作成します。
スナップショットの復元	スナップショットの復元	このコマンドは、仮想マシンを現在のスナップショットに戻します。スナップショットが存在しない場合、仮想マシンの状態は変更されません。
スナップショットに移動	go to snapshot <snapshot-name>	このコマンドは、仮想マシンの実行状態を、指定された名前のスナップショットの状態に変更します。
スナップショットの名前変更	rename snapshot <snapshot-name>|+|<new-snapshot-name>	このコマンドは、指定された名前で shapshot の名前を変更します。

スナップショットの削除	remove snapshot <snapshot-name> [<children>]	このコマンドは、指定された名前のスナップショットを削除します。オプションのブール値が true に設定されている場合、すべてのスナップショットの子が削除されます。
すべてのスナップショットの削除	すべてのスナップショットの削除	このコマンドは、仮想マシンのすべてのスナップショットを削除します。

必要な VMware 特権:

- *VirtualMachine.State.CreateSnapshot*
- *VirtualMachine.State.RemoveSnapshot* スナップショットの削除と全てのスナップショットの削除の両方
- *VirtualMachine.State.RenameSnapshot*
- *VirtualMachine.State.RevertToSnapshot* スナップショットの復帰とスナップショットへの移動の両方

ゲスト・ツール・コマンド

以下の表では、ゲスト・ツール・コマンドについて説明します。

Name (名前)	構文	Description
マウント・ツール	マウント・ツール	このコマンドは、VMware Tools インストーラーを仮想マシンにマウントします。
アップグレード・ツール	アップグレード・ツール	このコマンドは、仮想マシン用の VMware ツールをアップグレードします。

必要な VMware 特権:

- *VirtualMachine.Interact.ToolsInstall* マウントツールとアップグレードツールの両方

ゲスト VLAN コマンド

次の表では、ゲスト VLAN コマンドについて説明します。

Name (名前)	構文	Description
VLAN の変更	change vlan <vlan-name>|+|<adapter-number>	このコマンドは、仮想マシンの VLAN を切り替えます。

必要な VMware 特権:

- *VirtualMachine.Config.EditDevice*

クラウド分析のアクティブ化

クラウド・プラグインがインストールされている場合、BigFix 10 の新しいクラウド関連機能を利用するには、次の BES サポート分析をアクティブにする必要があります。

「アクティブ化」:

- BES コンポーネントのバージョン (BES Component Versions)
- クラウド相関データ
- *Plugin_name* プラグイン設定
- *Plugin_name* リソース

- 必要な分析の一部は、クラウド・プラグイン・ダッシュボード (（ページ） *282)*からもアクティブにすることができます。
- WebUI からクラウド・プラグインをインストール (（ページ）)すると、必要な分析の一部が自動的にアクティブになります。

クラウド分析データ

分析はすべてのクラウド仮想インスタンスに送信され、特定のクラウド・プラグインで検出可能です。このプラグインは分析を評価し、そのステータスを報告します。これにより、クラウド仮想インスタンスの指定プロパティーを BigFix コンソールから監視できるようになります。

分析により提供されるデータ:クラウド・プロバイダーのプラグイン設定

これらの分析から、以下のプラグイン設定の情報が報告されます。

- 設定: すべての設定のリストを表示する複数のプロパティー。
- バージョン: プラグインのバージョン。
- リフレッシュ間隔: 分単位で指定された値が、**プラグイン更新リフレッシュ間隔**というタスクに設定されます。

分析により提供されるデータ: Amazon Web Services リソース

この分析からは、各プロバイダーに固有の情報が報告されます。特に、Amazon Web Services リソース分析では、以下のようなデータが報告されます。

- ステート AWS: 仮想マシンの状態。
- 起動時刻: インスタンスが起動した時刻。
- タイプ AWS: 特定のインスタンス・タイプ
- イメージ ID: イメージ ID 番号。
- 所有者 ID: 所有者 ID 番号。
- タグ AWS: 使用されている定義済みのタグ。
- リージョン AWS: 仮想マシンが設置されているリージョン。
- テナンシー: 使用されているテナンシー・モデル。専用または共有のいずれかを使用します。デフォルトのテナンシー・モデルは共有テナンシーです。
- プラットフォーム: 値は、Windows インスタンスの場合は Windows です。それ以外の場合は空白です (これは API AWS の制限です)。
- プライベート/パブリック IP AWS: 仮想マシンに割り当てられたプライベート IP アドレスまたはパブリック IP アドレス。
- プライベート/パブリック DSN 名: 仮想マシンに割り当てられたプライベート DNS またはパブリック DNS。
- セキュリティー・グループ: セキュリティー・グループは、インスタンスの仮想ファイアウォールとして機能し、インバウンド・トラフィックとアウトバウンド・トラフィックをコントロールします。
- アベイラビリティー・ゾーン: リージョン内の独立した場所で、他のアベイラビリティ・ゾーンで障害が発生した場合に影響を受けないよう設計されています。
- キー名: 仮想マシンに割り当てられたキー名。

- VPC ID: 使用されている特定の仮想プライベート・クラウド (VPC) の ID 番号。
- イメージ名: 使用されているイメージの固有名称。
- インスタンス ID AWS: インスタンスの ID 番号。
- アカウント・エイリアス AWS: 使用されているアカウントのエイリアス。
- 相関 ID AWS: 相関 ID 番号。
- IAM ロール AWS: インスタンスの発見に使用される IAM ロール。

クラウド・プラグインは追加情報も含めて報告できる可能性があります。

分析により提供されるデータ: Microsoft Azure リソース

この分析からは、各プロバイダーに固有の情報が報告されます。特に、Microsoft Azure リソース分析では、以下のようなデータが報告されます。

- ステート Azure: 仮想マシンの状態。
- プロビジョニング・ステート: 仮想マシンのプロビジョニングの状態。
- プロビジョニング時刻: 仮想マシンのプロビジョニング時刻。
- タイプ Azure: 仮想マシンのタイプ。
- イメージの発行者: イメージを作成した発行者または組織。
- イメージ・オファー: 発行者が作成した関連イメージのグループ名。
- タグ Azure: 使用されている定義済みのタグ。
- リージョン Azure: 仮想マシンが設置されているリージョン。
- プライベート/パブリック IP Azure: 仮想マシンに割り当てられたプライベート IP アドレスまたはパブリック IP アドレス。
- リソース・グループ: 仮想マシンが属するリソースのグループ。
- インスタンス ID Azure: インスタンスの ID 番号。
- アカウント・エイリアス Azure: 使用されているアカウントのエイリアス。
- 相関 ID Azure: 相関 ID 番号。

クラウド・プラグインは追加情報も含めて報告できる可能性があります。

分析により提供されるデータ: VMware リソース

この分析からは、各プロバイダーに固有の情報が報告されます。特に、VMware リソース分析では、以下のようなデータが報告されます。

- カスタム属性: 仮想マシン用に定義されたカスタム属性。
- オペレーティング・システム: 使用されているオペレーティング・システム。
- 電源状態 VMware: 仮想マシンの電源状態。
- ステータス VMware: 仮想マシンの状態。
- BIOS UUID: BIOS UUID (汎用固有 ID) 番号。
- ホスト: 仮想マシンのホスト。
- VM UUID: 仮想マシンの BIOS UUID (汎用固有 ID) 番号。
- アカウント・エイリアス VMware: 使用されているアカウントのエイリアス。

クラウド・プラグインは追加情報も含めて報告できる可能性があります。

分析により提供されるデータ: Google Cloud Platform リソース

この分析からは、各プロバイダーに固有の情報が報告されます。特に、Google Cloud Platform リソース分析では、以下のようなデータが報告されます。

- インスタンス ID GCP: インスタンスの ID 番号。
- アカウント・ラベル GCP: 資格情報ラベル。
- タグ GCP: 使用されている定義済みのタグ。
- 状況メッセージ GCP: 状況メッセージ。
- 状況 GCP: 状況。
- タイプ GCP: マシン・タイプ。
- 作成時刻: 作成時のタイム・スタンプ。
- CPU プラットフォーム: CPU プラットフォーム。
- ゾーン GCP: ゾーン。
- ネットワーク・インターフェース名: ネットワークの IP インターフェースの名前。
- ネットワーク・サブネット・アドレス: ネットワークの IP インターフェースのサブネット・アドレス。
- プライベート IP GCP: ネットワークの IP インターフェースのアドレス。
- パブリック IP GCP: ネットワークの IP インターフェースの外部アドレス。
- IP 転送: IP アドレスが転送可能な場合。
- 相関 ID GCP: 相関 ID。
- プロジェクト ID GCP: GCP インスタンスが属するプロジェクトのプロジェクト ID。

クラウド・インスペクター

クラウド・インスペクターは、BigFix エージェントから以下のような情報を収集します。

- 仮想マシンであるかどうか。
- 仮想マシンである場合:
  - クラウド・プロバイダーの名前。
  - インスタンスのリージョンおよびアベイラビリティー・ゾーン。
  - インスタンスの固有 ID (インスタンス ID または VM ID)。
  - プライベート IP。

さまざまなクラウド・プロバイダーの構成プロパティーの詳細については、各製品資料を参照してください。

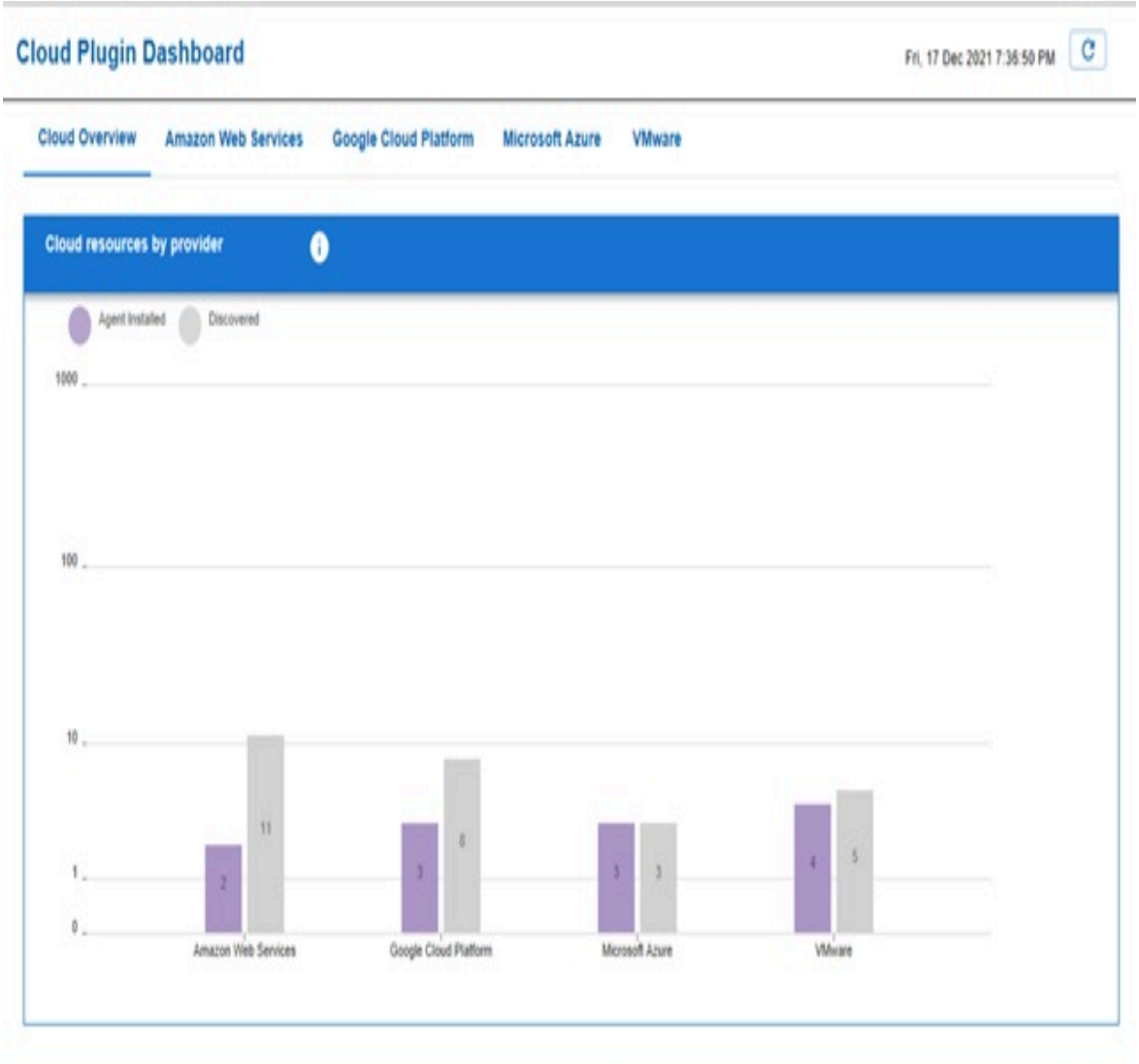
クラウド・プラグイン・ダッシュボードの操作

クラウド・プラグイン・ダッシュボードで表示される情報。

BES サポートの**クラウド・プロバイダー**分析をアクティブにして、クラウド・プラグイン・ダッシュボードの使用を開始します。

クラウド・プラグイン・ダッシュボードにアクセスするには、BigFix コンソールで次の手順を実行します。

1. コンソールの左端のパネルの**「すべてのコンテンツ」**ドメインをクリックします。
2. **「ダッシュボード」**エントリーを展開します。
3. **「BES サポート」**エントリーを展開します。
4. **「クラウド・プラグイン・ダッシュボード」**をクリックします。ダッシュボードが表示されます。



「クラウドの概要」タブに、検出されたすべてのクラウド・リソース (BigFix エージェントの管理対象と管理対象外の両方) が表示されます。紫色のバーは、BigFix エージェントがインストールされているクラウド・リソースを示します。グレーのバーは、BigFix エージェントがまだインストールされていないクラウド・リソースを示します。

紫色のバーまたはグレーのバーをクリックすると、そのバーで表されるすべてのクラウド・リソースの詳細が表形式で表示されます。次に、表にリストされている項目を任意の列名でフィルタリングできます。「名前」列には、対応するコンピューターのプロパティーへのハイパーリンクが含まれています。

- **「Amazon Web Services」**タブには、AWS プラグインによって検出されたクラウド・リソースに関するキー情報が含まれています。
- **「Microsoft Azure」**タブには、Azure クラウド・プラグインによって検出されたクラウド・リソースに関するキー情報が含まれています。
- **「VMware」**タブには、VMware プラグインによって検出されたクラウド・リソースに関するキー情報が含まれています。
- **「Google Cloud Platform」**タブには、Google Cloud Platform プラグインによって検出されたクラウド・リソースに関するキー情報が含まれています。

検出されたリソースへの BigFix エージェントのインストール

クラウド・プラグインによって取得されたリソースには、必ずしも BigFix エージェントがインストールされている必要はありません。クラウドで検出されたすべてのリソースを一覧表示する**クラウド・プラグイン・ダッシュボード**を使用することで、まだ管理対象外のリソースに BigFix エージェントをインストールできます。

1. ダッシュボードの**「クラウドの概要」**タブから、クラウド・プロバイダーのグレーのバーをクリックします。そのクラウド・プロバイダーのすべての管理対象外リソースを含むテーブルが表示されます。
2. テーブルで、行をクリックして、BigFix エージェントをインストールするリソースを選択します。Ctrl または Shift ボタンによる複数選択がサポートされています。すべてのリソースを一度に選択する場合は、**「非管理リソースの選択」**ボタンをクリックします。
3. **「エージェントのデプロイ」**ボタンをクリックします。

ダッシュボードにクライアント適用ツール・ウィザードが表示されます。インストールする対象のリストは既に入力されています。インストールは、クライアント適用ツール・ウィザードの標準のフローに従います。クライアント適用ツール・ウィザードの使用方法の詳細については、クライアント適用ツールの使用 (（ページ）　　)を参照してください。

注: クライアント・プラグイン・ダッシュボードからクライアント適用ツール・ウィザードを起動すると、次のようになります。

- ウィザードの**「対象の資格情報の設定」**ページで、**「対象の追加」**ボタンが無効になります。
- ウィザードの**「詳細設定」**ページの**「OS ファミリー」**セクションでは、クラウド・インフラストラクチャーでサポートされていないプラットフォームはグレー表示されます。
- ウィザードの**「詳細設定」**ページの**「クライアント・バージョン」**プルダウン・メニューで、10 以降のみ選択できます。

BigFix エージェントがインストールされると、選択されたリソースには、プロキシー・リソースとネイティブ・リソースの 2 つの表現があります。BigFix サーバーはこれらを 1 つの相関コンピューターで関連付けます。

BigFix クラウド・リソースへのエージェントのインストール

パッチ 2 以降では、対応する BigFix クラウド・プラグインを通じてクラウド・プロバイダー・サービスを使用して、検出されたクラウド・リソース (AWS および Azure) に BigFix エージェントをインストールできます。

BigFix 検出されたクラウド・リソースにエージェントをデプロイできるようするために、プラットフォームにより、BigFix Enterprise Suite (BES) サポートで 2 つの新しいタスク (クラウド・プロバイダーごとに固有) が提供されます。

タスクは、BigFix コンソールと WebUI の両方から使用できます。クラウド・プロバイダーに応じて、次のいずれかのタスクを使用します。

- `4774 Install BigFix Client through Amazon Web Services`
- `4775 Install BigFix Client through Microsoft Azure`

タスクの実行内容

タスクにより、ターゲット・クラウド・リソースでローカルに完了する次のアクションが実行されます。

- software.bigfix.com からインストール・スクリプトを取得します。
- software.bigfix.com から、プラグイン・ポータルのバージョンに基づいてクライアント・インストーラーを取得します。
- 未認証リレーから、タスクへの入力のために提供されるデプロイメント・マストヘッドを取得します。
- クライアントをインストールし、未認証リレーに登録します。
- クライアントの登録が完了するまで待機します。

要件

この新しいインストール機能を使用するには、検出されたクラウド・リソースが、クラウド・プロバイダー固有の多数の構成要件を満たしている必要があります。これらの要件のサブセットは、タスクに含まれる関連性によって自動的にチェックされます。クラウド・リソースが正しく構成されていることを確認する必要があります。

検出されたクラウド・リソースにエージェントをインストールする場合の主な要件は次のとおりです。

- プロバイダー固有のクラウド・エージェントを、クラウド・リソースにインストールする必要があります。詳細については、「AWS (（ページ） 287)」および「Azure (（ページ） 288)」を参照してください。
- クラウド・リソースはアクティブで、実行中である必要があります。
- 未認証リレーの名前 (タスクの実行中に入力として指定)。詳しくは、入力としてのリレー名 (（ページ） 289)を参照してください。
- クラウド・リソースで実行されているオペレーティング・システムを、BigFix クライアントがサポートしている必要があります。詳しくは、サポートされるオペレーティング・システム (（ページ） 289)を参照してください。
- プラグイン・ポータルは、最低でもパッチ 2 レベルである必要があります。

重要な考慮事項

この機能は、次のリレーと設定をサポートしていません。

- 認証リレー
- インストール時に設定されるカスタム・クライアント設定

AWS の要件

VM の要件

- アクティブで実行中の AWS Systems Manager (SSM) エージェントが VM に存在する必要があります。
- IAM インスタンス・プロファイルが存在し、IAM 役割を介してそのプロファイルに `AmazonSSMManagedInstanceCore` IAM ポリシーを関連付ける必要があります。IAM インスタンス・プロファイルは VM にアタッチする必要があります。この要件により、AWS Systems Manager は、VM でコマンドを安全に実行できます。

IAM ID の要件

AWS Cloud Plugin で設定する場合、IAM ID (ユーザーとロール) が常に満たす必要がある基本的な要件は次のとおりです。

- MFA を無効にする必要があります
- プログラムによるアクセス・タイプが必要です

AWS Cloud Plugin のクレデンシャルとして使用するアクセス・キー ID と シークレット・アクセス・キーのペアに関連付けられた IAM ユーザーで BigFix エージェントをインストールする場合、必要な権限は以下のとおりです。

- 少なくとも次の ec2 権限が必要です。ec2:Describe* アクションを * リソースで許可する必要があります。
- 少なくとも次の ssm 権限が必要です。ssm:DescribeInstanceInformation、ssm:SendCommand、および ssm:GetCommandInvocation アクションを * リソースで許可する必要があります。

AWS Cloud Plugin で構成された IAM ユーザーが引き受けることができる IAM ロールで BigFix エージェントをインストールする場合、必要な権限は次のとおりです。

ユーザーの場合:

- 対象のロールで sts:AssumeRole を実行できる必要があります。

ロールの場合:

- 少なくとも次の ec2 権限が必要です。ec2:Describe* アクションを * リソースで許可する必要があります。
- 少なくとも次の ssm 権限が必要です。ssm:DescribeInstanceInformation、ssm:SendCommand、および ssm:GetCommandInvocation アクションを * リソースで許可する必要があります。
- ロールを引き受ける IAM ユーザーは、ロールの信頼された ID である必要があります。

注: AWS ロールが挿入されると、AWS プラグインは、取得元の資格情報ではなく、検出時に AWS ロールを使用します。クラウド環境で検出するすべての AWS デバイスがこれらの役割に含まれるようにする必要があります。そうしない場合、一部のマシンが検出されない可能性があります。

前提条件の完全なリストについては、「Systems Manager の前提条件」を参照してください。2017 年 9 月以降の日付の Amazon Linux ベース API には、デフォルトで Systems Manager が含まれます。他の Amazon Machine Image (AMI) またはカスタム AMI では、Systems Manager (SSM) エージェントがまだ存在しない場合は、手動でインストールする必要があります。オペレーティング・システム・サポートの詳細については、「Systems Manager でサポートされるオペレーティング・システム」を参照してください。

Azure の要件

- VM エージェントが VM に存在する必要があります。
  Azure には、プラットフォームに基づいて、コマンドを実行する 2 つのエージェントが用意されています。
  - Azure Linux エージェント
  - Azure 仮想マシン・エージェント
- オペレーター (ディスカバリー資格情報) には、少なくともアクション権限が必要です。

エージェントが存在しない場合、OS の要件が満たされているのであれば、前の参照で示したようにエージェントをインストールできます。Azure VM でコマンドを実行するには、オペレーターが少なくともアクション権限を持っている必要があります。`Microsoft.Compute/virtualMachines/runCommand/action`.コントリビューター役割またはそれ以上の組み込みの役割には、この権限はデフォルトで含まれます。ただし、特定のカスタム役割を定義することもできます。

入力としてのリレー名

タスクにより、未認証リレーの名前を入力するように求められます。次のいずれかの形式で入力します。

- リレーのホスト名。例えば、「*myhostname*」と入力します。
- リレーの完全修飾ドメイン名 (FQDN)。例えば、「*myhostname.mydomain.com*」と入力します。
- リレーの IP アドレス。例えば、「*10.10.10.10*」と入力します。

サポートされるオペレーティング・システム

この機能を使用するには、次のプロバイダー固有のエージェントのいずれかがインストールされ、アクティブになっている必要があります。

- AWS の SSM エージェント
- Azure の VM エージェント

したがって、プロバイダー・エージェントでサポートされている一部のオペレーティング・システムが、BigFix でサポートされない場合があります。このオペレーティング・システムのサブセットは、将来変更される可能性があります。

AWS VM でサポートされる OS

AWS SSM エージェントが現在サポートされているオペレーティング・システムのタイプについては、「AWS SSM でサポートされる OS」を参照してください。

現在サポートされている BigFix エージェントのオペレーティング・システムについて詳しくは、「Detailed system requirements (（ページ）　　)」を参照してください。

Azure VM でサポートされる OS

Azure VM エージェントが現在サポートされているオペレーティング・システムのタイプについては、次の文書を参照してください。

- Windows VM エージェントのサポート OS
- Linux VM エージェントのサポート OS
- 拡張エージェント・バージョンのサポート

現在サポートされている BigFix エージェントのオペレーティング・システムについて詳しくは、「Detailed system requirements (（ページ）　　)」を参照してください。

トラブルシューティング

エージェントのインストール失敗をトラブルシューティングする方法は複数あります。

- 以下に示すカスタム終了コードのリストを確認できます。
- ActionResultsStore.db という名前の SQLite データベースであるトラブルシューティング機能を使用できます。このデータベースは、クラウド・プラグインごとに使用できます。
- アクション・ログを確認できます。ログは手動で確認できます。デフォルトの場所については後述します。

インストール・スクリプトの終了コード

表 4. ネイティブ・エージェントのインストール終了コード

終了コード	原因	解決方法
209	クライアント・ログ・ファイルが見つかりません。この原因として考えられるのは、インストール対象のクライアントを指定の時間内にインストールできなかったことです。	クライアントが登録されるまで数分待機してください。問題が解決しない場合、クライアントを手動でアンインストールし、再度インストールしてください。
210	クライアントの登録が失敗しました。ターゲットがリレーに接続できないため、またはマス	ターゲット・インスタンスがリレーの FQDN、IP、またはホスト名を解決するかどうか、およびリレーのマ

表 4. ネイティブ・エージェントのインストール終了コード (続く)

終了コード	原因	解決方法
	トヘッドが破損しているため、これが発生した可能性があります。	ストヘッドで指定されたホスト名を解決するかどうかを確認してください。
211	BESClient サービスが見つかりません。	/tmp/BigFix または %TEMP%/BigFix で、ターゲットのクライアント・インストール・ログを確認してください。クライアントをアンインストールし、再度インストールしてください。
212	クライアントは既にインストールされています。ターゲットにクライアントのインストールが存在します。	
213	入力パラメーターが欠落しています。これは、手動エラーが原因である可能性があります。	プロンプトが表示されたら入力パラメーターの値を入力し、再試行してください。
215	チェックサムによって誤った値が返されました。ダウンロードしたインストール・スクリプトが破損しています。	ターゲットにネットワーク接続があるかどうか、および BES サポート・サイトに到達できるかどうかを確認してください。
216	Linux ターゲットで curl および wget ユーティリティーが見つかりません。	Linux ターゲットに wget または curl ユーティリティーをインストールしてください。
217	取得した OS データで、スクリプトを実行できません。この原因として考えられるのは、ターゲットにインストールされてい	クライアントを手動でインストールするか、CDT を使用してインストールしてください。

表 4. ネイティブ・エージェントのインストール終了コード (続く)

終了コード	原因	解決方法
	る OS がサポートされていないこと、またはターゲットから返される OS に関する情報が不十分であることです。	
218	十分な OS データが取得されていません。この原因として考えられるのは、ターゲットにインストールされている OS がサポートされていないこと、またはターゲットから返される OS に関する情報が不十分であることです。	クライアントを手動でインストールするか、CDT を使用してインストールしてください。
219	Linux ターゲットで shasum および sha256 を使用できません。	Linux ターゲットに shasum または sha256 ユーティリティーをインストールしてください。
220	インストーラー・ファイルが見つかりません。	ターゲットが software.bigfix.com に到達できるかどうかを確認してください。
221	マストヘッド・ファイルが見つかりません。	ターゲットがリレーに到達できるかどうかを確認してください。リレーが認証を行う場合は、クライアントを手動でインストールするか、CDT を使用してインストールしてください。

表 5. その他の一般的なインストール終了コード

終了コード	原因	解決方法
6	Linux ターゲットで、curl エラー「Could not resolve host」が発生しています。	ターゲットが software.bigfix.com、リレー、および BES サポート・サイトに到達できるかどうかを確認してください。
22	Linux ターゲットで、curl エラー「Request has encontered an error greater than 400」が発生しています。	ターゲットが software.bigfix.com、リレー、および BES サポート・サイトに到達できるかどうかを確認してください。

ActionResultsStore データベース

ActionResultsStore データベースは、各プラグイン・フォルダー内に保存されます。このデータベースには、失敗状態またはエラー状態のアクションの結果が格納されます。

このデータベースは、次の列がある ACTION_RESULTS という名前のテーブルで構成されます。

- ActionID
- DeviceID
- InstanceID
- Results
- Date

1 次キーは (ActionID, DeviceID) になります。
データベースの Results 列には JSON 文字列が格納されます。Results の下の各 JSON には、次のキーがあります。

- **status**: アクションの状況を表します。
- **exitCode**: 使用可能な終了コードがある場合。
- **output**: プロバイダーへの要求の出力。

テーブルの例を以下に示します。

```
{
    "status":"Error",
    "exitCode":1,
    "output":{
       "CloudWatchOutputConfig":{
          "CloudWatchLogGroupName":"",
          "CloudWatchOutputEnabled":false
       },
       "CommandId":"461eee16-e70f-4cf9-b64c-
  e2902e355f49",
       "Comment":"BES Plugin Shell Cmd",
       "DocumentName":"AWS-RunShellScript",
       "DocumentVersion":"",
       "ExecutionElapsedTime":"PT0.011S",
       "ExecutionEndDateTime":"2020-11
  -20T08:08:40.693Z",
       "ExecutionStartDateTime":"2020-11-20T08:08:40.693Z",
       "InstanceId":"i-
   04f8e15e41aaf1544",
       "PluginName":"aws:runShellScript",
       "ResponseCode":1,
       "StandardErrorContent":"mkdir: missing operand\nTry 'mkdir --help'
 for more
       information.\nfailed to run commands: exit status 1",
       "StandardErrorUrl":"",
       "StandardOutputContent":"",
       "StandardOutputUrl":"",
       "Status":"Failed",
       "StatusDetails":"Failed"
    }
 }
```

データベースには、BigFix SQLite インスペクターを使用してリモートからアクセスできます。この例として、Fixlet デバッガーを使用し、照会チャネル機能と BES ポータル・エンドポイントの ID を設定し、次の例のような照会を評価できます。

```
Q: rows of statement <sql statement> of sqlite database of file
 "ActionResultsStore.db"
of folder <plugin folder>
```

1 日 1 回、ActionResultsStore データベースは自動的にクリーンアップされます。どのクリーニングでも、指定した日数より古いエントリーがすべて削除されます。この日数は、各プラグインで使用できる `<plugin_name>_ActionResultsStore_CleanupDays` という名前の設定によって構成できます。この設定の値は、0 (ゼロ) より大きくする必要があり、日数で指定します。

インストーラー、マストヘッド、およびログ・ファイルの場所

インストーラー、マストヘッド、およびインストール・タスクの結果の一部であるログ・ファイルは、「BigFix」という名前のフォルダーに格納されます。このフォルダーは、各ターゲット・インスタンス内の次のディレクトリーで見つけることができます。

Windows の場合

%LOCALAPPDATA%\BigFix

Linux の場合

/tmp/BigFix

トラブルシューティング

このセクションは、一般的な問題または制限のトラブルシューティングに役立ちます。

プラグイン・ポータル・ログの「InspectorDataJSONCallback の予期しないエラー」メッセージ

説明: クラウド・プラグインをインストールした後、検出されたリソースが BigFix コンソールに表示されず、プラグイン・ポータルのログに次のエラー・メッセージが含まれます。

```
Mon, 30 Mar 2020 18:31:56 +0200 - 28965245 - Unexpected error in
 InspectorDataJSONCallback
No suitable servers found: `serverSelectionTimeoutMS` expired: [
connection refused calling ismaster
on 'localhost:27017'] .
```

理由: このエラー・メッセージは、MongoDB が停止していることが原因である可能性があります。

解決方法: MongoDB を開始します。Windows では、これは Windows サービス・ツールから、Linux では `systemctl start mongod` コマンドを発行して実行できます。

クラウド・プラグインのインストール後に BESPluginPortal が実行されていません

説明: クラウド・プラグインをインストールした後に、BESPluginPortal プロセスが実行されていません。

理由: この動作は予期しないものであり、調査を必要とする予測できない理由が原因である可能性があります。

解決方法:

- Windows の場合:
  - Windows サービス・ツールから BESPluginPortal プロセスを再起動してみてください。
  - それでも BESPluginPortal が実行されない場合は、Windows イベント・ビューアーで BESPluginPortal プロセスに関連するエラーがないかどうか確認します。
- Windows 以外:
  - `/etc/init.d/bespluginportal start` コマンドを発行して BESPluginPortal プロセスを再起動してみてください。
  - それでも BESPluginPortal が実行されない場合は、`/var/log/messages` で BESPluginPortal プロセスに関連するエラーがないかどうか確認します。

認証に失敗したため、AWS プラグインはリソースを検出できません (ステータス・コード: 401)

説明: AWS クラウド・プラグインはリソースの検出の実行に失敗し、次のエラー・メッセージをログに記録します。

```
2020/03/30 15:50:22 - [error] Got error calling DescribeRegions:
 AuthFailure:
AWS was not able to validate the provided access credentials
status code: 401, request id: 1879798f-b446-4ar1-acdc-w35a1ut3y0
u
```

理由: エラー・メッセージは、次の場合に表示されます。

1. 指定された `Access Key ID` / `Secret Access Key` ペアが間違っているか非アクティブである。
2. 指定された `Access Key ID` / `Secret Access Key` ペアが、IAM ロールを引き受けることによって作成された一時的な資格情報セットの一部として SessionToken に関連付けられている。
3. AWS プラグインがインストールされているコンピューターの日時が正確でない。

解決方法: ユース・ケースに応じて、ソリューションは次のように異なります。

1. 指定された `Access Key ID` / `Secret Access Key` ペアが正しいこと、およびそのステータスがアクティブであることを確認します。
2. セッション・トークンに関連付けられた一時的な資格情報はサポートされていません。資格情報を IAM ユーザーに関連付けられた `Access Key ID` / `Secret Access Key` ペアに置き換えます。
3. AWS プラグインがインストールされているコンピューターのクロック (日付、時刻、タイムゾーン) を調整します (+/- 5 分は、AWS が許容する最大偏差です)。着信要求の署名の詳細については、AWS の資料を参照してください。

相関コンピューターが自動コンピューター・グループに含まれていません

説明:相関コンピューターの ID を参照する包含条件を持つ自動コンピューター・グループを作成する場合、相関コンピューターも相関表現もグループに含まれません。

理由:相関コンピューターは、BigFix サーバーのみが認識している論理エンティティーを表すため、相関コンピューターの ID に対して BigFix エージェントが応答することはありません。

解決方法: なしこれは予想された動作です。

個別の VMware コンピューターが相互に関連付けられています

説明: VMware タイプの 2 つの異なるプロキシー・コンピューターは、同じ相関コンピューターの下で関連付けられます。

理由: 意図的に、BigFix サーバーは、「VMware リソース」分析の「BIOS UUID」プロパティーに同じ値を報告している VMware コンピューターを相互に関連付けます。この値は通常、VMware で一意であると想定されていますが、VMware コンバーターを使用した VM の変換や VM (テンプレートではない) のクローン作成など、値が重複する可能性があるケースも文書化されています。BIOS UUID の重複の詳細については、VMware 知識ベースを参照してください。

解決方法: 重複する BIOS UUID を排除するには、VMware の公式ドキュメント (例:Editing a virtual machine with a duplicate UUID.bios) を参照してください。VMware プラグインの次回の検出時に、BIOS UUID の重複が削除された後、予期しない相関関係は自動的に削除されます。

VMware プラグインが一部のリソースを検出しない

説明: 一部のデバイスは VMware プラグインによって検出されません。この例では、次の 2 つのインスタンスが破棄されます。

```
2022/01/12 12:18:31 +0100 - [info] Refresh all: Discovery return
ed 8 unique devices
```

```
2022/01/12 12:18:31 +0100 - [debug] Refresh all: Reported instan
ces: 10 - Unique instances: 8 -
Terminated instances: 0 - Null instances: 0 - Other errors: 0
```

理由: VMware プラグインは、ディスカバリー時に vm.uuid を固有キーとして使用します。このキーを持たないデバイスが検出された場合、プラグインは考慮されません。

解決方法: なしこれは予想された動作です。

クラウド・コンピューターがオフラインとして表示されることがよくある

説明:BigFix コンソールには、クラウド・プラグインによって検出されたプロキシー・コンピューターがオフラインとして表示されることがよくあります。

理由: デフォルトでは、クラウド・プラグインの検出頻度は 120 分です。この時間間隔は、BigFix コンソールの「オフラインと判断するまでの時間」設定のデフォルト値 (45 分) よりも長いです。その結果、クラウド・プラグインによって実行された最新の検出の 45 分後、次の検出が実行されるまで、プロキシー・コンピューターが BigFix コンソールにオフラインとして表示されます。

解決方法: クラウド・プラグインの検出頻度と BigFix コンソールの「オフラインと判断するまでの時間」設定でデフォルト値を使用した場合、これは予想される動作です。

第 16 章. 永続的な接続

永続的な接続を確立する機能が製品に追加されました。

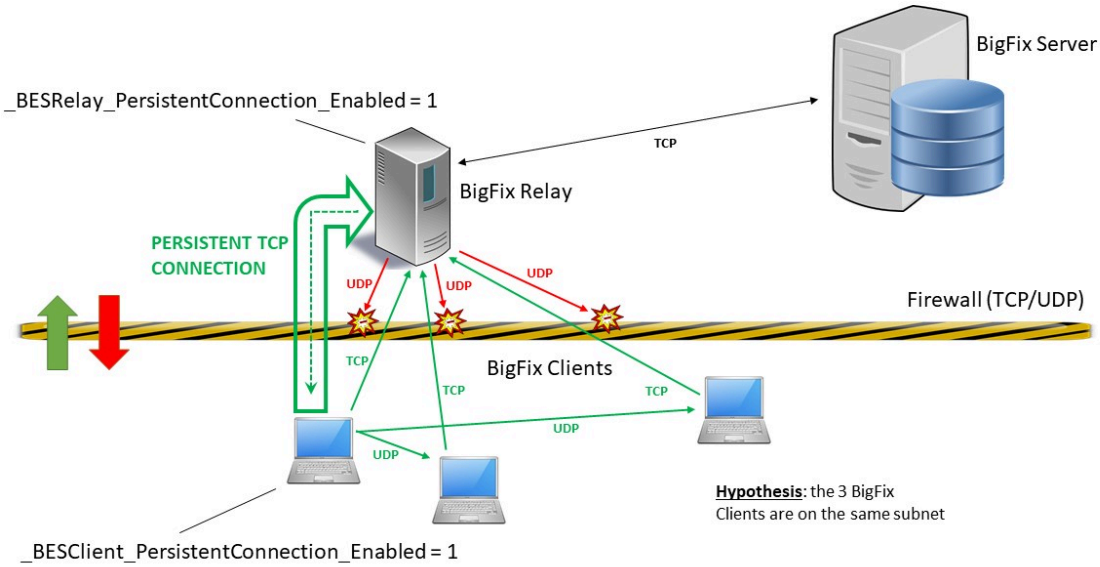
ファイアウォールまたは NAT の背後のクライアント

ファイアウォールや NAT は、BigFix Query 機能の正常な動作を妨げることがあります。これは、子クライアントに照会を送信するために親リレーが使用する UDP 通知が通常クライアントに到達できないためです。他の製品機能と異なり、BigFix Query はクライアント・ポーリングを巧みに利用してダウンストリーム通信におけるこの制限を打開することができません。

この制限は、親リレーとその 1 つ以上の子クライアント間に持続的な TCP 接続を確立することで打開できます。永続的な接続は、常にクライアントによって開始され、永続的に接続されたそのクライアント (PCC) の同じサブネット内のすべてのクライアントに UDP 通知を送信する目的でリレーによって使用されます。

概要

次の図は、クライアントとリレーの間に確立された持続的な TCP 接続と、PCC から同じサブネットの別のクライアントに送信される UDP 通知を示しています。



リレーでの永続的な接続の有効化

1. BigFix コンソールにマスター・オペレーターとしてログインします。
2. リレー・コンピューターを見つけ、右クリックします。**「コンピューター設定の編集...」**を選択します。
3. コンピューターに次の設定を追加します。

```
_BESRelay_PersistentConnection_Enabled = 1
```

4. 設定を有効にするため、リレー・プロセスを再始動します。

注: 9.5 パッチ 11 よりも前のバージョンが含まれるリレー・コンピューターにこの設定を追加した場合、その子クライアントの動作は変化しません。

注: この設定は、BigFix サーバー・コンピューターには使用できません。

クライアントでの永続的な接続の有効化

1. BigFix コンソールにマスター・オペレーターとしてログインします。
2. クライアント・コンピューターを見つけ、右クリックします。**「コンピューター設定の編集...」**を選択します。
3. コンピューターに次の設定を追加します。

```
_BESClient_PersistentConnection_Enabled = 1
```

永続的な接続の確立

有効に設定した後、通常、クライアントの次の登録時にクライアントとその親リレーの間の持続的な TCP 接続が確立されます。

次の登録が発生するときには、クライアントの登録対象であるリレーにより、そのリレーがすでに処理している永続的な接続の総数とサブネット別のそれらの区分に基づき、そのクライアントに永続的な接続を開く資格があるかどうかがチェックされます。クライアン

トに資格がある場合、リレーはそのことを適宜通知します。クライアントは、通知を受けてから 60 秒待ちます。この時間内にテスト UDP 通知がリレーから届かなければ、クライアントは永続的な接続を開くことができます。

永続的な接続の確立に失敗した場合、クライアントは 3 分後に永続的な接続のオープンをもう一度試します (合計で最大 4 回試行)。

通常、永続的な接続は、すべての前提条件が引き続き満たされている限り、クライアントが新しい登録を実行するたびにいったん閉じて再確立できます。永続的な接続の遮断は、クライアントまたはリレーが再起動操作と停止操作に対応しなければならない場合にも発生する可能性があります。

永続的な接続での通信

直接:

持続的に接続しているクライアント (PCC) にリレーが UDP 通知を送信する必要がある場合、リレーはその永続的な接続を使用し、対象クライアントに直接通知を送信します。

同じサブネットの別のクライアントによって対応されている場合:

PCC によって対応されている、サブネット内の特定のクライアントにリレーが UDP 通知を送信する必要がある場合、リレーはその通知と対象クライアント情報 (登録段階で格納されたホスト名/IP アドレス) を PCC に送信します。PCC は、その通知を読み取り、UDP を通じてその通知を対象クライアントに送信します。対象クライアントは、その通知を通常どおり処理し、通常どおり応答を直接リレーに返します。クライアントに対応できる、利用可能な PCC が同じサブネット内に複数存在する場合、リレーは利用可能な PCC すべてではなく 1 台の PCC にのみ通知を送信します。

永続的な接続の管理

永続的な接続は、いくつかの設定を構成することによって管理できます。詳細は、「永続的な TCP 接続 (（ページ）　　)」を参照してください。

第 17 章. DMZ 内のリレー

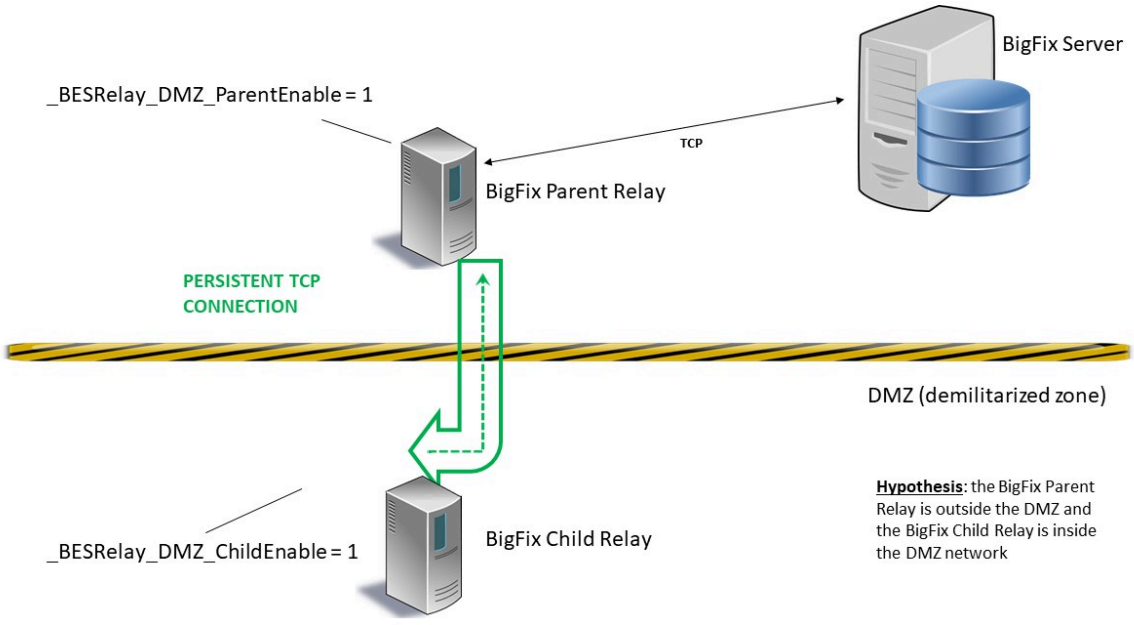
より安全なゾーンにある親リレーと DMZ ネットワーク内のその子リレーの間に永続的な TCP 接続を確立する機能が製品に追加されました。この機能を使用することで、非武装地帯 (DMZ ネットワーク) 内のシステムを管理できます。

DMZ 内のリレーがそのイントラネット・ネットワーク内の親リレーの制御下にある環境では、イントラネットと DMZ 間の通信はすべて、どのようなアップストリーム通信も許可しないファイアウォールを通じて行われると見なすことができます。この場合、DMZ 内の子リレーによる親リレーとの通信開始の試みはすべて失敗します。

この制限は、親リレーと DMZ 内のその子リレーの間に持続的な TCP 接続を確立することで打開できます。永続的な接続は、常に親リレーによって開始されます。ネットワーク制限があるため、通信を子リレーによって開始することはできません。

概要

次の図は、親リレーと子リレー間に確立された持続的な TCP 接続を示しています。



この図には以下のものが示されています。

- 緑色: より安全なゾーンにある親リレーと非武装地帯にある子リレー間で確立された持続的な TCP 接続。
- 黄色と黒: 非武装地帯 (DMZ ネットワーク) の範囲を示す線。

親リレーと子リレー双方での永続的な接続の有効化

BigFix クライアントがまだ BigFix サーバーに登録されていなかった子リレーでは

1. BigFix コンソールにログインします。
2. 親リレー・コンピューターで `Relays in DMZ: Enable Parent Relay and set Child Relay List` Fixlet を実行します。

   注: Fixlet を実行する前に、「説明」タブのテキスト・フィールドで子リレーのリストを指定する必要があります。

3. 子コンピューターに BigFix クライアントを手動でインストールします。詳しくは、『Windows でのクライアント・インストール (（ページ）　　)』と『Linux でのクライアント・インストール (（ページ）　　)』を参照してください。
4. 次の Web サイトから、お使いのオペレーティング・システムに適応するパッケージをダウンロードして、子コンピューターに BigFix リレーを手動でインストールします。 http://support.bigfix.com/bes/release/

   注: 一般的なシナリオでは、まず親リレーで Fixlet を実行してから、子リレーを手動で設定します。

5. 子コンピューターで、クライアントとリレーのプロセスが停止していることを確認します。
6. Windows の子リレーで、Windows レジストリーに `HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\BigFix\EnterpriseClient\Settings\Client\_BESRelay_DMZ_ChildEnable` キーを追加し、その文字列値 (REG_SZ) を 1 に設定します。
7. Linux の子リレーでは、besclient.config ファイルがまだ存在していない場合、/var/opt/BESClient/ ディレクトリーにあるファイル besclient.config.default をコ

ピーして名前を besclient.config に変更します。次の新しいセクションを追加して、besclient.config を手動で編集します。

```
[Software\BigFix\EnterpriseClient\Settings\Client\_BESRelay_DMZ_ChildEnable]
value                           = 1
```

8. まずリレー・プロセスを再始動します。
9. リレー・プロセスを再起動して最低 1 分待ってから、クライアント・プロセスを再起動します。

注: 親リレーが認証リレーとして設定されている場合、リレー認証を一時的に無効にし、子リレーの初めての登録が正常にできるようにする必要がある可能性があります。子リレーが正常に登録されたら、リレー認証を再び有効にします。

BigFix クライアントが BigFix サーバーに既に登録されていた子リレー

1. BigFix コンソールにログインします。
2. 親リレー・コンピューターで `Relays in DMZ: Enable Parent Relay and set Child Relay List` Fixlet を実行します。

   注: Fixlet を実行する前に、「説明」タブのテキスト・フィールドで子リレーのリストを指定する必要があります。

3. 子リレー・コンピューターで `Relays in DMZ: Enable Child Relay` Fixlet を実行します。

   注: 一般的なシナリオでは、まず親リレーで Fixlet を実行してから、子リレーで実行します。

4. 両方の Fixlet がリレー・プロセスを再起動します。

永続的な接続の確立

親リレーは、ポート 52311 で子リレーに対してソケットを開こうとします。

子リレーは、親と通信するために親が使用しているソケットをグラブでき、ping メッセージを定期的に送信することによってそのソケットを維持できます。同時に、子リレーはそのループバック・アドレスでのみ、52312 などの異なるポートでの listen を開始します。このポートが使用され、以前にグラブされたソケット (親によって開かれたソケット) を通じてすべてのトラフィックが転送されます。

子リレーに到着するすべての要求のうち (子リレーの下のクライアントの登録が行われるときやレポート目的などで) アップストリームへ伝播する必要があるものは、イントラネット内の親リレーに送信されるようにするため、内部的にループバック・アドレスにルーティングされます。

永続的な接続での通信

この要件を達成するため、親リレーはそれ自体の子リレーと通信を開始し、後で子リレーがアップストリーム通信を必要とする場合に子リレーから親リレーに対して使用できるように、接続が確立した状態を持続させます。

永続的な接続の管理

いくつかの設定を構成することによって DMZ の持続的な接続でリレーを管理できます。詳しくは、DMZ 内のリレー (（ページ）　　)を参照してください。

第 18 章. PeerNest を使用した作業

BigFix クライアントには PeerNest という新機能が搭載されており、この機能を使用すると、同じサブネット内にあるクライアント間でバイナリー・ファイルを共有できます。この機能は、BigFix バージョン 9.5 パッチ 11 以降で使用可能です。

実用的なユース・ケースは、低速リンクを介してデータセンターに接続された支店です。以前の BigFix バージョンで提案された構成では、大きなペイロードをダウンロードしてキャッシュするには、支店でリレーをダウンロードする必要がありました。

PeerNest では、BigFix クライアントはダウンロードしたバイナリーを共有できるため、リレーがローカルにインストールされていない場合でも、支店から送信される通信の数が削減されます。このようにして、リレー上で複数のクライアントが単一クライアントのダウンロード負荷を生成します。1 つのクライアントのみがリレーからダウンロードし、ダウンロードをピアと共有できるためです。

注: BigFix 10.0.7 までは 、異なる BigFix デプロイメントに属するクライアントもホスティングするサブネット上にある BigFix クライアントで有効になっている場合、PeerNest 機能は機能しません。BigFix 10.0.7 以降、異なるデプロイメントに属するクライアントからのピア要求は無視されるため、この制限は適用されなくなります。

PeerNest を使用すると、一部の複雑な BigFix デプロイメント・シナリオでリレーの数を削減できるため、インフラストラクチャー・コストを削減できます。

概要ビデオは、次のリンクから確認できます: https://www.youtube.com/watch?v=tXRX3zlw1aQ。

PeerNest の有効化

PeerNest 機能を有効にするには、クライアントで次の構成設定を 1 に設定します。

```
_BESClient_PeerNest_Enabled = 1
```

クライアントは、バイナリーのダウンロードをローカルに最適化するために、すべての PeerNest 機能を有効にします。

この構成設定を有効にするには、クライアントを再起動する必要があります。

注: アクションの実行に必要なバイナリーのダウンロードを最適化するために BigFix を必要とする場合、ファイルのハッシュがプリフェッチ・ステートメント内で指定されていることを確認します。

PeerNest の構成設定

利用可能なすべての設定について詳しくは、『ピアツーピア・モード (（ページ）)』を参照してください。

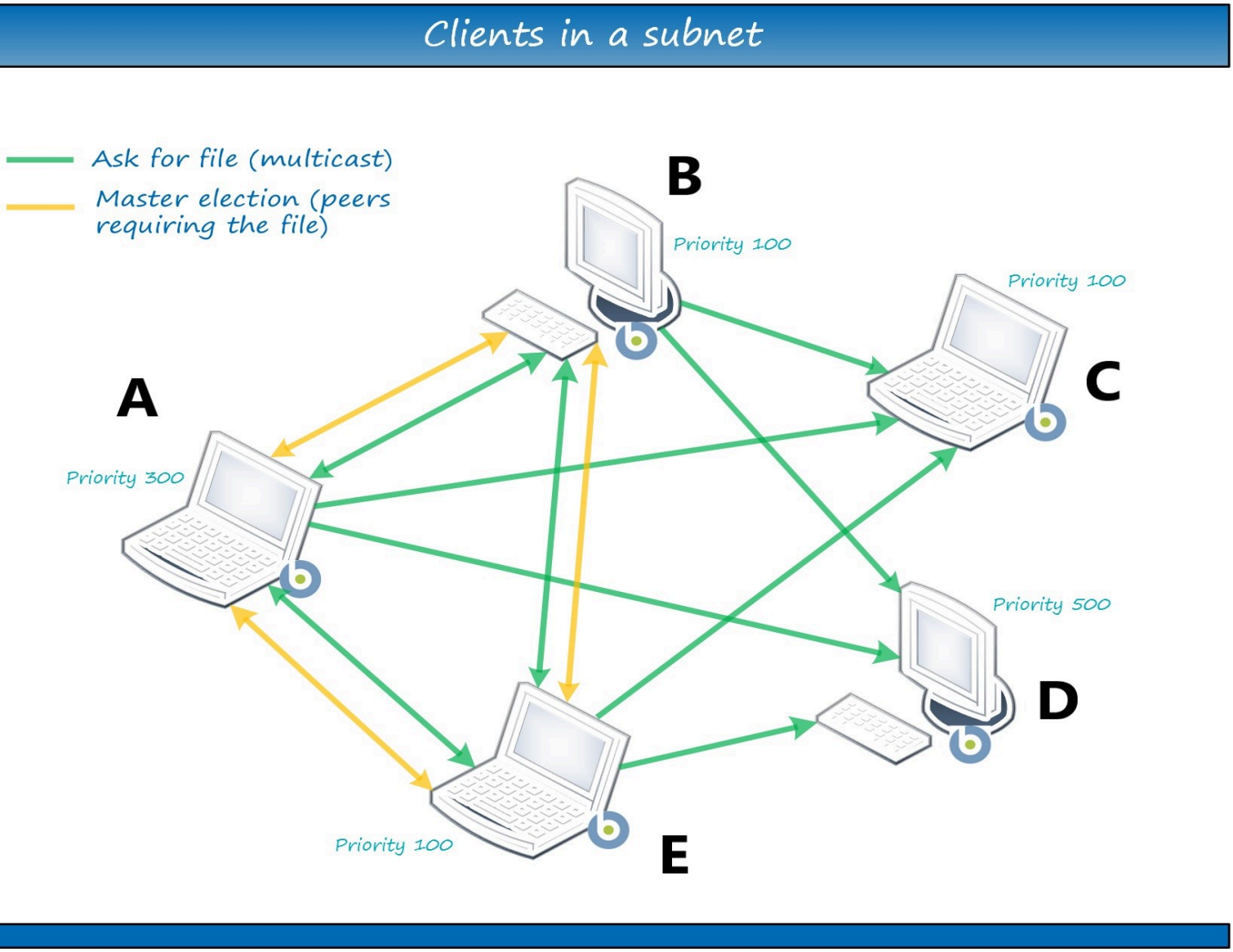
PeerNest の詳細

複数のクライアントがバイナリー・ファイルのプリフェッチを必要とするアクションを実行する場合、クライアントはファイルがサブネット内で既にキャッシュされているかどうかピアに確認します。バイナリーがキャッシュされていない場合、次にクライアントはリレーからのダウンロードを担当するクライアントを 1 つ選択できます (**図 1**)。ファイルを必要とするピアの中で、優先順位が最も高いピアがダウンロードを管理します。すべてのピアが同じ優先順位を持つ場合、ID が最も若いコンピューターがリレーからファイルをダウンロードします。これにより、特定のアクションに対して、すべてのファイルが同じコンピューターによってリレーからダウンロードされる可能性が高いため、ダウンロードが直列化されます (一度に 1 つのファイルが対象)。したがって、リレーとコンピューターの間のリンクで最小帯域幅の占有率が確保されます。

以下の図は、マスター選択プロセスの詳細を示しています。

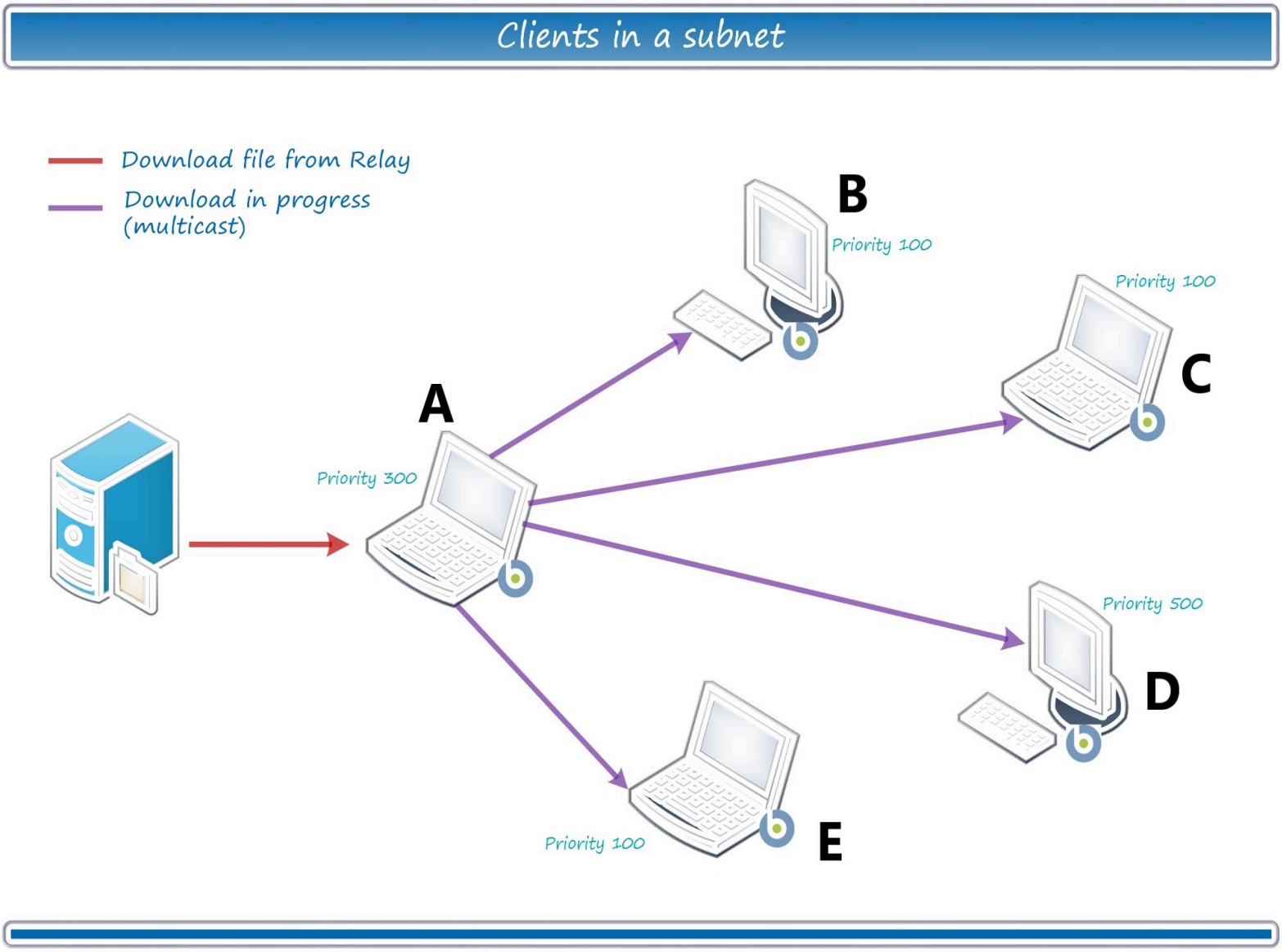
- **A** がマスターとして選択されます。これは、ファイルを必要とするピア (**A**、**B**、**E**) の中で最も優先順位が高いためです。
- **C** および **D** はファイルを必要としないため、**D** の優先順位が **A** より高くても、**D** はマスター選択プロセスに関与しません。

図 1: マスター選択



選択したマスター (**A**) がリレーからのファイルのダウンロードを開始するとすぐに (**図 2**)、ダウンロードが進行中であることを他のピアに通知するため、追加のアクションを実行せずに待機します。マスターは、ダウンロードに関する定期的な通知を送信します。

図 2: ダウンロード進行中



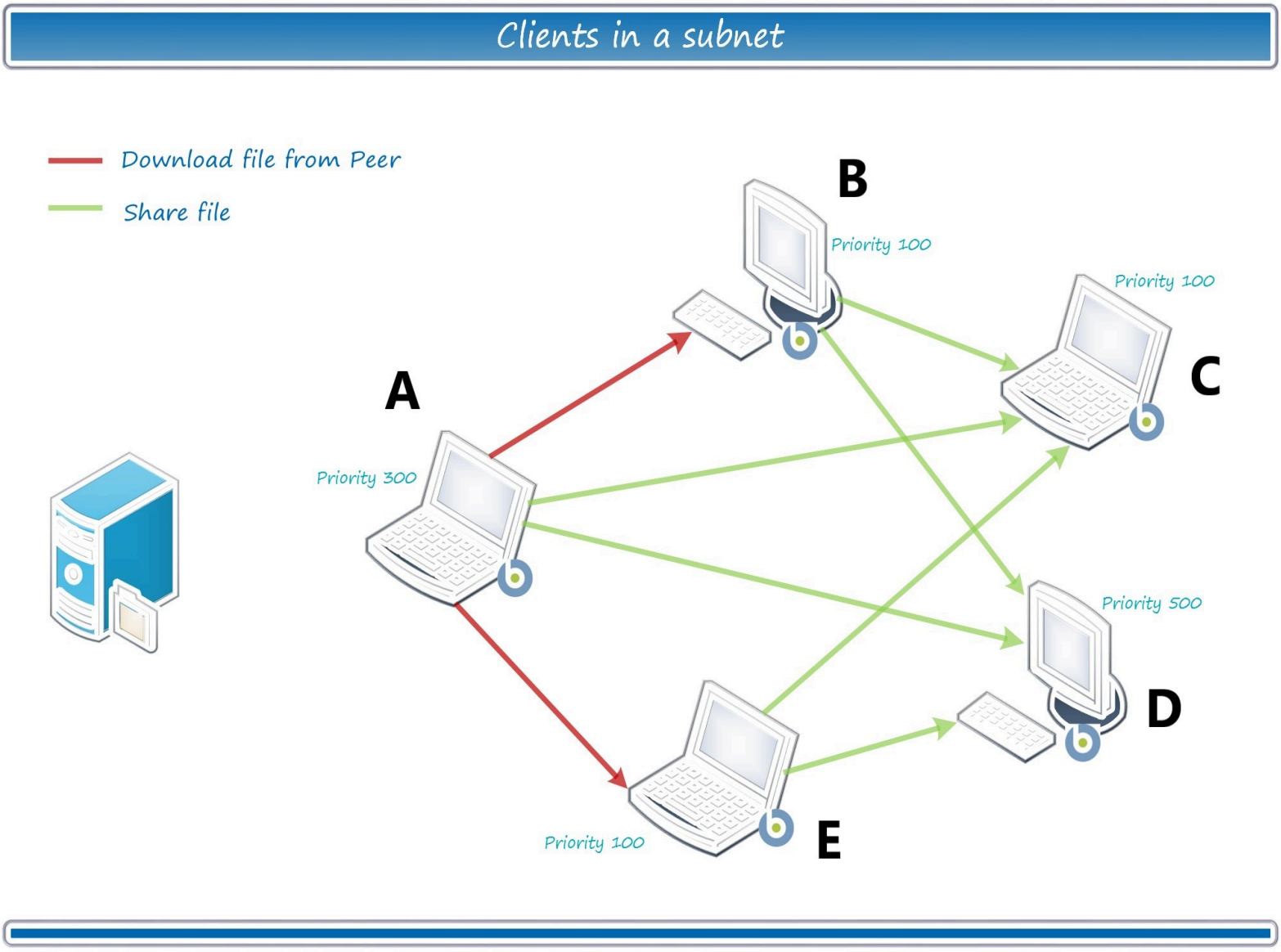
ピアがダウンロード進行中のメッセージを受け取らなかった場合 (ダウンロードの失敗、クライアントのダウン、ネットワークの問題)、新しい選択プロセスが開始され、別のピアがマスターになります。

選択したマスターは、ダウンロードを終了すると、ファイルを PeerNest キャッシュに移動し、その可用性について他のピアに通知します。ファイルに興味を持つピアは、同じメカニズムを使用してマスターからダウンロードを開始し、共有します (**図 3**)。このようにして、リレー上で複数のクライアントが単一クライアントのダウンロード負荷を生成しま

す。1 つのクライアントのみがリレーからダウンロードし、ダウンロードをピアと共有できるためです。

以下の図は、ファイル (**B**、**E**) に関心のあるピアが、リレーからダウンロードするのではなく、ピア **A** から直接ダウンロードを開始する方法を示しています。ダウンロードが完了すると、ファイルはキャッシュされ、今後のために使用できるようになります。したがって、クライアント **A**、**B**、**E** は、ダウンロードしたファイルを **C** および **D** と共有します。

図 3: ファイルの共有



ピアが有効になっているクライアントは、ポート 52311 でピア
転送をリスンする HTTP サーバーを開始します。各クライアント
は、`_BESClient_PeerNest_MaxActiveFileDownloads` の他のピアの最大数を同時に使用できます (デフォルト値は 5)。

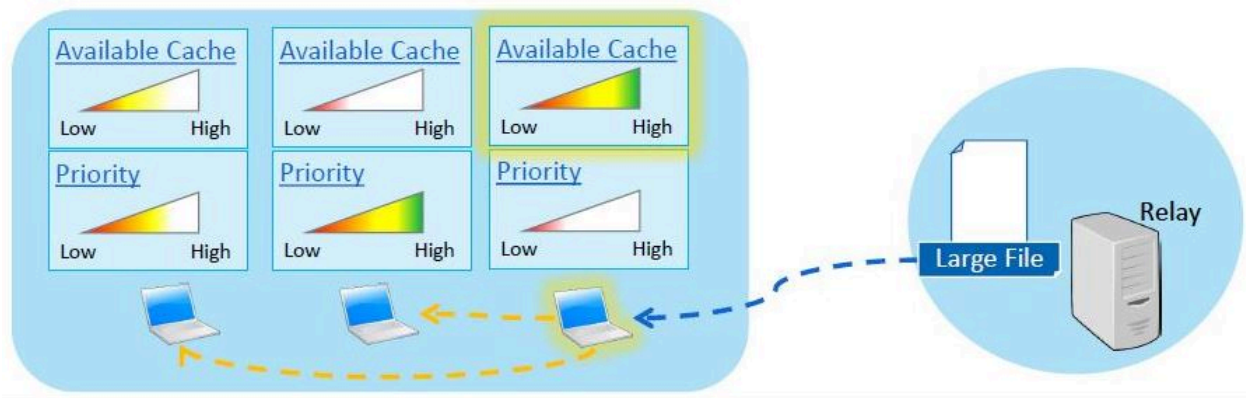
クライアントの優先順位は、同じファイルを共有できるピアが 2 つ以上ある場合にも機能します。ファイルをダウンロードするクライアントは、ファイルを提供するピアのメモリー・リストを作成します。優先順位に基づいて、重み付けされた確率でランダムにピアをピックアップします。例えば、メモリー・リストが優先度 W の C1 と優先度 2W の C2 の 2 つのピアで構成されている場合、C2 のピッキングは C1 の 2 倍の可能性があります。この場合、従来の retry-behavior が適用されます。`_BESClient_Download_RetryMinutes` と `_BESClient_Download_RetryLimit` のクライアントの設定によってルール化されており、以下の点が追加されます。

1. ダウンロード試行の失敗が発生したピアは、メモリー・ピア・リストから削除されるため (そのピアがメモリー内の唯一のピアでない限り)、次の試行は別のピアで行われます。
2. retry-count が制限に達すると、クライアントはリレーから直接ファイルをダウンロードします。

クライアントが接続しようとしたピアが既にダウンロードを 5 回提供しているためにファイルのダウンロードに失敗した場合 (デフォルトの事例)、再試行カウントは増加せず、ピアはメモリー・ピア・リストから削除されません。

PeerCache サイズより大きいファイルの PeerNest 動作

バージョン 10 パッチ 2 以降、新機能により、大きなファイルのプリフェッチを必要とするアクションの時間最適化が可能になりました。この機能により、ダウンロード用に選択されたピアが、単に優先順位が高いファイルではなく、実際にファイルを保管できるピアであることが保証されます。



実際には、バージョン 10.0.2 より前の PeerNest 構成で BigFix クライアントがバイナリー・ファイルのプリフェッチを必要とするアクションを実行している場合、ファイルがサブネットに既にキャッシュされているかどうかを確認します。バイナリー・ファイルがまだキャッシュされていない場合、BigFix クライアントは BigFix リレーからのダウンロードを担当するクライアントを 1 つ選択し、ピアと共有できます。通常、選択されたピアは、割り当てられた優先順位が最も高いピアです。選択されたピアにペイロードを保管するための十分なキャッシュがない場合、配布は失敗し、他のすべてのピアは不必要に待機を行います。

バージョン 10 パッチ 2 以降では、`_BESClient_PeerNest_UseNoSpaceDownload` 設定を使用して PeerNest を構成して、ペイロード・サイズよりも高いキャッシュ制限を持つピアのみをダウンロード対象として選択できます。同じオプションを使用すると、高優先度のピアが強制的に有効になりますが、パッシブ・モードではキャッシュ制限が十分ではありません。このようにして、ピアが無駄な待機を行ってファイルが順次ダウンロードされることを回避できます。このオプションを使用すると、最終的に失敗する可能性のある他のピアを待たずに、BigFix リレーから直接ファイルをダウンロードする機能もピアに提供されます。

サブネットのリストの除外

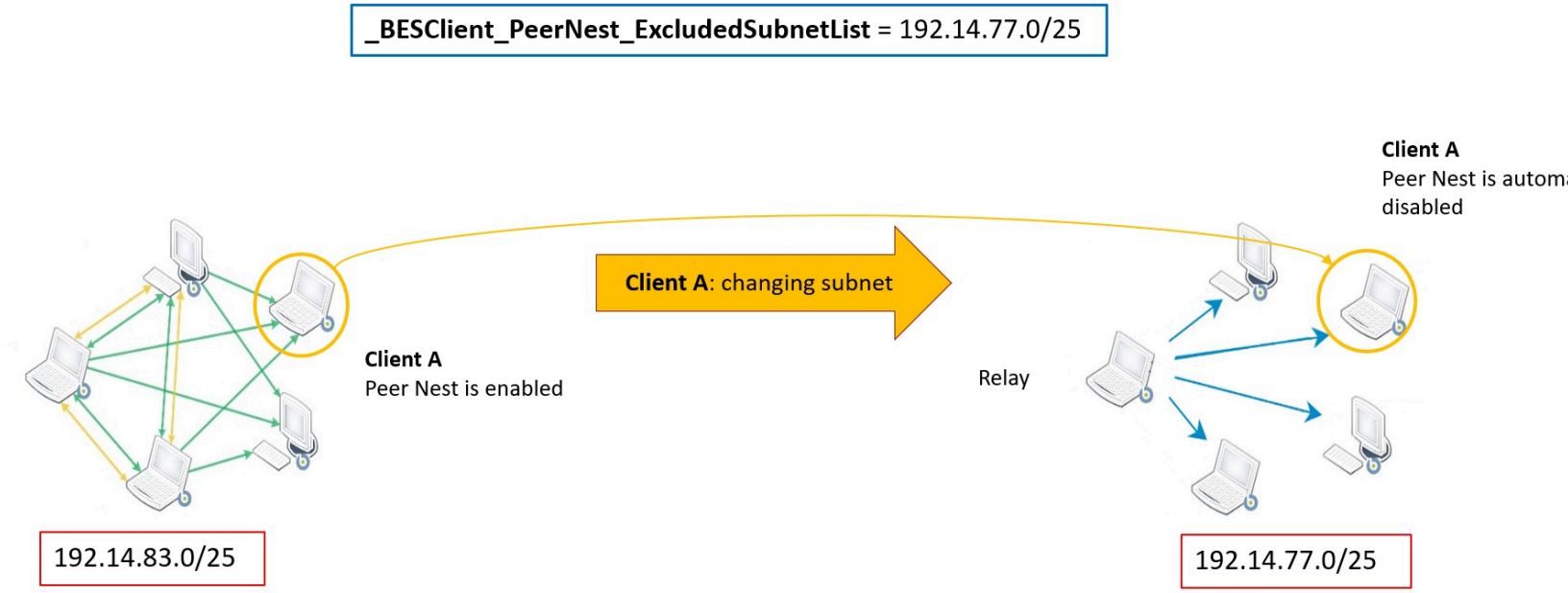
バージョン 10 パッチ 4 以降、新機能により、特定のサブネットに接続されている BigFix クライアン トの PeerNest メカニズムを無効にできます。

PeerNest メカニズムを利用するデバイスでは、UDP 通信プロトコルを用いてメッセージの送受信を行います。同じサブネット上に多数のデバイスが接続されている場合、それら

のデバイスはネットワーク・トラフィックを共有します。これにより、ネットワーク・リソースが過剰に消費される可能性があります。特定のニーズに応じて、一部の特定のネットワーク・サブネット上の UDP トラフィックを制限できます。例えば、VPN インフラストラクチャーで BigFix クライアントを実行している場合などです。

バージョン 10 パッチ 4 以降、サブネットのリストを含む `_BESClient_PeerNest_ExcludedSubnetList` 設定を使用して (例: 192.1.77.0/25;192.1.77.129/25)、リストに指定されたサブネットのいずれかに属するクライアントを PeerNest メカニズムから除外できます。PeerNest が有効になっているクライアント (`_BESClient_PeerNest_Enabled` = 1) が BigFix リレーに登録されている場合、新機能はクライアントのサブネットを識別し、そのサブネットがこのリストに属しているかどうかを確認します。属している場合、PeerNest メカニズムはクライアントに対して無効になります。

さらに、クライアントサブネットを変更する必要がある際 (例: VPN に接続する際)、新しいサブネットが既に除外サブネット・リストに含まれている場合は、次の図に示すように P2P メカニズムが自動的に無効になります。



新しい設定の値を追加/変更/削除するには、BigFix クライアントを再起動する必要があります。

ベスト・プラクティス

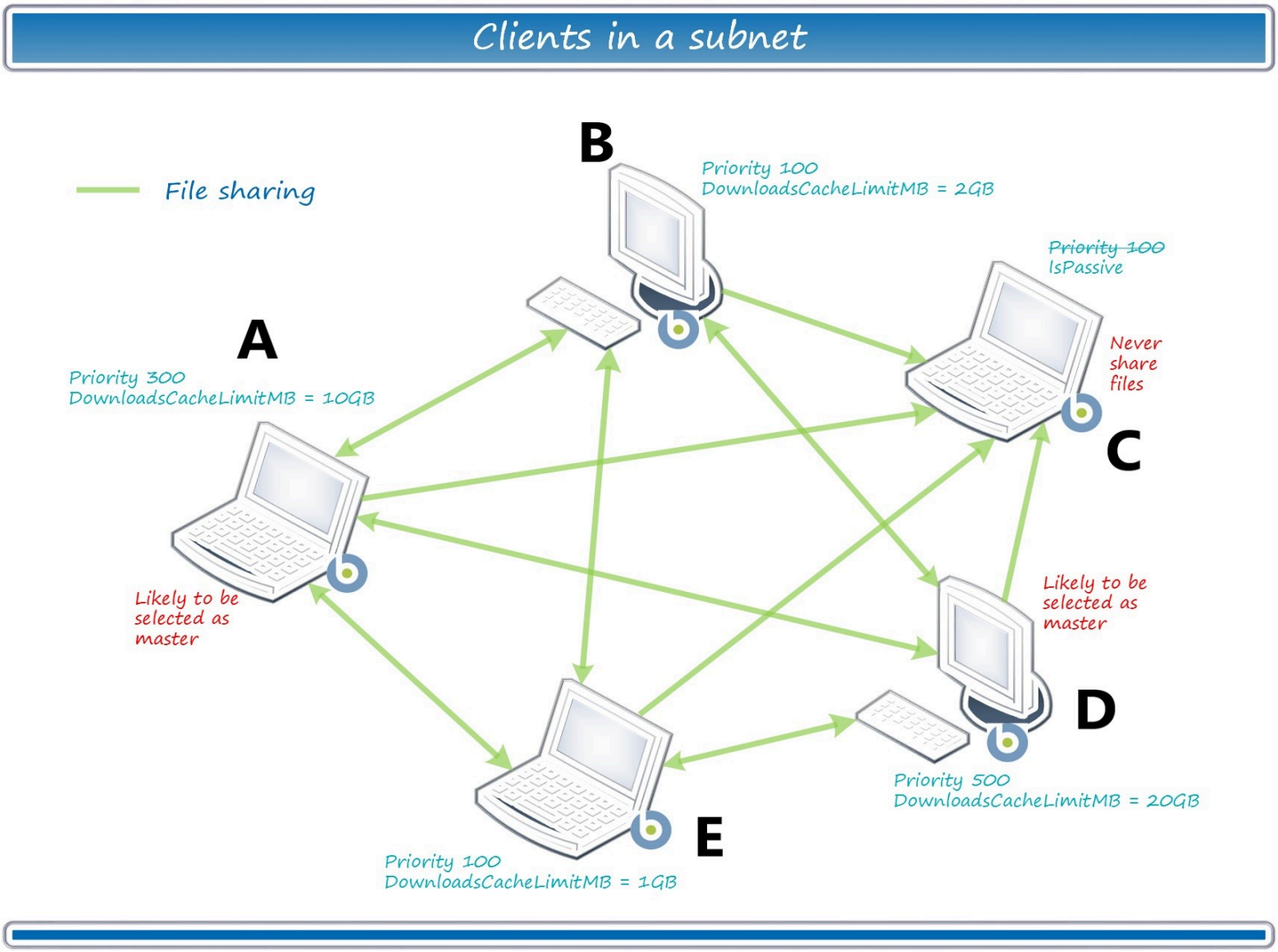
リレーへのより適切なリンクとサブネットの他のピア (そして、場合によっては安定した電源) にサービスを提供するのに十分なリソースを持つコンピューターに、より高い優先順位を割り当てることをお勧めします。

PeerNest では、ファイルをキャッシュするためにディスク・ストレージ・スペースを増やす必要があります。デフォルトの PeerNest キャッシュ・サイズは 2GB で、多くのシナリオで十分な量です。キャッシュに収まるように大きなファイルを転送する場合 (パッチ管理、ソフトウェア配布など) は、サイズを増やす必要があります。PeerNest キャッシュは一時ストレージを目的としているため、以下のパラメーターを使用して使用量と有効期間を微調整できます。`_BESClient_PeerNest_DownloadsCacheLimitMB`、`_BESClient_PeerNest_MinimumDiskFreeMB`、`_BESC`

PeerNest では、BigFix クライアント同士が通信できるようにする処置として、UDP 通信 (ポート 52311) を有効にする必要があります。

また、BigFix クライアントがピアからファイルをダウンロードできるようにする TCP (ポート 52311) と、マルチキャストをサポートするサブネットも必要です。

このポートを開けないクライアントや、キャッシュのために追加のディスク領域を使用したくないクライアントでは、PeerNest をパッシブ・モードで (`_BESClient_PeerNest_IsPassive` 構成設定を使用して) 設定することが推奨されます。パッシブ・クライアントは他のピアからのみダウンロードしますが、コンテンツは共有しません。次の図は、構成の例を示しています。



帯域幅スロットリング

ピアが有効になっているクライアントは、HTTP サーバーを起動します。他のピアはダウンロードのために接続できます。

他のピアにファイルを提供する各クライアントは、その目的のために割り当てられる帯域幅の量を制御できます。この `_BESClient_HTTPServer_ThrottleKBPS` 設定は、クライアントが 1 秒あたりに結合されたすべてのピアに与える合計キロバイト数を定義します (0 は

制限なしを意味します)。値が 1000 KB/秒で、10 のピアが同時にダウンロードしている場合、クライアントは各ピアに 100 KB/秒でデータを送信します (合計 1000 KB/秒)。

トラブルシューティング・シナリオ 1

インターネット・プロトコル・バージョン 6 (**IPv6**) が無効または構成されていないオペレーティング・システムでホストされる BigFix クライアント:

PeerNest 機能を使用する場合は、以下を行う必要があります。

1. これらのクライアントで、`_BESClient_Comm_IPCommunicationsMode` 構成設定を次のように設定します。

   ```
   _BESClient_Comm_IPCommunicationsMode = OnlyIpv4
   ```

2. 変更内容を有効にするために、クライアントを再起動します。

トラブルシューティング・シナリオ 2

`_BESClient_Comm_CommandPollEnable` および `_BESClient_Comm_CommandPollIntervalSeconds` 構成設定を使用して、アクティブなポーリングが設定されている BigFix クライアント:

PeerNest 機能を使用する場合、これらのクライアントを「パッシブ」PeerNest エージェントになるように構成しないでください。これらのクライアントで `_BESClient_PeerNest_IsPassive` 構成設定を有効にしないでください。有効にすると、ポーリングのタイミングによっては、サブネット内の複数のクライアントが同じバイナリーを共有せずにダウンロードできてしまいます。

第 19 章. BigFix でのクライアント・ファイルのアーカイブ

BigFix クライアントから複数のファイルを 1 つのアーカイブに収集し、それらをリレー・システム経由でサーバーに移動することができます。

こうすることで、BigFix 管理者は、特定の管理対象コンピューターのデータを自動的にログに記録することができます。

これを行うために、定期的にまたはコマンドに応じてファイルを収集できる **Archive Manager** という新規コンポーネントが BigFix クライアントに追加されています。これは、結果として圧縮された tarball を BigFix クライアントの **Upload Manager** に渡します。Upload Manager には、アップロードするファイルをキューに入れる入力ディレクトリーがあります。

Upload Manager はアップロード操作を一度に 1 つ実行して、ネットワーク・トラフィックを削減するために、管理しやすいチャンク形式でデータを移動させます。チャンクは一番近い BigFix リレーまたはサーバーに送信されます。そこで、チャンクは **PostFile** プログラムによって元のファイルに再アセンブルされます。

次に、PostFile はこのファイルをチェーンの上流である、次の BigFix リレーまたは BigFix サーバー上の最終的な宛先に渡します。ここでも Upload Manager を使用して、ファイルをチャンクにスライスし、それを階層内の次の PostFile プログラムに送信します。ファイルが最終的に BigFix サーバーに到達すると、クライアント・コンピューターの ID に基づいて、特別なディレクトリー・ロケーションに保存されます。ここまでの段階で、Upload Manager と PostFile プログラムのどちらでも、ネットワーク・トラフィックを滑らかにするために、チャンク・サイズを変更したり、アップロード速度のスロットリングを実行したりできます。

これらのコンポーネントに関連する構成設定については、クライアント・ファイルのアーカイブ (（ページ）　　)を参照してください。

注: Upload Manager は、未登録の BigFix クライアントが検出されると、一時停止します。これは、ネットワークのダウン、サーバーのビジー状態、またはクライアントの切断など、さまざまな理由で発生する場合があります。BigFix クライアント

を BigFix システムに再び登録できるようになるとすぐに Upload Manager が再起動され、中断した場所から処理が続行されます。

Archive Manager の設定

標準的なアーカイブとは、定期的にコンパイルされてサーバーに送信される、ログと構成ファイルのコレクションです。ロギング・ニーズをカスタマイズできるように、多数の設定が用意されています。

このコンポーネントに関連する構成設定について詳しくは、BigFix 構成設定 (（ページ） 325)の「Archive Manager (（ページ）)」を参照してください。

カスタム・アクションの作成

BigFix クライアントに関する属性をアーカイブ・ファイルに送信することができるカスタム・アクションを作成できます。

カスタム・アクションを作成するには、以下の手順を実行します。

1. BigFix コンソールを起動します。
2. **「コンピューター」** タブを選択します。
3. フィルターまたはリストから、アクションのターゲットとするコンピューターのセットを選択します。
4. **「ツール」**メニューから**「カスタム・アクションの実行」**を選択します。
5. **「アクション・スクリプト」** タブを選択します。
6. 表示された **「アクション・スクリプト」** テキスト・ボックスに、目的のアクション・スクリプトを入力します。

Archive Manager

Archive Manager は、定期的にまたはコマンドに応じてファイルを収集できる、BigFix クライアントのコンポーネントです。これは、結果として圧縮された tarball を BigFix クライアントの Upload Manager に渡します。

このコンポーネントに関連する構成設定について詳しくは、BigFix 構成設定 (（ページ） 325)の「Archive Manager (（ページ）)」を参照してください。

Archive Manager の内部変数

Archive Manager コンポーネントの内部変数を以下に示します。

__BESClient_ArchiveManager_LastArchive

Archive Manager は、アーカイブを送信するたびにこの設定を更新します。設定の値は、送信されたファイルのセキュア・ハッシュ・アルゴリズム (sha1) です。

__BESClient_ArchiveManager_LastIntervalNumber

BigFix クライアントは、アーカイブを送信するたびこの設定を更新します。これは、1970 年から、アーカイブが最後に収集されたときまでの間隔数を表します。この間隔が 1 日単位 (デフォルト) の場合、設定には、1970 年から、アーカイブを最後に作成した日までの日数が示されます。この値は、間隔数が変わるときが新規アーカイブが作成される時であるものとして計算されます。

この値は、アーカイブの収集が重ならないよう、コンピューター ID に対応した時間によるオフセットも行なわれます。

Archive Manager のインデックス・ファイル形式

Archive Manager はアーカイブの構築時に、アーカイブに関するメタデータが含まれたインデックスを作成します。

以下は、単一ファイルでのアーカイブのインデックス例です。

```
MIME-Version: 1.0
Content-Type: multipart/x-directory2; boundary="==="
Unique-ID: 1077307147
Archive-Size: 105
SendAll: 0
Date: Wed, 17 Mar 2004 02:23:01 +0000
FileSet-(LOG): c:\temp\log\newfile.log


--===
```

`URL:` file:///c:/temp/log/newfile.log

```
NAME: (LOG)newfile.log
SIZE: 105
TYPE: FILE
HASH: 3a2952e0db8b1e31683f801c6384943aae7fb273
MODIFIED: Sun, 14 Mar 2004 18:32:58 +0000

--===--
```

Upload Manager

Upload Manager は、PostFile プログラムへのチャンク形式でのファイル送信を調整します。帯域幅を確保するためにアップロード・データ・フローをスロットリングすることができます。ファイル・システムは 64 ビットを使用しており、最大バイト長 2^{64} - 1 のファイル・サイズにも十分に対応できます。

このコンポーネントに関連する構成設定について詳しくは、BigFix 構成設定 (（ページ） 325)の「Upload Manager (（ページ）)」を参照してください。

PostFile

PostFile プログラムは、Upload Manager によって送信されたファイルのチャンクを受け取り、それをファイルの独自のコピーに追加します。Upload Manager では、送信されるバイトの範囲と、ファイル名として使用される、ファイルの sha1 を指定します。このコンポーネントに関連する構成設定について詳しくは、BigFix 構成設定 (（ページ） 325)の「PostFile (（ページ）)」を参照してください。

これらのパラメーターは、以下の例のように URL に追加されます。

```
postfile.exe?sha1=51ee4cf2196c4cb73abc6c6698944cd321593007&range=1000,1999,
20000
```

ここで、sha1 値はファイルを識別します。この場合の範囲は、20,000 バイトを 1,000 バイトずつのチャンクに分けたときの 2 つ目のチャンクを指定しています。

PostFile は、ファイルのチャンクを受け取ると、まずそれが正しいセグメントであることを確認します。そうである場合は、送信されたデータをそのファイルのローカル・コピーに追加します。そして、このファイルのサイズに加えて、現在のチャンク・サイズ、およびスロットリングの BPS 設定を返します。

PostFile は、このプログラムにデータを提供する複数の BigFix クライアントを同時に処理する必要があります。その負荷の平衡を取るために、スロットリング比率を調整します。効果的なスロットリング比率は、PostFile の制限比率を、同時にアップロード中のファイル数で除算して判別されます。

例えば、PostFile のスロットリング設定が 100 KBPS であり、50 のクライアントが同時にファイルをアップロードしている場合、各クライアントに返されるスロットリング値は 2 KBPS に調整されます。カスタムのスロットリング値を個別の BigFix リレーに設定することで、ネットワーク内のボトルネックを効率的に処理できます。

PostFile は、部分的にアップロードされたファイルを、そのファイル名の前に下線を付けて Upload Manager のバッファー・ディレクトリーに保管します (Upload Manager は、下線が先頭に付くファイルをアップロードしません)。PostFile は、そのファイルの最後のチャンクを受け取ると、ファイルの sha1 を計算して、それが URL の sha1 パラメーターと一致しているかどうかを確認します。一致している場合は、先行する下線を削除します。

これで、Upload Manager は、ファイルを階層の上位にある次のリレー (あるいは、他のサーバーが指定されている場合はそのサーバー) にアップロードできます。

PostFile は、Upload Manager が稼働中かどうかを判別します。稼働中でなければ、PostFile はすでにそのルート・サーバー宛先に到達したと想定します。その場合、アップロードされたファイルの名前を変更し、アーカイブからファイルを解凍して、Upload Manager のバッファー・ディレクトリーのサブフォルダーにファイルを保管します。

プログラムは、コンピューター ID のモジュラスを使用して、サブフォルダー・パスを計算します。こうすると、ファイル・ディレクトリーへのアクセスを拡散して、単一のディレクトリーにアクセスが集中することを防ぐ効果があります。

例えば、コンピューター ID1076028615 からファイル「log」へのパスは、パス「BufferDir/sha1/15/1076028615/log」に変換されます。ここで、**15** は ID のモジュロ 100 の余り (下位 2 桁) です。

アップロードされたファイルが有効な BigFix アーカイブで、正常に解凍された場合、元のアップロード・ファイルは削除されます。

リソースの例

具体的な一部の例。

例 1

この例では、`c:\log` フォルダー内のすべてのファイルと `c:\myapp` フォルダー内のすべての .ini ファイルを、1 時間に 1 回収集します。サイズが 1,000,000 バイトを超える場合には、異なる部分だけを送信し、アーカイブは送信しないようにします。これを設定するには、BigFix コンソールで以下の設定を作成します。

```
_BESClient_ArchiveManager_FileSet-(Log) = c:\log
_BESClient_ArchiveManager_FileSet-(Ini) = c:\myapp\*.ini
_BESClient_ArchiveManager_OperatingMode = 1
_BESClient_ArchiveManager_Interval_Seconds = 3600
_BESClient_ArchiveManager_SendAll = 0
_BESClient_ArchiveManager_MaxArchiveSize = 1000000
```

例 2

この例では、上記と同じファイル・セットだけでなく、クライアント・コンピューターから一部の有用な属性 (取得したプロパティー) も収集する必要があります。カスタム・アクションを使用すると、これらの属性を生成して、完了時にアーカイブをトリガーできます。使用する設定は上記と同じですが、次のように操作モードを 2 に設定して、**archive now** アクション・コマンドを有効にします。

```
_BESClient_ArchiveManager_OperatingMode = 2
```

ここで、カスタム・アクションを作成して、収集する属性を指定します。例えば、オペレーティング・システム、コンピューター名、および DNS 名をログ・ファイルに追加する場合には、次のようなカスタム・アクションを作成します。

```
appendfile {"System:" & name of operating system}
appendfile {"Computer:" & computer name}
```

```
appendfile {"DNS name:" & dns name}
delete "c:\log\properties.log"
copy __appendfile "c:\log\properties.log"
archive now
```

appendfile コマンドにより、**__appendfile** という名前の一時テキスト・ファイルが作成されます。このコマンドを呼び出すたびに、指定したテキストがこの一時ファイルの末尾に付加されます。

delete コマンドと **copy** コマンドは、古いログ・ファイル (存在する場合) を消去し、__appendfile をログにコピーします。これにより、新しい properties.log ファイルが作成されることになります。OperatingMode が 2 に設定されている限り、**archive now** コマンドによって、アーカイブがすぐに作成されます。

選択した任意のスケジューリングを使用して、このアクションのターゲットを BigFix クライアントの任意のサブセットにすることができます。この方式のバリエーションを使用して、毎晩の差異部分の送信に加えて、週に一度の完全アーカイブを実行できます。

第 20 章. BigFix 構成設定

BigFix には、BigFix スイートの動作を実質的に制御する手段として使用できる詳細な構成設定がいくつかあります。これらのオプションを使用することで、ネットワーク内の BigFix サーバー、リレー、およびクライアントの動作をカスタマイズできます。

概要

この構成設定は、BigFix サーバー、リレー、およびクライアントに適用され、コンソールの**「コンピューターの状態」**ダイアログのカスタム設定構成で管理できます。

- 設定は、ユーザーが変更するまでデフォルト値を使用します。
- 設定に無効な値を指定すると、デフォルト値に戻ります。
- 構成値はすべて、文字列としてレジストリー (または構成ファイル) に格納されます。
- 古い設定が新しい設定をオーバーライドしないように、各設定には「開始日」が関連付けられています。古い開始日のアクションが、新しい開始日の設定を上書きすることはありません。開始日は、アクションが実行された時間に設定されます。
- 数値は文字列として格納され、最大 32 ビットの符号なし整数に収まると想定されます (最大値は 4,294,967,296)。
- 負の数値、最大値より大きい数値、または数値以外の文字を含む数値は、無効な値として処理されます。このような場合、設定はデフォルト値に戻ります。
- ブール値は文字列として格納されます。*true* および *false* に対応して、それぞれ *“1”* または *“0”* として格納されます。これらのいずれの許容値も含まないブール値は、無効な値として処理されます。このような場合、設定はデフォルト値に戻ります。

⚠ **警告:** 構成設定は慎重に使用してください。誤って使用すると、動作が最適でなくなったり、BigFix が適切に機能しなくなる可能性があります。確信が持てない場合は、サポート技術者にお問い合わせください。

詳しくは、「設定のリストと詳細な説明」を参照してください。

第 21 章. 追加の構成手順

これ以降の各トピックでは、現在の環境で実行できる追加の構成手順について説明します。

E メール通知サービスのインストールおよび構成

E メール通知サービスを使用して、実行したの完了状況を通知する自動 E メール通知を送信できます。この通知機能を使用するには、最初にを実行して通知サービスをインストールする必要があります。サービスをインストールした後、別のを実行してサービスを構成します。

BigFix サーバーに BigFix Server Plugin Service をインストールし、BES サポートの Fixlet 1294 を使用して REST API マスター・オペレーターの資格情報を構成する必要があります。

インストールおよびセットアップのが、BES サポート・サイトに用意されています。が確実に正しい順序で実行されるように、インストールおよびセットアップのは、実行する必要がある順序で「関連」状態になります。つまり、が「関連」状態になるのは、それを実行する必要があるときだけです。

以下のステップを実行して、通知サービスをインストールおよびセットアップします。

1. 通知サービスをインストールするには、BES サポートのナビゲーション・ツリーの**「Fixlet とタスク」**ノードから、通知サービスを Windows または Linux のどちらにインストールするかに応じて、以下のいずれかの を選択します。
    - タスク 2238 「`Install Latest Notification Service`」 (Windows オペレーティング・システムに通知サービスをインストールする場合)。このを実行する際に、BigFix サーバーをターゲットにし、通知サービスで listen するポート番号を入力します。このタスクにより、通知サービスがダウンロードされ、インストールされます。
    - タスク 2241 「`Install Latest Notification Service (RHEL)`」 (Linux に通知サービスをインストールする場合)。このを実行する際に、BigFix サーバーをターゲットにし、通知サービスで listen するポート番号を入力します。このタスクにより、通知サービスがダウンロードされ、インストールされます。

インストールが正常に完了しない場合、**「通知」>「警告」**フォルダー内でが「関連」状態になっていないか確認してください。**「警告」**フォルダー内に関連状態のが存在する場合、その関連 を実行します。その の完了後、インストール を再度実行します。

2. 分析 2243 「`Notification Service Settings`」をアクティブにします。
3. 2240 「`Configure Settings for Notification Service`」を実行して、E メール通知サービス用の SMTP ログイン設定を構成します。以下のようにフォームに入力し、**「アクションの実行」**をクリックします。
    - **通知サービス・ポート**: この値を変更して、通知サービスが listen するポート番号を、 2238 「`Install Latest Notification Service`」または 2241「`Install Latest Notification Service (RHEL)`」で設定した値に更新できます。
    - **差出人メール・アドレス**: この値を変更して、通知 E メールの「`From`」フィールドに表示されるデフォルトの `From` アドレスおよび E メール・アドレスを、タスク 2244 **「E メール通知の送信」** (E メール通知を送信するために使用する) で構成したとおりに更新できます。
    - **SMTP メソッド**: 通知サービスの SMTP メソッド (**「標準」**または**「なし」**のいずれか) を選択します。**「なし」**を選択した場合、SMTP 認証は使用されず、**「ユーザー名」**フィールドと**「パスワード」**フィールドは無効になります。**「標準」**を選択した場合、ユーザー名とパスワードを使用した認証が必要になります。
    - **SMTP ホスト**: SMTP サーバーの IP アドレス、ホスト名、または完全修飾ホスト名を入力します。これは必須フィールドです。
    - **SMTP ポート**: デフォルトのポート番号値 25 を受け入れるか、SMTP サーバーのポート番号として別の値を入力します。これは必須フィールドです。
    - **ユーザー名**: E メール・アカウント用のユーザー名を入力します。これは、SMTP メソッドとして**「標準」**を選択した場合は必須フィールドです。

- **パスワード**: E メール・アカウント用のパスワードを入力します。これは、SMTP メソッドとして**「標準」**を選択した場合は必須フィールドです。
- **確認パスワード**: 前のフィールドに入力したパスワードを確認します。これは、SMTP メソッドとして**「標準」**を選択した場合は必須フィールドです。

このステップを完了すると、通知サービスがセットアップされ、作動可能になります。

通知サービスがセットアップされ、設定に従って構成されました。

注: 通知サービスをアンインストールするには、`Task 2239 Uninstall Notification Service` を使用して Microsoft Windows プラットフォームからアンインストールするか、`Task 2242 Uninstall Notification Service (RHEL)` を使用して Linux プラットフォームからアンインストールします。

E メール通知の送信

E メール通知サービスを使用して、実行したベースラインの完了状況を通知する自動 E メール通知を送信できます。

E メール通知を送信するには、通知サービスをセットアップおよび構成する必要があります。

通知サービスをインストールおよび構成した後、ベースラインの完了状況を通知する E メール通知を送信できます。

E メールを通知を送信するには、以下のステップを実行します。

1. 通知サービスを使用するために、 2245 「`Sample Task: Send an Email Notification`」の変更済みコピーを、通知を送信するポイントでにコンポーネントとして追加します。例えば、が完了したときに E メール通知を送信する場合、の最後のコンポーネントとしてを追加します。をコピーして、以下のように変更します。
   - 2245 「`Sample Task: Send an Email Notification using Static Action Script Content`」をコピーします。
   - 関連度「True」以外のすべての関連度を削除し、アクション・スクリプトの詳細を変更し、以下のキーに固有の設定を含めることによって、タスクのコピー

を変更します。以下の例に含まれているキーは、通知を送信するために最低限必要なキーを表しています。

```
// NOTIFICATION_START
// to: "< your comma separated list of recepient email addresses
 goes here >"
// from: "< your single email address that will be shown as the
 sender of
            the email goes here >"
// subject: "< your email title goes here >"
// body: "< your main email detail content goes here >"
// NOTIFICATION_END
```

キーの完全なリストを以下の表に示します。

表 6. 通知のキー

キー	説明\n
`to`	通知の送信先の E メール・アドレス。有効な値は、有効な E メール・アドレスのコンマ (,) 区切りリストです。これは必須キーです。
`from`	通知 E メールの送信者として表示される E メール・アドレス。この E メール・アドレスは存在する必要はありませんが、指定する値は有効な E メール・アドレス形式でなければなりません。これは必須キーです。
`subject`	E メールの件名として表示される通知の要約。任意のテキストがこのキーに対して有効な入力です。入力するテキストには、実行時に置換されるトークンを含めることができます。これは必須フィールドです。
`body`	E メールのメインの通知テキスト・コンテンツ。実行時に置換されるトークンを含む、任意のテキストを

キー	説明\n
	このキーに対して入力できます。これは必須フィールドです。
`failure-trigger`	通知が送信されることになる失敗の回数を定義します。このキーを指定すると、指定した数の失敗が発生したときに通知が送信されます。このキーを省略すると、通知は完了時に送信されます。0 より大きい整数を入力してください。これはオプションのキーです。
`scope`	アクションが失敗したか正常に完了したかを判別する際に、結果状況についてどのアクションを検査するかを決定します。このキーを省略すると、これらの通知コンポーネントを含むアクションの結果が検査されます。有効な値は `parent` です。これは、通知コンポーネントを含むアクションの代わりに、親の結果を検査することを意味します。これはオプションのキーです。

• 変更した のコピーを保存します。

• 変更済みのコピーを、電子メール通知送信の対象である に追加します。

の変更方法の全情報について の記述を確認し、に含めます。サンプルが以下の**「サンプル」**セクションに記載されています。

2. を実行します。 が完了すると、指定したメール・アドレスに電子メール通知が送信されます。

以下のサンプルは、キーの使用方法を示しています。
以下の通知コメントを伴ったタスクをコンポーネントとしてベースラインに追加すると、このタスク (コンポーネント) の完了時に電子メールが送信されます (たとえば、ベースラインの中間に追加すると便利です)。

```
// NOTIFICATION_START
// to: "me@me.com, you@you.com"
// from: "noreply@bigfixteam.mycompany.com"
```

```
// subject: "Basline component '{actionName}' has completed successfully"
// body: "Baseline is 50% complete now"
// NOTIFICATION_END
```

以下の通知コメントで構成された をコンポーネントとして に追加すると、 全体の完了時に電子メールが送信されます。

```
// NOTIFICATION_START
// to: "me@me.com, you@you.com"
// from: "noreply@bigfixteam.mycompany.com"
// subject: "Basline '{actionName}' with ID {actionID} has completed
 successfully"
// body: "The Baseline is complete!"
// scope: "parent"
// NOTIFICATION_END
```

上の 2 つの例は、静的に対象指定されたベースラインにのみ追加できます。グループ、プロパティー、または名前リストにより対象指定されたベースラインでは、通知送信は有効になりません。

静的または動的に対象指定されるベースラインでは、以下のサンプルを指定できます。

```
// NOTIFICATION_START
// to: "me@me.com, you@you.com"
// from: "noreply@bigfixteam.mycompany.com"
// subject: "Basline '{actionName}' with ID {actionID} has failed on 5 or
 more computers"
// body: "Review the results of Baseline '{actionName}' (ID: {actionID})"
// failure-trigger: "5"
// scope: "parent"
// NOTIFICATION_END
```

FillDB の構成

FillDB プロセスは、BigFix サーバー・システムで実行され、BigFix エージェントから返された情報を BigFix データベースに保管します。

この情報には、以下のようなものがあります。

- データ (取得されたプロパティーの値や、適用性の関連度の評価結果、Fixlet とタスクの成功条件など)。
- BigFix Query の結果を返すレポートに含まれる情報。

FillDB 処理を構成する方法は以下のとおりです。

- FillDB データベースのパフォーマンスの構成 (（ページ） *332)*
- FillDB バッファー・ディレクトリーのサイズの増加 (（ページ） *334)*
- FillDB 並列処理の有効化 (（ページ） *335)*

FillDB データベースのパフォーマンスの構成

DatabaseBoostLevel パラメーター

FillDB ユーティリティーには、BigFix のパフォーマンスを最適化するための `DatabaseBoostLevel` というパラメーターが用意されています。

Windows システムの場合:`DatabaseBoostLevel` パラメーターに指定できる値は、1 (有効) と 0 (無効) です。デフォルト値は 1 です。

Linux システム (フレッシュ・インストールおよびアップグレード) の場合:`DatabaseBoostLevel` パラメーター値は常に、「Database Boost Level」が「ON」で、maxBatchSize = 1000 です。

BigFix のパフォーマンスの向上は、BigFix が実行されている環境によって異なります。ご使用の環境に対してパフォーマンス構成を最適化するには、以下のように FillDB データベースでデータ挿入メカニズムを調整します。

1. パフォーマンス・ログを有効にします。
   以下のストリング値を `[HKLM\Software\Wow6432Node\BigFix\Enterprise Server\FillDB]` レジストリーに設定します。

```
"PerformanceDataPath"[REG_SZ] = "[BigFix Server
 folder]\FillDB\FillDBperf.log"
```

2. FillDB サービスを再始動し、しばらくパフォーマンス・ログをモニターして、さまざまな表の「行/秒 (rows per second)」のデータベース挿入レートを記録します。
3. `DatabaseBoostLevel` DWord 値をレジストリー・キー `HKLM\Software\Wow6432Node\BigFix\Enterprise Server\FillDB` に追加し、これを 0 に設定します。
4. FillDB サービスを再始動し、再度しばらくパフォーマンス・ログをモニターして、さまざまな表の「行/秒 (rows per second)」のデータベース挿入レートを記録します。
5. `DatabaseBoostLevel` パラメーターの新しい値を設定する前と後で挿入レートを比較します。モニター中は作業負荷のレベルを同じに保ってください。秒あたりに処理される行数が多いほど、パフォーマンスは向上します。モニター対象とする主な表は、questionresults、fixletresults、actionresults、および longquestionresults です。表あたりに処理される行数は、ご使用の環境内でのその表の重要度を示します。

DisableReplicationOfNextTables パラメーター

FillDB ユーティリティーには、 DSA インフラストラクチャーでサーバー間の WEbUI テーブルの複製を無効化するための `DisableReplicationOfNextTables` というパラメーターが用意されています。

Windows システムの場合: 以下の DWORD 値を `[HKLM\Software\Wow6432Node\BigFix\Enterprise Server\FillDB]` レジストリーに設定します。

```
"DisableReplicationOfNextTables"=1
```

Linux システムの場合:`[Software\BigFix\Enterprise Server\FillDB]` セクションの下の BESServer構成ファイルで、以下のパラメーターを設定します。

```
DisableReplicationOfNextTables=1
```

バージョン 9.5 パッチ 9 以降、このパラメーターは、APAR IJ05097 の影響を緩和するために使用可能です。

この設定を有効にすると、WebUI テーブルが DSA サーバー間で複製されるのを防止します。複製処理の速度が向上します。複製サーバー上で災害復旧およびフェイルオーバーが

発生した場合、WebUI データを複製サーバー上で使用できなくなります。このため、このパラメーターはサポート管理下においてのみ設定することを強くお勧めします。

FillDB バッファー・ディレクトリーのサイズの増加

FillDB バッファー・ディレクトリーは、クライアントからのレポートがデータベースに保管される前に、それらを一時的に保管します。

デフォルトでは、ディレクトリーがいっぱいになるのは、含まれるファイルの合計が 3MB を超える場合、または含まれるファイルが 10,000 個を超える場合です。その結果、情報が BigFix サーバーに迅速に送信されず、深刻な問題となる可能性があります。

以下の手順を実行して、FillDB バッファー・ディレクトリーと、保持ファイルの最大数を構成することができます。

Windows システムの場合:

1. 以下のキーをレジストリー・パスに追加します。`HKLM\Software\Wow6432Node\BigFix\Enterprise Server\PostResults`:

   BufferDirectoryMaxSize

   FillDB バッファー・ディレクトリーの最大サイズをバイト単位で定義します。デフォルト値は 3MB です。

   注: HCL サポートからの具体的な指示がない限り、この値を 20MB よりも大きくしないでください。

   BufferDirectoryMaxCount

   FillDB バッファー・ディレクトリーに許可されるファイルの最大数を定義します。デフォルト値は 10,000 です。
2. FillDB サービスを再始動します。

注: Windows システムの場合、これらのキーを BigFix サーバーではなく、BigFix リレーに追加すると、キーを追加するレジストリー・パスは `HKLM\Software\BigFix\Enterprise Server\PostResults` になります。

Linux システムの場合:

1. 以下の行を `/var/opt/BESServer/besserver.config` ファイルに追加します。

```
[Software\BigFix\Enterprise Server\PostResults]
BufferDirectoryMaxSize = <SIZE_IN_BYTES>

[Software\BigFix\Enterprise Server\PostResults]
BufferDirectoryMaxCount = <MAX_NUMBER_OF_FILES>
```

   各部の意味は以下のとおりです。

   BufferDirectoryMaxSize

   FillDB バッファー・ディレクトリーの最大サイズをバイト単位で定義します。デフォルト値は 3MB です。

   BufferDirectoryMaxCount

   FillDB バッファー・ディレクトリーに許可されるファイルの最大数を定義します。デフォルト値は 10,000 です。
2. FillDB サービスを再始動します。

FillDB 並列処理の有効化

BigFix サーバーで、FillDB プロセスは、以下のアクティビティーを単一スレッドで実行します。

- バッファー・ディレクトリー・コンテンツの読み取り。
- 以下を含む、レポートの解析。
  - 暗号化されたレポートの復号。
  - 圧縮されたレポートの解凍。
- データベースへのレポート・データの保管。
- 他の DSA サーバーからのコンテンツの複製 (オプション)。

追加のスレッドが、BigFix Query 処理によって返されたレポートに対して同じタイプの処理を実行します。

V9.5 パッチ 5 以降、フレッシュ・インストール時およびアップグレード時に以下のルールに従い、並列処理はデフォルトで有効になります。

- マシンに 6 個から 9 個のコアがある場合、3 個の解析スレッドおよび 3 個のデータベース更新スレッドを構成することにより、並列処理は標準レポートに対して有効になります。
- マシンに少なくとも 10 個のコアがある場合、並列処理は標準レポートと照会レポートの両方に対して有効になります。それぞれのレポート用に 3 個の解析スレッドおよび 3 個のデータベース更新スレッド、合計で 12 個のスレッドを構成します。

BigFix サーバーで以下の設定を構成することにより、FillDB 並列処理を手動で有効または無効にすることができます。

- ParallelismEnabled
- ParallelismEnabledForQuery

FillDB 並列処理を有効にした後、BigFix サーバーで以下の設定を指定することによって、その動作を構成できます。

- NumberOfParsingThreads
- NumberOfDBUpdatingThreads
- MaxNumberOfReportsReadyForDB
- MinNumberOfReportsReadyForDB
- MaxNumberOfReportsInParsingQueue
- NumberOfParsingThreadsForQuery
- NumberOfDBUpdatingThreadsForQuery
- MaxNumberOfQueryReportsReadyForDB
- MinNumberOfQueryReportsReadyForDB
- MaxNumberOfQueryReportsInParsingQueue

これらの設定について詳しくは、「並列 FillDB の構成 (（ページ） *337*) 」を参照してください。
上記の 1 つ以上の設定に対する変更をアクティブ化するには、以下のステップを実行します。

BigFix サーバーが Windows システムにインストールされている場合:

1. BES FillDB サービスを停止します。
2. Windows レジストリーで設定の値を必要に応じて更新します。
3. BES FillDB サービスを開始します。

BigFix サーバーが Linux システムにインストールされている場合:

1. besfilldb を停止します (例: `/etc/init.d/besfilldb stop`)。
2. besserver を停止します (例: `/etc/init.d/besserver stop`)。
3. `besserver.config` ファイルで設定の値を必要に応じて更新します。
4. besserver を開始します (例: `/etc/init.d/besserver start`)。
5. besfilldb を開始します (例: `/etc/init.d/besfilldb start`)。

並列 FillDB の構成

名前/説明	値		影響を受けるコンポーネント	適用可能なバージョン	参照
ParallelismEnabled この設定では、エージェン	デフォルト値	1 (有効)	サーバー	9.5.5 以降	FillDB 並列処理の有効化 (（ページ） *335)*
	設定タイプ	Windows では REG_DWORD、Linux では besserver.config			

名前/説明	値		影響を受けるコンポーネント	適用可能なバージョン	参照
ト・レポートの並列処理を有効にできます。並列が有効になる状況についての詳細は、「FillDB並列処理		のエントリー			
	値の範囲	0 ～ 1			
	使用可能なタスク	いいえ			
	再起動が必要	Windowsシステムでは FillDBの再起動、Linuxでは FillDBおよび BesRootServerの再起動			

名前/説明	値		影響を受けるコンポーネント	適用可能なバージョン	参照
の有効化 (（ページ） *335)*」を参照してください。					
NumberOfParsingThreads この設定では、レポートの解析を担当するスレッ	デフォルト値	3	サーバー	9.5.5 以降	FillDB 並列処理の有効化 (（ページ） *335)*
	設定タイプ	Windows では REG_DWORD、Linux では besserver.config のエントリー			
	値の範囲	1 ～ 5			
	使用可能なタスク	いいえ			

名前/説明	値		影響を受けるコンポーネント	適用可能なバージョン	参照
ド数を定義します。並列が有効な場合にのみ使用されます。	再起動が必要	Windows システムでは FillDB の再起動、Linux では FillDB および BesRootServer の再起動			
NumberOfDBUpdatingThreads この設定では、DB にレポート・デー	デフォルト値	3	サーバー	9.5.5 以降	FillDB 並列処理の有効化 (（ページ） *335)*
	設定タイプ	Windows では REG_DWORD、Linux では besserver.config のエントリー			

名前/説明	値		影響を受けるコンポーネント	適用可能なバージョン	参照
タを保管するスレッド数を定義します。並列が有効な場合にのみ使用されます。	値の範囲 使用可能なタスク 再起動が必要	1 ～ 5 いいえ Windows システムでは FillDB の再起動、Linux では FillDB および BesRootServer の再起動			
MaxNumberOfReportsReadyForDB この設定は、	デフォルト値 設定タイプ	1000/ NumberOfDBUpdatingThreads Windows では	サーバー	9.5.5 以降	FillDB 並列処理の有効化（ページ） *335)*

名前/説明	値		影響を受けるコンポーネント	適用可能なバージョン	参照
スレッドを更新するDB が 1 回の実行で処理できる最大レポート数を示します。並列が有効な場合にのみ使		REG_DWORD、Linux では besserver.config のエントリー			
	値の範囲	500 ～ 5000/ NumberOfDBUpdatingThreadsStandard			
	使用可能なタスク	いいえ			
	再起動が必要	Windows システムでは FillDB の再起動、Linux では FillDB および BesRootServer の再起動			

名前/説明	値		影響を受けるコンポーネント	適用可能なバージョン	参照
用されます。					
MinNumberOfReportsReadyForDB この設定は、スレッドを更新するDB が 1 回の実行で処理できる最小レポート数を示	デフォルト値	MaxNumberOfReportsReadyForDB/2	サーバー	9.5 以降	FillDB 並列処理の有効化 (（ページ） *335)*
	設定タイプ	Windows では REG_DWORD、Linux では besserver.config のエントリー			
	値の範囲	MaxNumberOfReportsReadyForDB/1 ～ 3			
	使用可能なタスク	いいえ			
	再起動が必要	Windows システムでは FillDB の再起動、Linux			

名前/説明	値		影響を受けるコンポーネント	適用可能なバージョン	参照
します。 並列が有効な場合にのみ使用されます。		では FillDB および BesRootServer の再起動			
MaxNumberOfReportsInParsingQueue この設定は、解析スレッドが処理するのを待機	デフォルト値	10 * NumberOfParsingThreads	サーバー	9.5.5 以降	FillDB 並列処理の有効化 (（ページ） *335)*
	設定タイプ	Windows では REG_DWORD、Linux では besserver.config のエントリー			
	値の範囲	(2 ～ 20) * NumberOfParsingThreads			

名前/説明	値		影響を受けるコンポーネント	適用可能なバージョン	参照
しているキューに入れることのできる最大レポート数を示します。並列が有効な場合にのみ使用されます。	使用可能なタスク	いいえ			
	再起動が必要	Windows システムでは FillDB の再起動、Linux では FillDB および BesRootServer の再起動			

名前/説明	値		影響を受けるコンポーネント	適用可能なバージョン	参照
ParallelismEnabledForQuery この設定では、BigFix照会レポートの並列処理を有効にできます。	デフォルト値	0	サーバー	9.5.5以降	なし
	設定タイプ	WindowsではREG_DWORD、Linuxではbesserver.configのエントリー			
	値の範囲	0 ～ 1			
	使用可能なタスク	いいえ			
	再起動が必要	Windowsシステムでは FillDBの再起動、Linuxでは FillDBおよび			

名前/説明	値		影響を受けるコンポーネント	適用可能なバージョン	参照
		BesRootServer の再起動			
NumberOfParsingThreadsForQuery この設定では、BigFix 照会レポートの解析を担当するスレッド数を定義します。BigFix 照会レ	デフォルト値	3	サーバー	9.5.5 以降	FillDB 並列処理の有効化 (（ページ） *335)*
	設定タイプ	Windows では REG_DWORD、Linux では besserver.config のエントリー			
	値の範囲	1 ～ 5			
	使用可能なタスク	いいえ			
	再起動が必要	Windows システムでは FillDB の再起動、Linux では FillDB			

名前/説明	値		影響を受けるコンポーネント	適用可能なバージョン	参照
ポートの並列処理が有効化されている場合のみ使用されます。		および BesRootServer の再起動			
NumberOfDBUpdatingThreadsForQuery この設定では、DB に BigFix 照会レ	デフォルト値 設定タイプ	3 Windows では REG_DWORD、Linux では besserver.config のエントリー	サーバー	9.5.5 以降	FillDB 並列処理の有効化 (（ページ） *335)*

名前/説明	値		影響を受けるコンポーネント	適用可能なバージョン	参照
ポート・データを保管するスレッド数を定義します。BigFix 照会レポートの並列処理が有効化されている場	値の範囲	1 ～ 5			
	使用可能なタスク	いいえ			
	再起動が必要	Windows システムでは FillDB の再起動、Linux では FillDB および BesRootServer の再起動			

名前/説明	値		影響を受けるコンポーネント	適用可能なバージョン	参照
合のみ使用されます。					
MaxNumberOfQueryReportsReadyForDB この設定は、スレッドを更新するDBが1回の実行で処理できるBigFix照会	デフォルト値	1000/NumberOfDBUpdatingThreadsForQuery	サーバー	9.5.5以降	FillDB 並列処理の有効化 (ページ 335)
	設定タイプ	WindowsではREG_DWORD、Linuxではbesserver.configのエントリー			
	値の範囲	(500 ～ 5000)/NumberOfDBUpdatingThreadsForQuery			
	使用可能なタスク	いいえ			
	再起動が必要	Windowsシステムでは			

名前/説明	値	影響を受けるコンポーネント	適用可能なバージョン	参照
レポートの最大数を示します。BigFix 照会レポートの並列処理が有効化されている場合のみ使用されます。	FillDB の再起動、Linux では FillDB および BesRootServer の再起動			

名前/説明	値		影響を受けるコンポーネント	適用可能なバージョン	参照
MinNumberOfQueryReportsReadyForDB この設定は、スレッドを更新するDBが1回の実行で処理できるBigFix照会レポートの最小数を示	デフォルト値	MaxNumberOfQueryReportsReadyForDB/2	サーバー	9.5.5以降	FillDB 並列処理の有効化 (（ページ） *335)*
	設定タイプ	Windowsでは REG_DWORD、Linuxでは besserver.config のエントリー			
	値の範囲	MaxNumberOfQueryReportsReadyForDB/(3 ～ 1)			
	使用可能なタスク	いいえ			
	再起動が必要	Windowsシステムでは FillDB の再起動、Linuxでは FillDB および			

名前/説明	値		影響を受けるコンポーネント	適用可能なバージョン	参照
します。BigFix 照会レポートの並列処理が有効化されている場合のみ使用されます。		BesRootServer の再起動			
MaxNumberOfQueryReportsInParsingQueue この設定は、解	デフォルト値 設定タイプ	10 * NumberOfParsingThreadsForQuery Windows では REG_DWORD、Linux	サーバー	9.5.5 以降	FillDB 並列処理の有効化 (ページ 335)

名前/説明	値		影響を受けるコンポーネント	適用可能なバージョン	参照
析スレッドが処理するのを待機しているキューに入れることのできるBigFix照会レポートの最大数を示しま		では besserver.config のエントリー			
	値の範囲	(2 〜 20) * NumberOfParsingThreadsForQuery			
	使用可能なタスク	いいえ			
	再起動が必要	Windows システムでは FillDB の再起動、Linux では FillDB および BesRootServer の再起動			

名前/説明	値	影響を受けるコンポーネント	適用可能なバージョン	参照
す。BigFix 照会レポートの並列処理が有効化されている場合のみ使用されます。				

ODBC データ・ソースの構成

Open DataBase Connectivity (ODBC) を使用して、BigFix とともに ODBC データ・ソースをセットアップおよび構成する方法。

データベースを使用する BigFix コンポーネントは、Open DataBase Connectivity (OBDC) を用いてデータベースに接続します。

ODBC 接続を開くには、以下が必要です。

- ODBC ドライバー。アプリケーションによるデータベース・エンジンの ODBC API の呼び出しを可能にするソフトウェアの一部。
- ODBC データ・ソース。ターゲット・データベースを識別し、接続パラメーターの詳細を示す情報。

同じコンピューターに複数の ODBC ドライバーがインストールされ、複数の ODBC データ・ソースがそこに保存されている場合があります。多くの場合、データ・ソースはターゲット・データベースへの接続に使用するドライバーも指定します。

各 ODBC データ・ソースは、データ・ソース名 (DSN) によって識別されます。DNS (ドメイン・ネーム・システム) と混同しないよう注意してください。頭字語は似ていますが、内容はまったく異なります。

Windows の ODBC

Windows では、ODBC データ・ソースは 32 ビットまたは 64 ビットです。1 つが 32 ビットで、もう 1 つが 64 ビットの場合、同じ名前のソースが 2 つ存在する可能性があります。

同じ名前でビットが異なる ODBC データ・ソースは、多くの場合同じデータベースを参照し、指定する ODBC ドライバーのパスを除いて同じ接続パラメーターを有します。

ODBC ドライバでは多くの場合、32 ビットアプリケーションの場合は 32 ビット dll、64 ビットアプリケーションの場合は 64 ビット dll があります。

ODBC データ・ソースは、以下のレジストリーに保存されます。

- 32 ビット ODBC データ・ソースは、HKEY_LOCAL_MACHINE SOFTWAREWow6432Node ODBC ODBC に表示されます。
- 64 ビット ODBC データ・ソースは、HKEY_LOCAL_MACHINE SOFTWARE ODBC ODBC に表示されます。

これらのツールは、直接編集することも、以下の専用ツールを使用して構成することもできます。

- 開始 > windows 管理ツール > ODBC データ・ソース (32 ビット)
- 開始 > Windows 管理ツール> ODBC データ・ソース (64 ビット)

BigFix プラットフォーム・コンポーネントは、Windows 上で次の ODBC データ・ソースを使用します。

- enterprise_setup。特にアップグレード時にインストーラーによって使用される
- bes_bfenterprise。BigFix サーバーがデータベースにアクセスするために使用される
- LocalBESReportingServer。Web レポートがそのデータベースにアクセスするために使用される

上記のすべての ODBC データ・ソース名には、対応する 32 ビット・ソースと 64 ビット・ソースがあります。

合計で 6 つの BigFix データ・ソース (3 つのペア) があります。

BigFix 管理ツール (BESAdmin) は通常、BigFix サーバーと同じ ODBC データ・ソースに依存しますが、実行するコマンドによって異なるデータ・ソースを使用する場合があります。

BigFix WebUI は ODBC データ・ソースを使用せず、その接続データを内部的に保存します。

BigFix ODBC データ・ソースを変更する場合は、次に留意してください。

- データ・ソースの 32 ビット・バージョンと 64 ビット・バージョンの両方を更新する。
- 他のデータ・ソースの対応する設定に一貫性があることを確認する。

すべてのデータ・ソースを調整しない場合は、特定の BigFix 機能を使用しているときにデータベース接続の問題が発生する可能性がありますが、他の機能を使用している間は問題は発生しません。

例えば、bes_bfenterpriseを変更し、enterprise_setupの対応する情報を更新し忘れた場合、アップグレード前のチェック時 BigFix にのみ検出される場合があります。

BigFix コンポーネントは、SQLサーバーでホストされているデータベースに接続できます。接続方法は次の 2 通りです。

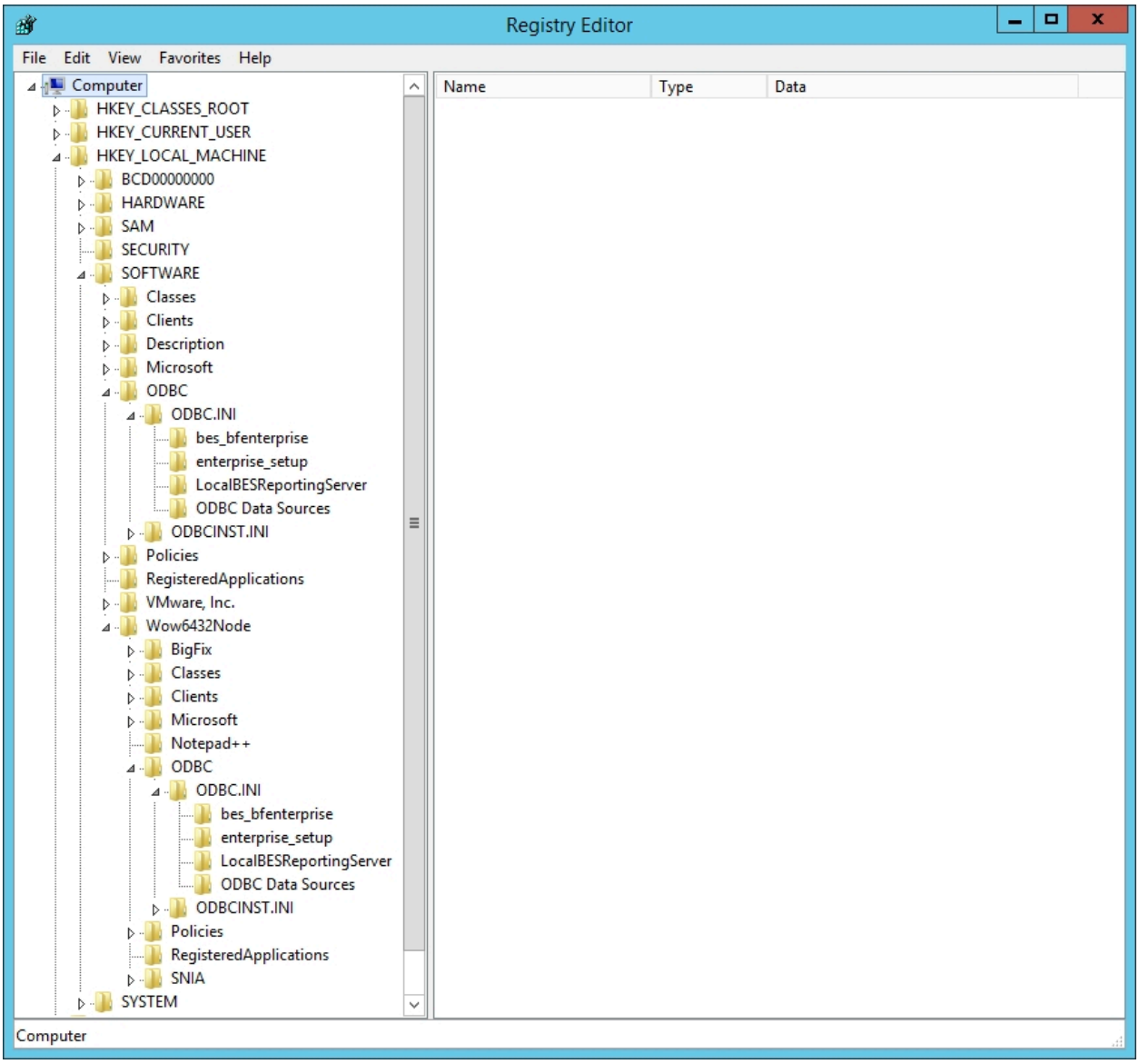
- Windows 認証経由。アプリケーション・プロセスを実行するユーザーの Windows 資格情報に依存します。
- SQLサーバー認証経由。Windows ユーザー資格情報とは関係のない別個の資格情報に依存します。

BigFix コンポーネントが Windows 認証を使用する場合、そのサービスを実行しているユーザーは、データベースにアクセスするユーザーと同じです。これは、BigFix WebUI を除くすべてのコンポーネントに当てはまります。

ローカル・データベースに接続する場合、BigFix はデフォルトで Windows 認証を使用します。

リモート・データベースに接続する場合、 BigFix は Windows 認証または SQL サーバー認証を使用するよう構成できます。

このイメージは、BigFix サーバーと Web レポートの両方がインストールされているコンピューターの ODBC キーを示しています。



この構成コードは、ローカル・データベースを使用し、接続にネイティブ・クライアントを使用するように構成された、サーバーおよび Web レポートを使用 BigFix するコンピューターの 64 ビットおよび 32 ビット ODBC キー Microsoft SQL Server示しています。

```
[HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBC.INI]


[HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBC.INI\bes_bfenterprise]
"Driver"="C:\\Windows\\system32\\sqlncli11.dll"
"Server"="(local)"
```

```
"Database"="BFEnterprise"
"LastUser"="SYSTEM"
"Trusted_Connection"="Yes"

[HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBC.INI\enterprise_setup]
"Driver"="C:\\Windows\\system32\\sqlncli11.dll"
"Server"="(local)"
"LastUser"="SYSTEM"
"Trusted_Connection"="Yes"

[HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBC.INI\LocalBESReportingServer]
"Driver"="C:\\Windows\\system32\\sqlncli11.dll"
"Server"="(local)"
"Database"="BESReporting"
"LastUser"="SYSTEM"
"Trusted_Connection"="Yes"

[HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBC.INI\ODBC Data Sources]
"LocalBESReportingServer"="SQL Server Native Client 11.0"
"bes_bfenterprise"="SQL Server Native Client 11.0"
"enterprise_setup"="SQL Server Native Client 11.0"

[HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\ODBC\ODBC.INI]

[HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\ODBC\ODBC.INI\bes_bfenterprise]
"Driver"="C:\\Windows\\SysWOW64\\sqlncli11.dll"
"Server"="(local)"
"Database"="BFEnterprise"
"LastUser"="SYSTEM"
"Trusted_Connection"="Yes"

[HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\ODBC\ODBC.INI\enterprise_setup]
```

```
"Driver"="C:\\Windows\\SysWOW64\\sqlncli11.dll"
"Server"="(local)"
"LastUser"="SYSTEM"
"Trusted_Connection"="Yes"

[HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\ODBC\ODBC.INI\LocalBESReportingSer
ver]
"Driver"="C:\\Windows\\SysWOW64\\sqlncli11.dll"
"Server"="(local)"
"Database"="BESReporting"
"LastUser"="SYSTEM"
"Trusted_Connection"="Yes"

[HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\ODBC\ODBC.INI\ODBC Data Sources]
"LocalBESReportingServer"="SQL Server Native Client 11.0"
"bes_bfenterprise"="SQL Server Native Client 11.0"
"enterprise_setup"="SQL Server Native Client 11.0"
```

「Trusted_Connection」レジストリー値は「はい」に設定されています。これは、Windows 認証を使用することを意味します。この値が欠落している場合、接続はデフォルトで SQL サーバー認証を使用します。

Linux の ODBC

Linuxでは、ODBC データ・ソースの概念がよりあいまいに定義され、データベース接続設定は DB2 クライアントおよびドライバー (通常はコマンド行を介して) と対話することによって設定されます。

Web レポートの結果数の構成

Web レポートの結果を `Explore Computers` レポートに表示する際のメモリーの過剰使用を防ぐために、`MaxReportResults` キーワードを設定できます。このキーワードにより、Web レポートに表示できる最大行数が設定されます。デフォルト値は `1000000` です。

キーワードの有効な値の範囲は 1 から 4294967295 までです。

以下のメッセージがレポート・ページに表示された場合:

```
Unable to update data table: server aborted or there was an error
 processing your request
```

かつ、以下の例外がログ・ファイルに表示された場合:

```
Too many results returned from computer report. Report execution has been
 aborted
```

この例外は、表示される行数がデフォルト値を超えていることを示しています。この場合、レポートのタイプ、コンピューターのプロパティー、および Web レポートが実行されているシステムのリソースを考慮に入れて、キーワード値を調整します。

以下の手順を実行すると、Web レポートが実行されている Windows システムおよび Linux システムのいずれでも、このキーワードを設定できます。

Windows システムの場合:

regedit を実行してパス `HKEY_LOCAL_MACHINE\Software\Wow6432Node\BigFix\Enterprise Server\BESReports` を見つけます。

ストリング値 (reg_sz) `MaxReportResults` を作成し、それを指定された値に設定します。

```
"MaxReportResults"[REG_SZ] = "1000000"
```

Linux システムの場合:

`MaxReportResults` キーワードを `beswebreports.config` ファイルの `[Software\BigFix \Enterprise Server\BESReports]` セクションに追加し、それを指定された値に設定します。

```
MaxReportResults = 1000000
```

Web レポート用の夏時間 (DST) の管理

Web レポートはデフォルトでは、ローカル・タイム・ゾーンではなく UTC を使用して、スケジュール済みアクティビティーの次の試行時間を管理します。

したがって、ローカル・タイム・ゾーンが変更される (例えば、夏時間から非夏時間に変更される) と、レポートは変更の方向に応じて、1 時間早くまたは 1 時間遅く実行されます。

BigFix バージョン 9.5 以降では、`AdjustScheduleForDST` キーワードを 1 に設定することで、夏時間による変更が原因でスケジュール済みレポートが 1 時間早くまたは 1 時間遅く実行されることを防ぐことができます。

以下の手順を実行すると、Web レポートが実行されている Windows システムおよび Linux システムのいずれでも、このキーワードを設定できます。

Windows システムの場合:

regedit を実行してパス `HKEY_LOCAL_MACHINE\Software\Wow6432Node\BigFix\Enterprise Server\BESReports` を見つけます。

ストリング値 `AdjustScheduleForDST` を作成し、これを 1 に設定します。

```
"AdjustScheduleForDST" = "1"
```

Linux システムの場合:

`AdjustScheduleForDST` キーワードを `beswebreports.config` ファイルの `[Software \BigFix\Enterprise Server\BESReports]` セクションに追加し、これを 1 に設定します。

```
AdjustScheduleForDST = 1
```

重要: 夏時間の設定は、変更した後に最初のイベントがトリガーされるときに有効になります。

BigFix 環境での FIPS 140-2 暗号方式

BigFix は BigFix 暗号モジュールを使用して、その環境全体で暗号機能を実行します。

例えば、オペレーターが BigFix コンソールにログインし、新規ユーザーを作成し、アクションを開始し、新規コンテンツをサブスクライブするたびに、このモジュールによって暗号化操作が実行されます。

BigFix 暗号モジュールは、FIPS (連邦情報処理標準) 140-2 標準に準拠するものとして NIST から認定されているOpenSSL FIPS Object Module 2.0 を使用します。

BigFix サーバーでの FIPS 140-2 の構成

FIPS 140-2 を使用するように BigFix サーバーを構成することができます。

このようにすることで、BigFix 暗号モジュールの状態がエラーである場合、BigFix は始動しないか、実行を停止します。

モジュールの適切なセットアップと初期化を検証するには、以下の手順を実行してクライアント・ログ・ファイルを確認する必要があります。

1. BigFix サーバーで、**「スタート」>「すべてのプログラム」>「BigFix」>「BigFix 管理ツール」**を選択して BigFix 管理ツールを起動します。
2. サイト・ライセンスのロケーションを参照し、**「OK」**をクリックします。
3. **「マストヘッドの管理」**タブを選択します。
4. **「マストヘッドの編集」**をクリックします。
5. **「FIPS 140-2 に準拠した暗号を使用する必要がある」**にチェック・マークを付けて、FIPS 140-2 を有効にします。
6. **「OK」**をクリックします。
7. アクションを実行するための管理者パスワードを入力します。
8. 設定が有効になったことを確認するには、クライアント・ログ・ファイル (ログのデフォルト・パスは) で`C:\Program Files\BigFix Enterprise\BES Client\__BESData\__Global\Logs\YYYYMMDD.log` 以下のタイプのメッセージを確認します。
   - **FIPS 140-2 有効のログ・ファイル・メッセージ**

     ```
     At 14:36:12 -0700 -
     FIPS mode enabled by masthead.
     At 14:36:13 -0700 -
     Cryptographic module initialized successfully in FIPS mode.
     ```

   - **FIPS 140-2 無効のログ・ファイル・メッセージ**

     ```
     At 14:58:28 -0700 -
     FIPS mode disabled by default.
     Unrestricted mode
     ```

クライアントで `_BESClient_Cryptography_FipsMode` 値を設定することで FIPS モードを強制できます。

注: クライアント設定 **_BESClient_Cryptography_FipsMode** は BES クライアントと Web レポート・コンポーネントのマストヘッドで指定した FIPS 設定を上書きします。値を **「なし」** に設定すると、BES クライアントと Web レポート・コンポーネントは FIPS ライブラリーを使用しません。値を **required** に設定すると、FIPS ライブラリーを使用します。

そうすることで、始動時に暗号モジュールでエラーが発生した場合に、クライアントは FIPS モードでは実行されなくなります。

FIPS で検証された暗号ライブラリーのみを使用するように BigFix コンポーネントを強制させるには、以下の手順を実行します。

1. BigFix コンソールを起動します。
2. **「コンピューター」** タブで、リストされている任意の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。
3. **「追加」** をクリックします。
4. 「カスタム設定を追加」ダイアログで、次を入力します。**「設定名」** に `_BESClient_Cryptography_FipsMode` 、 **「設定値」** に `required`
5. **「OK」** をクリックします。
6. **「対象」** タブで `All computers` を選択します。FIPS モードが有効な場合、デジタル署名、暗号化、SHA1/SHA2 ハッシュなどのすべての暗号操作は、FIPS 140-2 レベル 2 の認定を受けた暗号モジュールを使用して実行されます。
7. ダイアログの **「実行」** タブで **「再び関連状態になるときは常にこのアクションを再適用する」** を選択し、**「OK」** をクリックします。

注: BigFix Linux サーバーで FIPS 140-2 を有効にするには、『Linux システムでのマストヘッドの編集 (（ページ） *413)*』で説明されている **`-advRequireFIPScompliantCrypto`** オプションを参照してください。

注:

- FIPS モジュールを有効にする場合、OpenSSL ライブラリーは、独自の自己診断テストを満たすために静的アドレスでロードされる必要があります。
- FIPS モードの始動に関連するエラーで最もよく発生するのは、AIX システムおよび HP-UX システムで暗号モジュールに使用できるシステム・エントロピーが不足している場合のエラーです。
- FIPS モードの設定とメッセージ・レベルの暗号化 (MLE) の設定は互いに独立しています。FIPS は MLE を設定しなくても設定できます。また、その逆も同様です。

メッセージ・レベルの暗号化については、メッセージ・レベルの暗号化 (MLE) の概要 (（ページ） 371)およびメッセージ・レベルの暗号化と DSA (（ページ） 73)を参照してください。

クライアントの暗号化の管理

機密情報への無許可アクセスを防ぐために、クライアントからサーバーおよびリレーへの通信を暗号化できます。これを有効にするには、キーを生成し、`_BESClient_Report_Encryption` という設定のためにクライアントに値を与える必要があります。

デフォルトでは、この設定の値は **オプション** に設定されています。この値はコンソールで設定します。この値については、「BigFixインストール・ガイド」 の「」を参照してください。

Windows サーバーでは、キーは BigFix 管理ツールの**「暗号化」**タブで生成します。

1. **「スタート」>「すべてのプログラム」>「BigFix」>「BigFix 管理ツール」**を選択して BigFix 管理ツールを起動します。
2. **「暗号化」**タブを選択します。ダイアログの上部に、現在の状態を示す記述があります。

クライアントの暗号化には 4 つの状態(「無効」、「保留中」、「有効」、および「ローテーションの保留中」) があります。

無効

この状態は、暗号化証明書が適用マストヘッドに含まれていないことを示します。したがって、クライアントは、レポートを暗号化するように指示されても、暗号化できません。暗号化証明書 (および、受信終了時にレポートを復号化するために使用できる、対応する秘密鍵) を作成するには、**「キーの生成」** をクリックします。状態は、**「保留中」**に設定されます。

処理待ち

この状態では、暗号化証明書が生成されており、適用の準備ができていますが、復号化する必要があるすべてのリレーおよびサーバーに秘密鍵がまだ配布されていません。秘密鍵を手動で配布した場合、**「暗号化の有効化」** ボタンをクリックして、証明書をマストヘッドに組み込み、それをすべてのクライアントに送信します。状態は、「有効」に設定されます。「無効」状態に戻るには **「キャンセル」** をクリックします。

使用可能

この状態では、適用マストヘッド内に暗号化証明書があり、したがって、適用環境内のすべてのクライアントに対して (前に説明した設定を使用して) 暗号化をオンにすることができます。いつでも **「新しいキーの生成」** をクリックすることで、新しい暗号化証明書を作成できます。これは、キー・ローテーション・ポリシーを使用する場合、または暗号化キーがこれまでに漏えいされたことがある場合 (次のセクションを参照) に役立ちます。新しいキーを生成すると、「保留中」状態に戻ります (ただし、次のセクションで説明するように、直ちに適用することを選択した場合は除きます)。また、**「無効」** をクリックして、「無効」状態に戻ることもできます。

ローテーションの保留中 (Pending Rotation)

この状態では、暗号化証明書が適用マストヘッドに組み込まれており、新しい証明書が生成されていて、既存の証明書と置き換える準備ができています。

Linux サーバーでは、スーパーユーザーとして以下のステップを実行することによりクライアントを暗号化できます。

1. 以下を実行してキーを生成します。

```
./BESAdmin.sh -reportencryption -generatekey -privateKeySize=max
 -deploynow=yes
-sitePvkLocation=<path+license.pvk> -sitePvkPassword=<password>
```

2. 以下を実行してキーをアクティブにします。

```
./BESAdmin.sh -reportencryption -enablekey
 -sitePvkLocation=<path+license.pvk>
-sitePvkPassword=<password>
```

使用可能なすべてのオプションをリストするには、以下を実行します。

```
./BESAdmin.sh -reportencryption -h
```

新規暗号化キーの生成

秘密鍵が漏えいしたか、またはキーのローテーションのポリシーを使用している場合、**BigFix管理ツール** から、新規キーを生成できます。

1. **「スタート」>「すべてのプログラム」>「BigFix」>「BigFix 管理ツール」**を選択して BigFix 管理ツールを起動します。
2. **「暗号化」**タブを選択します。
3. **「キーの生成」** ボタンをクリックします。「暗号化認証ファイルの作成」ダイアログが開きます。

4. このダイアログで、キー・サイズを選択します。デフォルトは 2048 です。これは、ほとんどの目的で十分な値です。このキーを直ちに使用するには、ボックスにチェック・マークを付けます。ただし、暗号化を使用するリレーを設定した場合は、新規キーをそれらのリレーに配布できるようになるまで、このボックスをチェック・マークなしのままにしてください。
5. **「OK」** をクリックして、この新規キーをクライアントに配布します。アクションを伝達するには、サイト管理秘密鍵を入力する必要があります。最後のダイアログで、確認を求められます。暗号化キー・サイズおよびサーバー要件について詳しくは、BigFix サポート・サイトの サーバー要件 に関する知識ベースの記事を参照してください。

最上位の復号化を行うリレーの作成

アクションがデプロイされると、短い時間フレーム内に数千のクライアントが (通常はリレーに対して) レポートを返す可能性があります。これらのレポートを暗号化することを選択した場合、リレーはレポートを 1 つにまとめてサーバーに渡します。サーバーは、それを分割して、それぞれのレポートを復号化する必要があります。

何千ものクライアントがある場合、これによりサーバーに対して著しい計算負荷がかかる可能性があります。

パフォーマンスを向上させるために、最上位のリレーが復号化の大部分を実行できるようにすることで、サーバーの負荷を軽減することができます。50,000 を超えるクライアントがある場合、復号化の作業をリレー・チェーンに移すことによって、サーバーの負荷を大

幅に削減できる可能性があります。リレーに独自の復号化キーがある場合、リレーはまずクライアント・メッセージを平文に復号化してから、数千のメッセージを単一のアーカイブにまとめることができます。これを圧縮し、暗号化して、サーバーに渡すことができます。その時点で、サーバーはアーカイブ全体に対して復号化を 1 回だけ実行すればよく、オーバーヘッドが著しく削減されます。

復号化タスクを分散させるには、暗号化キーを最上位リレーに配布します。通常のサーバー・レベルの暗号化の場合、HCL がユーザーに代わって暗号化キーを作成し、それをプログラム・フォルダーに入れます。

Windows システムの場合:

```
%PROGRAM FILES%\BigFix Enterprise\BES Server\Encryption Keys
```

Linux システムの場合:

```
var/opt/BES Server/Encryption Keys
```

負荷を最上位リレーに割り振るには、暗号化キーを、対応するリレー・ディレクトリーに入れます。

Windows システムの場合:

```
%PROGRAM FILES%\BigFix Enterprise\BES Relay\Encryption Keys
```

Linux システムの場合:

```
var/opt/BES Relay/Encryption Keys
```

これらの最上位リレーは、受け取ったすべての文書を復号化し、それらを 1 つにまとめ、単一の署名によって再署名します。フォルダーにキーを必要な数だけ入れることができます。リレーは、暗号化されたクライアント・レポートを取得したときに、それぞれのキーの使用を試みます。クライアントは、マストヘッド・ファイルで検出したキーを使用して暗号化を行います。そのキーは、最後に作成されたキーでなければなりません。ただし、クライアントが、何らかの理由で最新のアクション・サイトを収集しなかった場合、古いバージョンのマストヘッド (すなわち異なる暗号化キー) を使用してレポートを送信する可能性があります。

最上位の暗号化を使用する場合は、以下のベスト・プラクティスについて考慮してください。

- 新しい暗号化キーを作成するたびに、キー・ファイルを手動でサーバーからリレーに転送する必要があります。
- 転送処理中に、秘密鍵ファイルを公開しないようにすることが重要です。これは、インターネットを介してキーを移動してはならないことを意味します。listen している誰かが秘密鍵ファイルのコピーを作成できる可能性があるからです。代わりに、USB キーなどを使用して、あるコンピューターから別のコンピューターへ物理的にキーを転送します。
- 暗号化キーの作成処理中に、秘密鍵ファイルを作成するが、マストヘッドに伝達しないという選択肢があります。このステップによって、クライアントがそのキーを使用して暗号化メッセージの送信を開始する前に、新しいキー・ファイルをリレーに転送する時間の猶予が与えられます。

メッセージ・レベルの暗号化 (MLE) の概要

メッセージ・レベルの暗号化 (MLE) により、クライアントは、RSA 公開鍵と RSA 秘密鍵のペアと AES セッション・キーの組み合わせを使用して、アップストリーム・データを暗号化できます。

RSA キーのペアとして、2048 ビット長または 4096 ビット長のキーを使用できます。キーが長いほどセキュリティーは向上しますが、サーバーで復号化するための処理能力もより多く必要になります。AES セッション・キーでは、FIPS 推奨最大長の 256 ビットが使用されます。クライアント・データをリレーする前に復号化および再パッケージ化することで、サーバーの負荷を軽減するようにリレーを構成できます。

RSA 公開鍵により、セッション・キーが暗号化され、そのキーが AES 暗号化レポートに追加されます。BigFix サーバー (または復号化を実行するリレー) で、対応する RSA 秘密鍵を使用して AES セッション・キーが復号化され、次にそのキーを使用してクライアント・レポートが復号化されます。

レポートの暗号化には 3 つのレベルがあります。

必須

クライアントで、レポートおよびアップロードの暗号化が必須となります。クライアントが暗号化証明書を検出できない場合、またはクライアントの親リレーが暗号化済み文書の受信をサポートしていない場合、クライアントはファイルのレポートもアップロードも行いません。

オプション

クライアントで、レポートおよびアップロードの暗号化が優先されますが、必須ではありません。暗号化を実行できない場合、レポートおよびアップロードは平文で実行されます。

「なし」

暗号化証明書がある場合でも、クライアントは暗号化を行いません。

クライアントでの暗号化の設定方法について詳しくは、「インストール・ガイド」を参照してください。

クライアント・アイコンの変更

デフォルトでは、クライアント UI の左上隅のアイコンは、BigFix のロゴです。

この同じアイコンが、アクションが保留中の場合はトレイに表示され、プログラムが実行中の場合はタスクバーに表示されます。このアイコンは、アクションのソースが誰であるかをユーザーに明確に示すために、また、企業のブランドおよび商標の要件に従うために変更することができます。このアイコンを変更するには、以下の手順を実行します。

- Windows システムの場合:
  1. **「スタート」>「すべてのプログラム」>「BigFix」>「BigFix 管理ツール」**から BigFix 管理ツールを実行します。
  2. **「システム・オプション」** タブをクリックします。
  3. **「アイコンの追加」** をクリックし、 **「開く」** ダイアログを使用して、アイコン (.ico) ファイルを参照します。

Linux システムの場合:

1. 新しいアイコンのパスを調べます (例:`/IEM/newicon.ico`.
2. `/opt/BESServer/bin` コマンド・プロンプトから、以下のようにコマンド・ラインを開始します。

   ```
   ./iem login --server=servername:serverport --user=username
    --password=password
   ```

3. `/opt/BESServer/bin` コマンド・プロンプトから、以下のコマンドを実行します。

   ```
   ./iem post /IEM/newicon.ico admin/icon
   ```

   ここで、*/IEM/newicon.ico* は新しいアイコンの絶対パスを表し、`admin/icon` は新しいアイコンのアップロードに使用するパラメーターです。

アイコンはクライアントに伝達されますが、クライアントが再起動されるまでインターフェースには組み込まれません。その後、クライアント・インターフェースが (アクション、ダッシュボード、または提案に応答して) 開いたときに、指定したグラフィック・アイコンが組み込まれます。

クライアント UI の主要な操作

BigFix クライアント UI を使用すると、クライアント・コンピューターにログインしているエンド・ユーザーにメッセージ・ボックスが表示されます (プレアクション・メッセージ、アクション実行中のメッセージ、再始動/シャットダウンのメッセージを含む)。

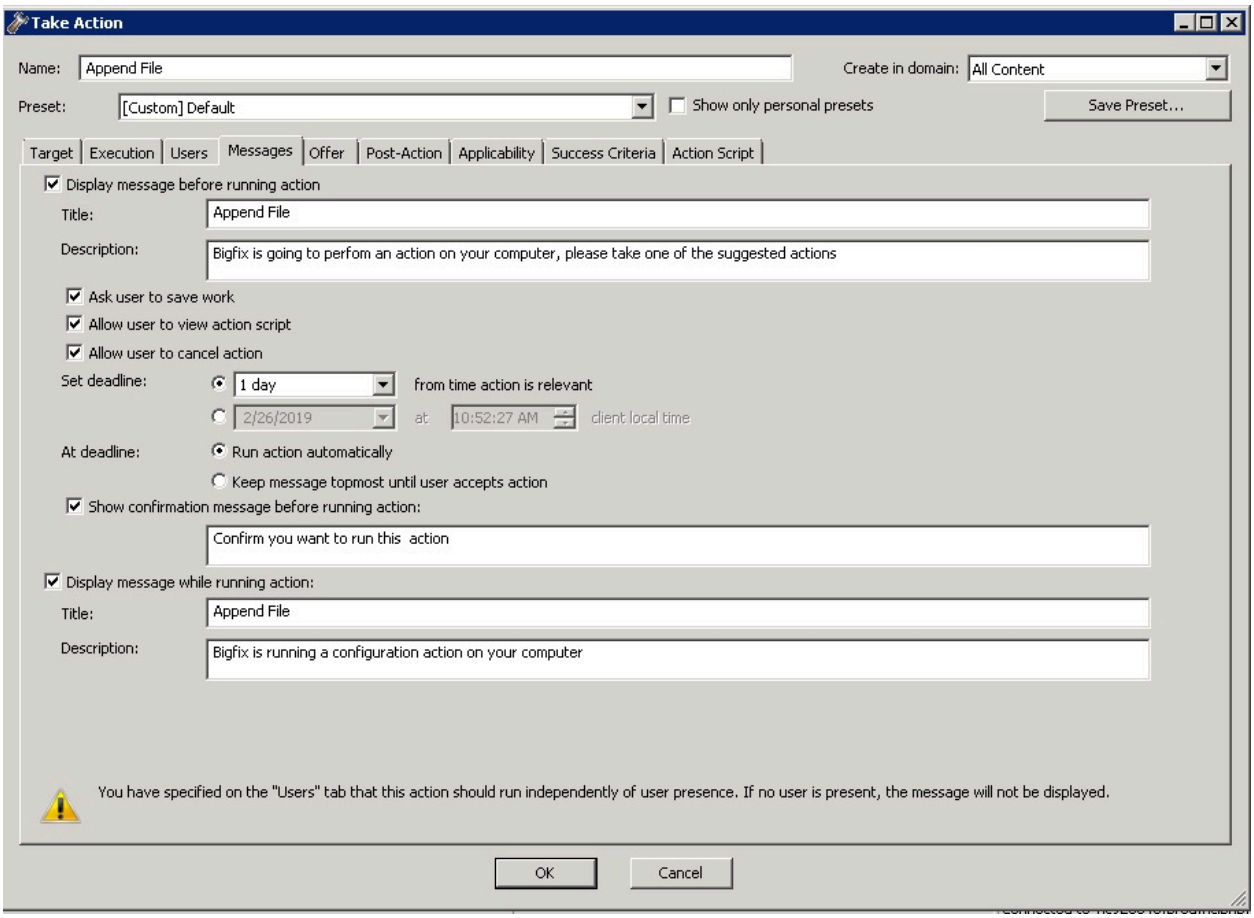
BigFix クライアント UI がユーザー・セッションで実行されていない場合、エンド・ユーザーに BigFix メッセージは表示されません。

BigFix クライアント UI がクライアント通知域 (例えば、Windows のタスクバー) にメッセージを表示します。

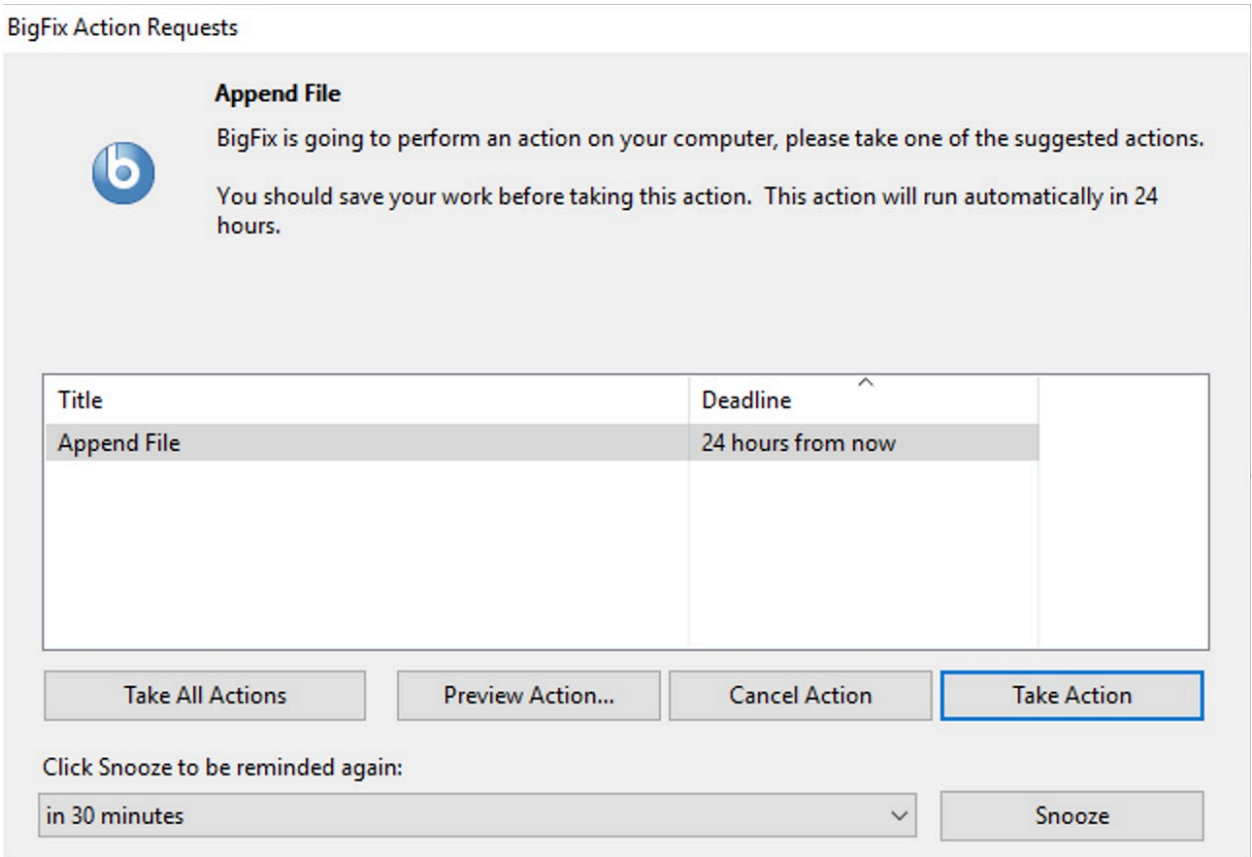
このトピックでは、特に、**「アクションの実行」** パネルの **「メッセージ」** タブおよび **「提案」** タブが BigFix コンソールからトリガーされる場合に、クライアント UI によってクライアント・コンピューター上に表示される事項について説明します。

シナリオ 1 (「メッセージ・タブ」)

BigFix コンソール・オペレーターが、**「アクションの実行」** パネルの **「メッセージ」** タブで以下の設定を指定する場合:



以下のスクリーン・キャプチャーでは、クライアント・コンピューター上のエンド・ユーザー向けの表示事項を示します。



エンド・ユーザーが実行できる主要な操作。

アクションの実行/すべてのアクションを実行

クライアント・コンピューター上でアクションを実行します。

アクションのプレビュー

アクションに含まれるアクション・スクリプトを表示します。

アクションのキャンセル

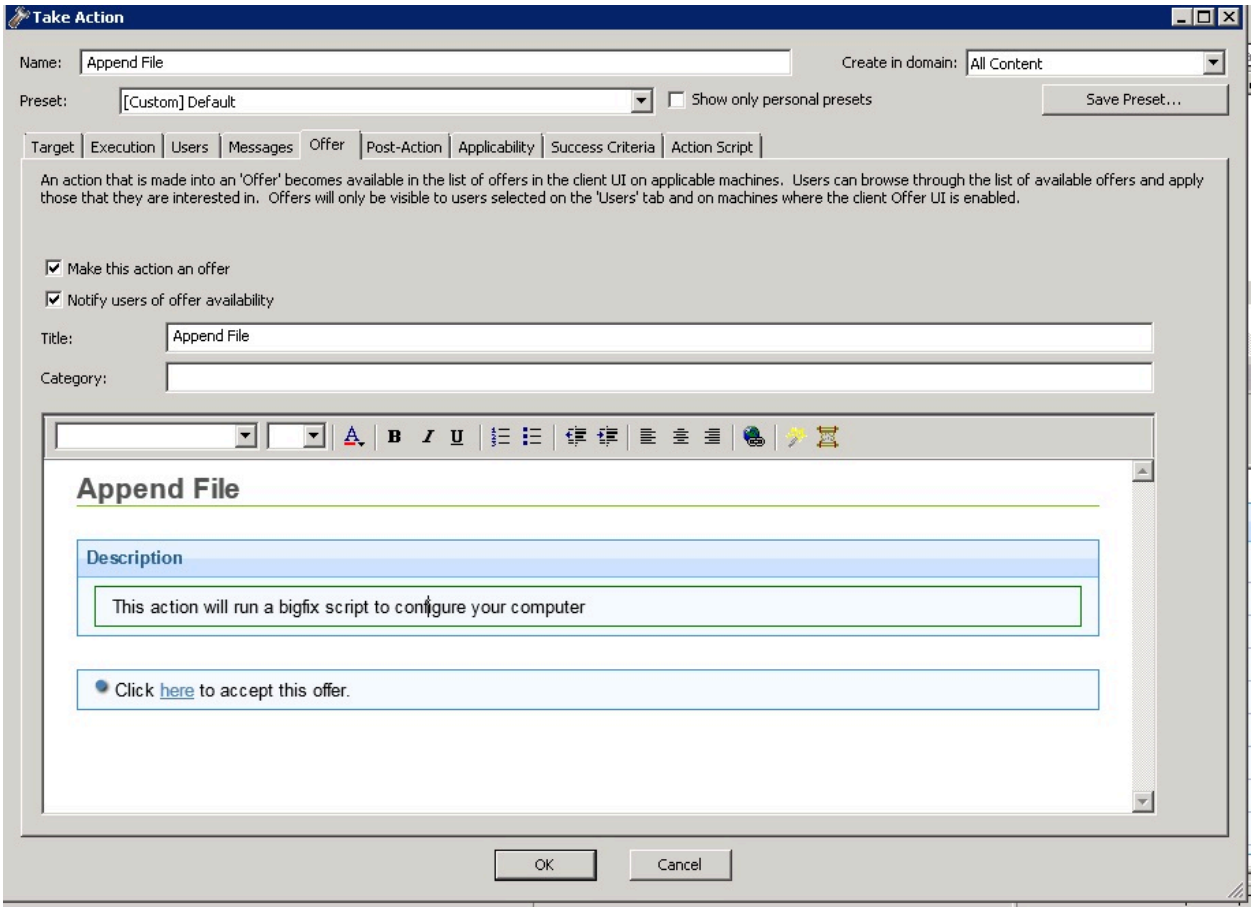
アクションを削除します。

スヌーズ

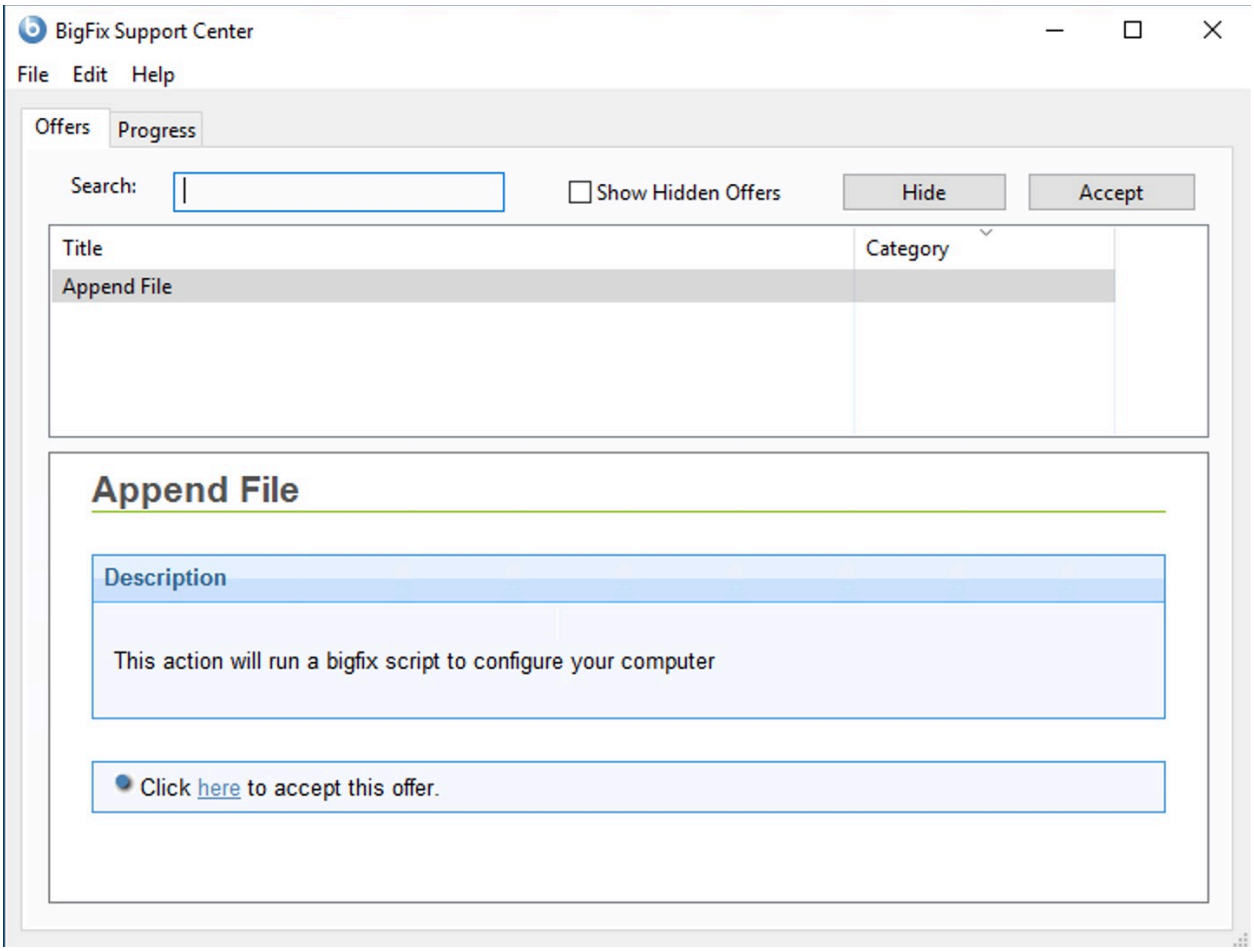
アクションを延期し、アクションの実行に関するリマインド時期を選択できます。

シナリオ 2 (「提案タブ」)

BigFix コンソール・オペレーターが、**「アクションの実行」** パネルの **「提案」** タブで以下の設定を指定する場合:



以下のスクリーン・キャプチャーでは、クライアント・コンピューター上のエンド・ユーザー向けの表示事項を示します。



エンド・ユーザーが実行できる主要な操作。

非表示

クライアント通知パネルを非表示にします。

受け入れる

クライアント・コンピューター上でアクションを実行します。

注: 「進行状況」 タブが前のアクティビティーすべて (以前に受けた提案) の状況を表示しますが、主要な操作は **「提案」** タブで実行できます。

サーバーの最適化

BigFix は効率的に動作し、ネットワーク・リソースには最小限の影響しか及ぼしません。ただし、推奨構成を拡張したインストールでは、割り当てられているサーバーの能力に対してクライアントの数が多すぎる場合があります。

最良の解決策は、環境に対して必要な特性を備えたサーバーを選択することです。ただし、一部の環境設定を変更することで、パフォーマンスを向上させることができる場合があります。これらの最適化のほとんどには、スループットと反応性のトレードオフが伴うため、注意して進める必要があります。HCL ソフトウェア・サポートは、ご使用の特定 デプロイメント に対してどのような変更が最適かについて詳しい情報をご提供いたします。

以下に、使用可能な最適化手法の一部を示します。

- サーバーの負荷を軽減するために、**リレー** を適用する。これは、BigFix のパフォーマンスと反応性を向上させる最も効果的な方法です。通常、リレーが多ければ多いほど、パフォーマンスはより向上します (概して、500 から 1000 のクライアントに対して 1 つのリレーを使用するのが適切な選択です。専用コンピューターを使用すると、パフォーマンスはさらに向上します)。
- **「ファイル」>「環境設定」**で、**クライアントのハートビート**の速度を下げる。これにより、クライアントが取得プロパティーを更新するために定期的にディスパッチするメッセージの頻度が低下します。この頻度を下げると、生成されるネットワーク・トラフィックの量が削減されますが、取得プロパティーの適時性も低下します。ただし、クライアントは、ハートビートの設定に関係なく、サーバーから更新 ping を受信するたびに常に最新情報を送信し、また Fixlet が適用対象であることを検出した場合も常に最新情報を送信します。
- **「ファイル」>「環境設定」**で、**Fixlet の一覧の更新**速度を下げる。これにより、コンソールに表示される情報の更新頻度が低下します。同時に接続する多数のクライアントまたはコンソールがある場合、またはデータベースが非常に大規模な場合、この頻度を下げることで、サーバーの負荷を大幅に軽減できます。複数のコンソール・オペレーターが同時にコンソールを使用する場合、BigFix データベースの負荷を軽減するために、更新速度をデフォルト (15 秒) よりもいくぶん大きい値に設定します。コンソール・オペレーターの数が多い場合、60 秒から 120 秒またはそれ以上の値に変更することを検討します。サーバーで BigFix 管理ツールを使用することで、グローバル最小更新速度を設定できます。

- データベース管理者が以下の最適化を行うことで支援できる場合があります。
  - BFEnterprise データベースの SQL Server リカバリー・モデルを **「単純トランザクション・ロギング (Simple transaction logging)」** に変更する。
  - Web サーバーとオペレーティング・システムでメモリー不足が発生しないように、SQL Server に割り当てられているメモリーのパーセンテージを 100% から 85% に減らす。

その他のパフォーマンスの推奨事項については、BigFix サポート・サイトを参照してください。

コンソールの最適化

応答性能を高めるために、コンソールには適切な CPU の処理能力、メモリー、およびキャッシュ・スペースが必要です。

コンソールのロード時間が長い場合、またはコンソールの実行速度が遅い場合、そのコンソールの速度を向上させる手法がいくつかあります。以下にそれらの方法を示します。

- **十分なメモリーがあることを確認する**。BigFix コンソールは、メモリーの容量が多いと非常に有利で、コンテンツ (Fixlet メッセージ、タスク、アクションなど) の表示、フィルター操作、およびソートが高速化されます。コンピューターに十分な物理メモリーが搭載されていない場合、コンソールの実行速度は非常に低下します。タスク・マネージャー (Ctrl+Shift+Esc) でメモリーの使用量を確認できます。「パフォーマンス」タブを選択し、「物理メモリー」セクションを参照します。使用可能メモリー量が合計メモリー量の 10% 未満の場合、RAM が不足しています。この場合、RAM を追加することでパフォーマンスを向上させることができます。
- コンソールとサーバーの間で **高速ネットワーク接続を使用する** (100 MBPS 以上の LAN 接続が望ましい) 。大規模ネットワークの場合、BigFix データベースのサイズが大きくなる場合があります。この場合、低速接続を使用するコンピューターからコンソールを実行すると、多くの場合、ロード時間は非常に長くなります。
- **リモート制御ソフトウェアを使用する**。ロードして表示するデータが大量にある場合、低速リンクを介してリモート・オフィスのコンソールを操作することは非常に面倒です。このような状況では、Citrix、ターミナル・サービスなどのソリューションや、その他のリモート制御ソフトウェアが便利な場合があります。サーバーに高速ア

クセスできるコンピューターにリモート制御サーバーをセットアップします。そのマシンがコンソールのインスタンスを表示できるようにし、支社がこれらのコンソールをリモートで実行できるようにします。データベースは本社にあり、リモート・オフィスは最適なパフォーマンスを得ることができます。詳しくは、「*BigFix* インストール・ガイド」を参照してください。

- **古いアクションを削除する**。BigFix データベースには、古いアクションに関する情報が保存されます。コンソールは、これらの情報を起動時にロードし、シャットダウン時に保存します。これらの古いアクションを追跡する必要がない場合、削除してかまいません。削除すると、コンソールのロードとクローズが高速になります。削除したアクションは、データベース内にはまだ存在しますが、コンソールや Web レポートにはロードされません。必要に応じて、削除を取り消すことができます。
- パフォーマンスの改善方法について詳しくは、「パフォーマンスの構成」を参照してください。

帯域幅の管理

通常のインストールでは、ファイルのダウンロードで帯域幅の大部分が使用されます。この帯域幅はスロットリングにより制御できます。スロットリングにより、秒あたりのバイト数が制限されます。

サーバーまたはクライアント (あるいはその両方) に対して、帯域幅スロットリングを指定することができます (両方に対して指定した場合、2 つの値のうちの小さい方が使用されます)。以下の状況のような帯域幅の問題がある場合、このスロットリングが重要になります。

- シン・チャネルを使用するリモート・オフィス
- リモート・ダイヤルイン・ユーザーまたは低速接続のユーザー
- 優先順位が高いアプリケーションが使用される共有チャネル
- すでに飽和状態になっている WAN または LAN、あるいは厳しいロード要件がある WAN または LAN

帯域幅スロットリングの設定 (およびその他のリレー、サーバー、およびクライアントの設定) は、サポート・サイトのタスクを使用して実行できます。**「BigFix 管理」** ドメインを

選択し、ナビゲーション・ツリーで **「BES コンポーネント管理」** ノードを選択して、タスク・リスト全体を表示します。

帯域幅スロットリング

多くのネットワーク環境では、特定の地理的位置の間、VPN を使用しているオフィス・ユーザーとホーム・ユーザーの間などで、帯域幅が制限されています。大規模な Microsoft® パッチ (例えば、Windows 7 SP1 更新プログラムは 537 MB です) をデプロイすると、帯域幅が制限された接続が圧迫され、特定のユーザーまたはアプリケーションに帯域幅の問題を引き起こすことがあります。

帯域幅の使用量に関連する問題を回避するために、BigFix では次のメカニズムを使用しています。

- 適切に実装および保守されたリレー・アーキテクチャー
- 一定期間にわたるパッチ・デプロイメントの分散
- コンポーネント間での帯域幅スロットリング構成の適用

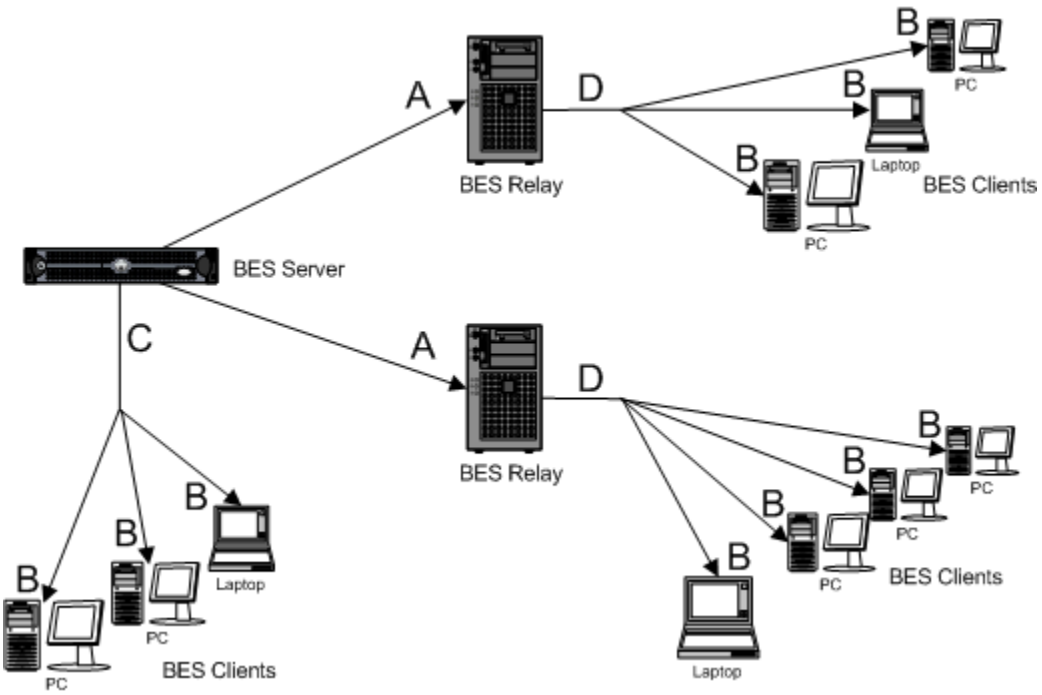
このセクションでは、帯域幅スロットリングを使用するように BigFix コンポーネントを構成する方法について説明します。BigFix コンソール・オペレーターは、クライアント構成設定によって、ネットワーク接続を介してファイルを送信する際に使用される最大バイト数/秒を制御でき、これは BigFix サーバー・コンポーネント、リレー・コンポーネント、クライアント・コンポーネントの各レベルで行うことができます。

帯域幅スロットリングの設定は、BES サポート・サイト内のタスクを使用して構成できます。コンソールで **「すべてのコンテンツ」 > 「Fixets とタスク」 > 「タスクのみ」 > 「サイト別」 > 「BES サポート」** の順にクリックして、スロットルという用語を検索します。設定はすべて、レジストリーのクライアント・セクションのエンドポイント上のレジストリー・キー・クライアント設定として設定されます (`HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\BigFix\EnterpriseClient\Settings\Client`、名前付きの値を持ち、種類は `String [REG_SZ]`)。値は数値として入力されます。例えば、500 bps は 500 と入力されます。

関連する構成設定について詳しくは、帯域幅スロットリング (（ページ）)を参照してください。

スロットリング方式

次の図は、特定セグメント・タイプの帯域幅スロットリングを構成するための設定を決定するのに役立ちます。



リレーを使用するスロットリング (A)

BigFix サーバーからダウンロードするリレー

BigFix リレーは、BigFix サーバーからダウンロードする際に、ファイル・ダウンロードをスロットリングするように構成できます。BigFix リレー・スロットリングが有効になっている場合、リレーは指定されたバイト数/秒以下で BigFix サーバーからダウンロードします。この設定は、サーバーへの接続が低速なリレー (リモート・ロケーションのリレーなど) には特に有用です。

クライアント・スロットリングは、リレーで **_BESGather_Download_LimitBytesPerSecond** 設定を使用して構成できます。この設定の構成方法について詳しくは、帯域幅スロットリング (（ページ）　　)を参照してください。

BigFix リレーの送信ダウンロード・トラフィック合計

BigFix リレーは、累積的ファイル・ダウンロードを常にスロットリングするように構成できます。このスロットリング設定が有効になっている場合、リレーはすべてのファイル・

ダウンロードについて、指定されたバイト数/秒以下で送信します (クライアントおよび子リレーを含む)。この設定は、ローカル・エリア・ネットワークで、多くのクライアントに同時にパッチが送信される際に、非常に多くの帯域幅が使用されることが懸念される場合に特に有用です。

リレーの累積的ダウンロードのスロットリングは、リレーで **_BESRelay_HTTPServer_ThrottleKBPS** 設定を使用して構成できます。この数値を、リレーが結合されたすべてのクライアントに提供する秒当たりのキロバイト数合計に設定します。

BigFix クライアントを使用するスロットリング (B)

サーバーまたはリレーからファイルをダウンロードするクライアント

BigFix クライアントは、BigFix サーバーまたは BigFix リレーからダウンロードする際に、ファイル・ダウンロードをスロットリングするように構成できます。BigFix クライアント・スロットリングが有効になっている場合、クライアントは指定されたバイト数/秒以下でサーバーまたはリレーからダウンロードします。この設定は、低速で接続している個々のコンピューターには特に有用です (移動中の営業担当員やダイアルアップを使用するホーム・ユーザーなど)。

クライアント・スロットリングは、クライアントで **_BESClient_Download_LimitBytesPerSecond** 設定を使用して構成できます。この設定の構成方法について詳しくは、帯域幅スロットリング (（ページ）)を参照してください。

BigFix サーバー/リレーを使用するスロットリング (C および D)

サーバー/リレーの送信ダウンロード・トラフィック合計

サーバーは、累積的ファイル・ダウンロードを常にスロットリングするように構成できます。このスロットリング設定が有効になっている場合、サーバー/リレーはすべてのファイル・ダウンロードについて、指定されたバイト数/秒以下で送信します (クライアントおよび子リレーを含む)。この設定は、ローカル・エリア・ネットワークで、多くのクライアントに同時にパッチが送信される際に、非常に多くの帯域幅が使用されることが懸念される場合に特に有用です。

BigFix サーバーの累積的ダウンロードのスロットリングは、サーバーで **_BESRelay_HTTPServer_ThrottleKBPS** 設定を使用して構成できます。この設定の構成方法について詳しくは、帯域幅スロットリング (（ページ）　　)を参照してください。

注: 構成設定への変更を有効にするには、該当する BES ルート・サーバーまたは BES リレー・サービスを再始動する必要があります。

スロットリング・オプション

BigFix には、異なるコンポーネントが相互にトラフィックをスロットリングする方法を制御する設定が多数あります。スロットリングは常に特定のコンポーネントに対して行われ、中にはスロットリングを使用できないコンポーネントもあります。

スロットリングされるコンポーネント

スロットリングされるコンポーネントとは、リレー、サーバー、Web レポートの `wwwrootbes` ディレクトリーから直接ダウンロードされるファイルです。アクションのためにダウンロードされるファイル (パッチ、サービス・パックなど) はこの方法でダウンロードされます。Fixlet サイトのコンテンツも、収集インタラクション中にこの方法でダウンロードされます (ただし、サイト・ディレクトリー・リストは別個にダウンロードされます)。

スロットリングされないコンポーネント

ただし、クライアント登録には多数の「スロットリングに似た」機能があります。生成されるダウンストリーム UDP トラフィックの量をリレー/サーバー側でレート制限したり、クライアント登録の頻度をクライアント側で減らしたり、リレー選択中に生成される ICMP トラフィックの量をクライアント側でレート制限することができます。クライアント登録は帯域幅を多く消費するわけではありませんが、負荷が高くなる可能性があります。ダウンロード要求プラグイン、ステータス・レポート・プラグイン、「Web レポート」プラグインなどです。クライアントにとって、このトラフィックはごくわずかであると予想されます。しかし、ダウンロード・ステータス・レポートのようなものは、コンソールとルート・サーバーの間に多数のトラフィックを発生させ、こうしたトラフィックはスロットリングできません。クライアントがリレーに「サイト X の最新のコンテンツを教えてください」 (「GatherActionMV コマンドを受信しました。バージョンの差異を収集しています」

と記録されます) と尋ねた場合、この対話はスロットリングされません。リレーの応答は一般に小規模です (0 ～ 40k)。どうしても必要な場合は、クライアントで収集間隔をあけて、この情報の取得頻度を減らすことができますが、このトラフィックはたいていはごくわずかです。

アップストリームへ向かうトラフィックの量を制御するために、「PostResults」インターフェースを使用して実行できることもあります。最も基本的な「スロットリング」メカニズムは、クライアント上で単に最小レポート間隔を大きくするか、ハートビート間隔を長くすることです。「ResultSizeLimit」および「ResultTimeLimit」リレー設定の組み合わせを使用して、リレーが送信できるトラフィックの量に対して大まかな制限を設定することもできます。ただし、これを行う前にカスタマー・サポートに確認する必要があります。期待する動作が得られる可能性はほとんどありません。結果の送信にはクライアントの登録よりも多くの帯域幅が必要になりますが、それでもコンポーネントのダウンロードやアップロードで使用される帯域幅の量よりはずっと少量です。ほとんどのデプロイメントでは、トラフィックの量はごくわずかです。

スロットリングの最小転送レート

スロットリングされた通信の間、BigFix クライアントはデータのチャンクを送信した後、必要以上に待機してから、次のチャンクを送信します。BigFix クライアントは、チャンクごとにデータ量と待機時間を変えることができます。チャンク当たりのデータ量を小さくして、チャンク間のデータ量を最大化することで、スロットリングの最小転送レートが設定されます。

実際のスロットリング作業は BigFix リレー・コンポーネントが行うため、BigFix をアップグレードして最小スロットリング・レートを変更する必要がある場合は、BigFix リレーと BigFix クライアントを新しいバージョンにアップグレードすることが重要です。

有効な最小スロットリング・レートは 100 バイト/秒です。

スロットリング設定はすべて BPS (バイト/秒) または KBPS で指定されます。上記の有効な最小値が指定されているため、スロットリング・レートを有効な最小値よりも低く設定しても、その値にはならず、有効な最小値になるだけです。また、有効な最小値は 1 KBPS 未満ですが、KBPS スロットリング設定に設定できる最小値は 1 のみであるため、有効な最小値には到達できません。すなわち、有効な最小レートは、設定によって制御される場合もあれば、クライアントのスロットリング制限によって制御される場合もあります。

チャンク・サイズの指定に使用される設定は、次のとおりです。

_BESClient_UploadManager_ChunkSize

値はバイトで指定されます (デフォルト: 131072。これは 128 KB に相当します)。

「0」は、アップロードが 1 つのチャンクで行われることを示します。

このコンピューターとアップストリーム・コンピューターとの間で値の不一致が生じた場合、チャンクのサイズは 2 つのうち小さい方の値に設定されます。

_BESRelay_UploadManager_ChunkSize

値はバイトで指定されます (デフォルト: 131072。これは 128 KB に相当します)。

「0」は、アップロードが 1 つのチャンクで行われることを示します。

このコンピューターとアップストリーム・コンピューターとの間で値の不一致が生じた場合、チャンクのサイズは 2 つのうち小さい方の値に設定されます。

アップロードのスロットリング

「アップグレード・マネージャー」によって生成されるアップロードは、クライアント側またはリレー/サーバー側からスロットリングできます。このコンポーネントでは、静的スロットリングのみをサポートします。着信接続に対して全体的なスロットリングを設定する「PostFile」設定と、発信接続に対してスロットリングを設定する「UploadManager」設定があります (1 度に 1 つの発信接続しかありません)。両方とも設定されている場合、子は 2 つの値のうち小さい方を使用する必要があります。

_BESRelay_PostFile_ThrottleKBPS

「0」は「制限なし」を意味します (デフォルト: 0)

この設定は、PostFile スロットリングには不十分で、この設定を機能させるには、クライアントで _BESClient_UploadManager_ThrottleKBPS を 0 以外の値に設定する必要があります。制限は 2 つの値のうち小さい方になります。

アップロード・インタラクションの開始時に、PostFilePlugIn はこの数値をリレーで現在進行中のアップロード数合計で割って、結果を子に送信します。子は、結果である制限を順守する必要があります。

サーバーとリレーで同じ設定

_BESRelay_UploadManager_ThrottleKBPS

デフォルト: 0

リレーにのみ関係があります (サーバーにはアップロード先がありません。ただし、DSA はある時点でこれを変更する可能性があります)

リレーがファイルを親にアップロードする際、リレーは自身をこのレートに制限します。インタラクションをチャンクに分割し、チャンクごとに接続を行って、間に待機時間を入れることで、制限を設定します (これは、インタラクションの間ずっと 1 つの接続を保持するダウンロードのスロットリングとは対照的です)。

「0」は「制限なし」を意味します

_BESClient_UploadManager_ThrottleKBPS

デフォルト: 0

クライアントにのみ関係があります

クライアントがファイルを親にアップロードする際、クライアントは自身をこのレートに制限します。インタラクションをチャンクに分割し、チャンクごとに接続を行って、間に待機時間を入れることで、制限を設定します (これは、インタラクションの間ずっと 1 つの接続を保持するダウンロードのスロットリングとは対照的です)。

「0」は「制限なし」を意味します

「Upload Manager」を使用してエンドポイントからアップロードされるファイルです。

ダウンロードのスロットリング設定の 2 つの主軸

「サーバー側」または「クライアント側」

サーバー側のスロットリング (サーバー/リレー/Web レポート) は、接続しているすべての子の間で共有される帯域幅の量として表されます。

クライアント側のスロットリング (クライアント/リレー) は、1 つのアップストリーム接続で使用される帯域幅の量として表されます。クライアントが複数の同時アップストリーム接続を使用している場合、クライアントはもっと多くの帯域幅を使用する可能性があります。

静的または動的

サーバー側およびクライアント側のスロットリングが有効な場合、BigFix コンポーネントは計算された帯域幅制限のうち低い方の値を使用します。動的および静的スロットリングの両方が有効な場合、静的スロットリング設定の代わりに動的スロットリング設定が使用されます。

サーバー側の静的スロットリングは、BigFix 以外のコンポーネント (Web ブラウザーなど) に影響を与える可能性のある唯一のスロットリング・タイプです。

静的スロットリング

サーバー側およびクライアント側の静的スロットリング設定について説明します。

サーバー側

サーバー側の静的スロットリング設定では、サーバーが静的スロットリングを使用してクライアントに送信するダウンロード・トラフィックの合計量を制御します。任意の書込み接続に割り当てられる帯域幅の量は、単純に「ThrottleKBPS」設定をアクティブな書込み接続の数で割った値です。プラグイン接続は、「書込み」接続としてカウントされないことに注意してください。ただし、静的または動的スロットリングが有効化されたファイル・ダウンロードは「書込み」接続としてカウントされます。以下の場合を考えます。

- ThrottleKBPS = 500
- 1 つのクライアントが動的スロットリングなしで接続されています
- 1 つのクライアントが動的スロットリングありで接続されています

この場合、動的スロットリングなしのクライアントには、帯域幅 250 KBPS が割り当てられます。動的スロットリングありのクライアントの帯域幅の使用量は、動的スロットリング・アルゴリズムによって決定されます。250 KBPS をはるかに下回るまたは上回る可能性があるため、サーバーの帯域幅の使用量合計は 500 KBPS であるとは限りません。

注: サーバー側の設定は KBPS 単位です。リレーおよびルート・サーバーの場合:

- _BESRelay_HTTPServer_ThrottleKBPS
  - デフォルト: 0
  - 「0」は「制限なし」を意味します
- Web レポートの場合:

    - _WebReports_HTTPServer_ThrottleKBPS
    - デフォルト: 0
    - 「0」は「制限なし」を意味します

クライアント側

クライアント側の静的スロットリングは、次のような単純な設定です。「クライアント」(クライアントまたはリレー) は親に「この速度でファイルを送信してください」と伝え、親はそれに応じます。設定は BPS 単位です。BES クライアントの場合:

- _BESClient_Download_LimitBytesPerSecond
    - デフォルト: 0
    - 「0」は「制限なし」を意味します

親からファイルをダウンロードする BES リレーの場合:

- _BESGather_Download_LimitBytesPerSecond
    - デフォルト: 0
    - 「0」は「制限なし」を意味します

スロットリング・グループ

「スロットリング・グループ」は静的スロットリング機能の一部で、スロットリング・グループを使用すると、一連のクライアントは個々にではなくグループとして自身をスロットリングできます (または、サーバー側スロットリングを使用する他の接続とともに)。クライアントが自身を「スロットリング・グループ」の一部であると識別すると、所属するグループの名前を、グループ全体に必要な速度とともに送信します。このため、クライアントは「私は「リモート」グループに属しているので、全体として 10000 BPS が必要です」などと伝えます。サーバーがそのクライアントにデータを送信する際、クライアント・グループの合計接続数に基づいてスロットリングします。従って、「リモート」グループに 5 つのアクティブな接続がある場合、クライアントは 2000 BPS を取得します。異なるクライアントが異なる「limit バイト/秒」値を送信する可能性があるため、別のクライアントが「私は「リモート」グループに属しているので、全体として 5000 BPS が必要です」と伝えて 1000 BPS を提供され、同時に最初のクライアントが 2000 BPS を提供されることもあります。特別なグループ「ipaddress」の場合、サーバーはこの接続を同じ

IP アドレスの他の接続とともにグループ化します。これは、リレー・アップストリーム・トラフィックのデフォルトです。クライアントはグループ "" にデフォルト設定されているため、クライアントの「LimitBytesPerSecond」設定は現在アクティブなすべてのファイル・ダウンロードで共有されます。

- _BESGather_Download_ThrottleGroup (Windows Server でのみ有効)
  - デフォルト: "ipaddress"
  - 親は、このリレーがここで指定されたグループの一部であると見なします
- _BESClient_Download_ThrottleGroup
  - デフォルト: 文字列としてのコンピューター ID
  - 親は、このクライアントがここで指定されたグループの一部であると見なします
  - これは文字列値です。旧バージョンの ClientSettings ドキュメントでは、これが数値であると間違って主張されていました。

動的スロットリングはスロットリング・グループの影響を受けません (このことの興味深い副作用として、2 つのファイルを同時にダウンロードする場合、使用可能な帯域幅の 20% をターゲットに設定したクライアントは 40% を使用することになります)。

動的スロットリング

重要: 動的帯域幅スロットリングの実装は推奨されません。静的な帯域幅スロットリングの構成を強くお勧めします。詳しくは、『静的スロットリング (（ページ） 388)』を参照してください。これらの帯域幅の計算は信頼性も低く予測できないため (低帯域幅ネットワーク環境では特にその傾向が高い) 、動的帯域幅スロットリングをデプロイメントで引き続き使用する場合は、デプロイメントおよびネットワーク内の動的帯域幅スロットリングの実装とテストについて HCL サポート・チームに連絡してください。

大規模なダウンロードが使用可能になった時、適用環境内の各リンクに、帯域幅に関する固有の問題が生じることがあります。

サーバーからクライアントのリンク、サーバーからリレーへのリンク、およびリレーからクライアントへのリンクについて検討する必要があり、それぞれに個別の調整が必要な場合があります。別のセクションで説明したように、単純にデータ・レートの最大値を設定 (スロットリング) することが可能であり、そのために、順守可能な幅広いポリシーが用意

されています。例えば、BigFix クライアントでリレーからのホップ数が 3 を超えている場合は、クライアントを 2 KB/秒にスロットリングすることが考えられます。ただし、最適なデータ転送速度は、現在の階層およびネットワーク環境に応じて、大幅に変わる可能性があります。

より適切な手法は、ネットワーク容量全体を監視して分析する動的な帯域幅スロットリングを使用することです。標準的なスロットリングでは最大データ・レートを指定するだけですが、動的なスロットリングでは「ビジー時間」の比率が加わります。これは、帯域幅のうち、ネットワークがビジーであるときに割り振る部分の割合です。例えば、ネットワーク・トラフィックの存在が検出された場合に、ダウンロードに使用される帯域幅を使用可能な帯域幅の 10% 以下に抑えるように指定できます。動的なスロットリングでは、ビジー率が低すぎて実際的でない場合のために、最小データ・レートも指定します。

任意のリンクに対して動的スロットリングを有効にすると、現在のデータ・スループットが監視され、分析されて、適切なデータ・レートが設定されます。競合するトラフィックが存在しない場合は、スループットが最大レートに設定されます。トラフィックが存在する場合は、指定したパーセンテージまたは最小レートのどちらか高い方までデータ・レートがスロットリングされます。動的スロットリングを正しく機能させるには、サーバー側とクライアント側の両方で動的スロットリングを有効にする必要があります。

動的な帯域幅スロットリングは、コンピューター設定で制御します。リンクごとに、以下の 4 つの基本的な設定があります。

DynamicThrottleEnabled

この設定のデフォルトはゼロ (無効) です。その他の任意の値の場合、指定されたリンクの動的スロットリングが有効になります。

DynamicThrottleMax

この設定は、通常、デフォルトでは最大の符号なし整数値になります。これはフルスロットルを示します。リンクによっては、この値に、最大データ転送速度 (1 秒あたりのビット数または K ビット数) が設定されます。

DynamicThrottleMin

この設定のデフォルトはゼロです。リンクに応じて、この値に、最小データ転送速度 (1 秒あたりのビット数または K ビット数) が設定されます。この値は、以下に示すパーセンテージ・レートの下限を設定します。

DynamicThrottlePercentage

この設定は、デフォルトでは 100% です。100% は、通常の (動的ではない) スロットリングと同じ効果があります。これは、最大帯域幅のうち、ネットワークがビジーであるときに使用する部分の割合です。通常は、5% から 10% の値が使用され、既存のネットワーク・トラフィックよりも優先されることがないようにします。(この設定にゼロを使用すると、100% と同じことになります。)

動的帯域幅設定は、他の設定と同じように作成または編集することができます。つまり、任意のコンピューター・リストで項目 (または項目のグループ) を右クリックし、コンテキスト・メニューから**「コンピューターの設定を編集」**を選択します。

具体的な変数名には以下のものがあります。

BigFix のサーバー設定とリレー設定:

```
_BESRelay_HTTPServer_DynamicThrottleEnabled
_BESRelay_HTTPServer_DynamicThrottleMaxKBPS
_BESRelay_HTTPServer_DynamicThrottleMinKBPS
_BESRelay_HTTPServer_DynamicThrottlePercentage
```

BigFix クライアントの設定:

```
_BESClient_Download_DynamicThrottleEnabled
_BESClient_Download_DynamicThrottleMaxBytesPerSecond
_BESClient_Download_DynamicThrottleMinBytesPerSecond
_BESClient_Download_DynamicThrottlePercentage
```

一括設定:

```
_BESGather_Download_DynamicThrottleEnabled
_BESGather_Download_DynamicThrottleMaxBytesPerSecond
```

```
_BESGather_Download_DynamicThrottleMinBytesPerSecond
_BESGather_Download_DynamicThrottlePercentage
```

注: 上記の設定を有効にするには、影響を受けるサービス (サーバー、リレー、またはクライアント) を再起動する必要があります。

サーバーとそれに接続されたクライアントに、異なる最大値と最小値を設定した場合、接続において、2 つの値のうち小さい値が選択されます。

ダウンロードの管理

BigFix では、ダウンロードを効率的に行うため、および使用可能な帯域幅を最大限活用するために、複数の手法が使用されます。その他の手法のうち、キャッシングがサーバー、リレー、およびクライアントを含むすべての BigFix 要素で幅広く使用されます。

クライアント上のアクションによって download ファイル・コマンドが実行されると、クライアントのローカル・キャッシュで最初にファイルの存在がチェックされます。クライアントは、ローカルでファイルを検出できない場合、その親 (通常はリレー) にファイルを要求します。次に、リレーがリレー自体のキャッシュをチェックします。リレーは、ファイルを検出すると、要求したクライアントにそのファイルを直ちに送信します。検出できなかった場合、その親に要求を渡します。親は別のリレーである場合があり、このプロセスがこの後も続けられます。最終的に、サーバーが内部サーバーまたはインターネットからファイルを取得し、それをキャッシュし、逆の経路でそのファイルを渡します。経路上にある各リレーは、ファイルを受信した後、それをキャッシュし、ファイルの転送を続行し、元のクライアントに届けます。元のクライアントもそのファイルをキャッシュします。

アクションの実行中にエージェントが download now コマンドを実行すると、アクション・スクリプトに指定された URL からファイルが要求されて収集されます。

各キャッシュでは、スペース不足になるまで、ファイルが保持されます。スペース不足になった時点で、スペースを確保するために、最も長い期間使用されていない (LRU) ファイルがキャッシュから消去されます。BES サポート・サイトから使用できる **「分析 ID# 227 BES リレーのキャッシュ情報 (Analysis ID# 227 BES Relay Cache Information)」** をアクティブにすることで、リレーのキャッシュ・サイズなどのリレー情報を表示できます。デ

フォルトのキャッシュ・サイズは 1 GB ですが、BES サポート・サイトで**「タスク ID# 148 BES リレー/サーバー設定: ダウンロード・キャッシュ・サイズ (Task ID# 148 BES Relay/Server Setting: Download Cache Size)」**を使用してこのサイズを変更することができます。

ファイルを手動でダウンロードし、キャッシュする必要がある場合があります。これは主にファイルが公開されていない場合です。この場合、ファイルをソースから直接ダウンロードする必要があります。具体的な手動キャッシュ要件について詳しくは、**「Fixlet の説明 (Fixlet Description)」** タブを参照してください。ファイルをダウンロード・キャッシュの場所 `__Download` にコピーすることで、ダウンロード・キャッシュを事前に作成できます。これらのファイルを手動で削除することもできます。

キャッシュは、プログラム・フォルダーのサブフォルダーとして格納されます。プログラム・フォルダーは、デフォルトでは `%PROGRAM FILES%\BigFix Enterprise` (Windows システムの場合)、および `/var/opt/BES Server` (Linux システムの場合) に作成されます。サーバーのダウンロード・キャッシュは `BES Server\wwwrootbes\bfmirror\downloads\sha1` で、クライアントのダウンロード・キャッシュは `BES Client\__BESData\__Global\__Cache\Downloads` にあります。

ダウンロード・キャッシュに加えて、リレーは、各アクションに必要なすべてのファイルを格納するアクション・キャッシュ (これも 1 GB) を保持し、クライアントは、ユーティリティー・キャッシュを保持します。

帯域幅やダウンロードなど、リレーのトラブルシューティングについて詳しくは、「Relay Health」を参照してください。

クライアントは、以下のいずれかの方法でアクション・スクリプトにリストされた URL からファイルを要求してそのファイルを収集します。

- アクション・スクリプトを解析してダウンロード一式を計算できる場合は、サーバーによってダウンロード一式が計算されます。特定のアクションに対してプリフェッチ・ダウンロードを入手できる場合は、エージェントが単一の要求でリレーを要求できます。この要求では、エージェントがアクション ID を送信し、サーバー応答によって全ファイルが入手可能かそうでないかが示されます。ファイルがすべて入手可能である場合は、エージェントがファイルをその序数 (1 はスクリプト内の 1 番目のファイルを示し、2 はスクリプト内の 2 番目のファイルを示します) で要求する処理

を開始します。ファイルが入手不可能である場合は、リレーがそのことをエージェントに通知し、ファイルをフェッチするプロセスを開始します。エージェントは、ダウンロードが入手可能になるのを待つことを通知し、そのアクションに対して 10 分間ダウンロード待ちの状態となります。10 分が経過してその特定のアクションに対してダウンロードが入手可能になった場合は、エージェントが再びリレーに要求します。

リレー上でアクションに対してダウンロードが入手可能になると、リレーの子に通知が送信され、その通知を使用してダウンロードの要求が加速されます。何らかの理由で通知メッセージがブロックされた場合は、エージェントによる 10 分間の「リレーへの再要求」動作によって最終的にダウンロードが入手可能であることが検出され、その収集が開始されます。アクション ID に基づいたダウンロード、および序数が入手可能になると、子リレーも親から通知を受けます。子リレーは、この通知を使用して再びダウンロードの要求を加速します。

- アクション・スクリプトにリストされているダウンロードの URL、サイズ、およびハッシュ値をエージェントだけが計算できるようになっているダウンロードの場合は、エージェントが、入手可能なダウンロードを項目化した要求を使用して親リレーに照会を行います。この要求には、特定のエージェントが必要とするダウンロード項目のリストが含まれています。リレーとクライアントは、上述したとおりに動作して、後続の要求を遅らせて通知を待ちます。

ダウンロードの再開

接続の問題が原因でダウンロードが失敗した場合、ダウンロード・プロセスは次のように再開されます。

- クライアントが BigFix のリレーまたはサーバーからダウンロードしている場合、10,000 バイトのチャンクでダウンロードを再開できます。つまり、クライアント・プロセスは、再開されると、既に受信している 10,000 バイトのブロックを検査して、検査した最後のブロックの後からダウンロードを再開します。
- クライアントが別のサーバーの URL からの直接ダウンロードを実行している場合、クライアント・プロセスの再開時に、ダウンロードは最初から開始されます。

インターネット・サイトからの直接ダウンロード

パッチ 1 以降の Download Direct の既存のクライアント設定に加えて、特定のリソースをそれらが配置されているサイトから直接ダウンロードするようにクライアントを構成して、VPN ベースのクライアントにサービスを提供するリレーのネットワークへの影響と帯域幅の要件を緩和することができます。

特定のドメイン・セットへのすべてのリソース要求を、リレーからではなくインターネットから直接ダウンロードする必要があることを指定できます。**_BESClient_Download_Direct_Domainlist** という名前のクライアント設定を使用して、直接ダウンロードが必要なドメインのリストを指定します。

注: PeerNest が有効になっている場合、動作は変わりません。リソースは引き続きピア URL から要求されます。

注: この設定は、**_BESClient_Download_Direct** の代わりに使用する必要があります。実際には、**_BESClient_Download_Direct** 設定では、指定されたドメインに関係なく、すべてのリソースがインターネットから直接ダウンロードされます。

インターネットからの直接ダウンロードが失敗した場合、クライアントが BigFix リレーまたはサーバーからファイルをダウンロードしようとするように指定できます。**_BESClient_Download_DirectRecovery** という名前のクライアント設定を使用して、この動作を有効にします。

注: この設定は、クライアント設定 **_BESClient_Download_Direct** または **_BESClient_Download_Direct_Domainlist** が有効になっている場合にのみ有効です。

これらの設定について詳しくは、ダウンロード (（ページ）　　)を参照してください。

ネットワークに基づく直接ダウンロードの有効化

パッチ 7 から、特定のサブネットに接続されている BigFix クライアントにのみ直接ダウンロードを許可できるようになりました。

直接ダウンロードを許可するサブネットのリストは、新しい設定 **_BESClient_Download_Direct_SubnetList** で指定できます。この設定では、CIDR 表記形式で指定されたサブネットのみを受け入れます。例: `192.1.77.0/25;192.1.0.0/16`。

複数のネットワーク・インターフェースを持つコンピューターの場合、許可されるリストを確認する際に考慮されるサブネットは、BigFix リレーに接続されている IP アドレスのサブネットです。

直接ダウンロードが進行中の場合、クライアントがリスト内のどのサブネットにも属さない新しい IP アドレスを使用して再登録すると、クライアントは進行中のダウンロードを中断します。

クライアントが進行中の直接ダウンロードを中断すると、以下のエラーがログに記録されます。

```
The direct download (Action <action_id>) was canceled after Relay Select:
the address connected to the relay is changed.
```

注: _BESClient_Download_Direct_SubnetList 設定は、直接ダウンロードが予期されるすべての状況 (例: **_BESClient_Download_Direct** 設定が有効になっている場合や、URL が **_BESClient_Download_Direct_Domainlist** 設定に属している場合など) に付加価値を提供します。

この設定について詳しくは、『ダウンロード (（ページ）　　)』を参照してください。

リレー・スイッチ後のダウンロード再開

パッチ 7 から、リレー・スイッチで進行中のダウンロードを中断できるようになりました。デフォルトでは、(前のリレーからの) ダウンロード操作の進行中に BigFix クライアントが新しいリレーに移動した場合、前のリレーがまだクライアントに到達可能であれば、ファイルのダウンロードは前のリレーから続行されます。

以前のリレーが到達不能になった場合にのみ、ダウンロードは失敗し、新しいリレーから新しいダウンロードが試行されます。

_BESClient_Download_ResetOnRelaySwitch という名前の新しい設定を有効にすると、以前のリレーからのダウンロードをまだ到達可能な場合でも停止し、新しいリレーからダウンロードを再開できます。

リレーの切り替え後に、クライアントが以前のリレーから進行中のダウンロードを中断すると、BigFix クライアントのログ・ファイルに以下のエラーが記録されます。

```
The download from Relay (Action <action_id>) was canceled after Relay
 Select:
the relay is changed.
```

この設定について詳しくは、『ダウンロード (（ページ）　　)』を参照してください。

HTTP から HTTPS への自動URLリダイレクト

従来のダウンロードに関する設定に加え、パッチ 8 からは、HTTP から HTTPS への URL リダイレクトが、Bigfix サーバー/リレーと BigFix クライアント (直接ダウンロード) の両方で処理されるようになりました。

HTTPS URL からのダウンロードをサポートするには、リモート・サーバー ID を検証するための適切な信頼された証明書を提供する必要があります。

HTTPS URL からダウンロードした場合、リモート・サーバーの証明書は、BES サポート・サイトを介して配布された CA バンドルを使用して検証されます。CA バンドルは、信頼できる権限のルート証明書と中間証明書を含むファイルです。

この新機能よりも前に、CA バンドルは BigFix サーバーにプリインストールされており、収集の目的で既に使用されていました。この新機能により、ダウンロードを目的として CA バンドルが配布され、BES サポート・サイトを通じて最新の状態に保たれます。

柔軟性を提供するために、サーバー/リレーとクライアントの両方に次の 2 つの新しいプロパティーが追加されました。

- **BESRelay_Download_UntrustedSites** (サーバー/リレー)。これはブール値設定です。0 (デフォルト値) に設定すると、信頼できないサイトからのダウンロードは許可されません。1 に設定すると、信頼できないサイトからのダウンロードが許可されます。

- **_BESRelay_Download_CACertPath** (サーバー/リレー)。これは、カスタム CA バンドルの絶対パスを設定できるストリング設定です。これは、デフォルトの値をオーバーライドします。
- **_BESClient_Download_UntrustedSites** (クライアント)。これはブール値設定です。0 (デフォルト値) に設定すると、信頼できないサイトからのダウンロードは許可されません。1 に設定すると、信頼できないサイトからのダウンロードが許可されます。
- **_BESClient_Download_CACertPath** (クライアント)。これは、カスタム CA バンドルの絶対パスを設定できるストリング設定です。これは、デフォルトの値をオーバーライドします。

注: _BESClient_Download_UntrustedSites および **_BESClient_Download_CACertPath** という名前のクライアント設定は、**BESClient_Download_Direct** という名前のクライアント設定が 1 (有効) に設定されている場合にのみ有効です。

注: _BESRelay_Download_UntrustedSites および **_BESRelay_Download_CACertPath** という名前のリレー設定は、**_BESGather_Download_CheckInternetFlag** という名前のリレー直接ダウンロード設定が 1 (有効) に設定されている場合にのみ有効です。

これらの設定について詳しくは、ダウンロード (（ページ）　　)を参照してください。

データの事前キャッシュの有効化

アクションの実行のためにダウンロードするデータがクライアント上で事前キャッシュされる必要があるかどうかとその方法を、クライアントごとに指定できます。

クライアント上でのアクションの実行のために要求されるデータのダウンロードを開始するタイミングとして、以下のいずれかを選択できます。

すべての制約が満たされた後。

この場合、すべての制約が満たされた後、*pre-fetch* 領域にデータがすべてダウンロードされるまで、アクションの実行は開始されません。この場合、データのダウンロードが制約自体になります。pre-fetch 領域へのデータのダ

ウンロードが完了した時点で、アクションのすべての制約が満たされ、データが *__Download* 領域に移動し、アクションの実行が可能になります。リスクとしては、ダウンロードの時間が予想以上に長くなる可能性があり、最悪の場合、アクションの実行タイム・ウィンドウがデータのダウンロードが完了する前に経過し、アクションの実行が妨げられる可能性があることが挙げられます。

すべての制約が満たされる前。

この場合、アクションがクライアントに関連付けられるとすぐに、開始時刻などの制約がすべて評価されないうちにデータのダウンロードが開始されます。データは、クライアント・ディスクの *pre-cache* 領域にダウンロードされます。すべての制約が満たされた時点で、ダウンロードされたデータが pre-cache 領域から *pre-fetch* 領域に移動します。アクションの開始準備が整った時点で、データが *__Download* 領域に移動し、アクションの実行が開始されます。

このため、アクションを早く開始できます。ユーザーに表示されるオファーの場合には、オファーを受け入れた後、オファーがダウンロードされるのをユーザーが待機する時間が短縮されます。

この場合の潜在的なリスクとして、ディスク容量の不足が原因となってアクションのデッドロックが発生する可能性が挙げられます。また、クライアント・システムにアクション・グループのダウンロードすべてを同時に配置することがディスク容量構成で許可されていないために、グループ・アクションを実行した場合、そのグループがまったく開始されないということも起きえます。
BigFix V9.5.10 以降では、クライアント設定
_BESClient_Download_PreCacheStageContinueWhenDiskLimited
を使用することにより、このリスクがグループ・アクションの処理に影響を及ぼすのを防ぐことができます。**_BESClient_Download_PreCacheStageContinueWhenDiskLimited=1** を設定すると、ダウンロードを必要とする最初のサブアクションがそのサブアクションのためのダウンロードをすべて収集できる限り、そのクライアントでは他のすべての制約が満たされていると想定してグループ・アク

ションを開始できます。また、ディスク容量要件 (**DiskLimited** 制約または **DiskFreeLimited** 制約) が原因となって、すべてのサブアクションのために事前キャッシュされたダウンロードがシステム上で同時に使用できない場合でも、アクション処理を続行できます。

注: この設定はシングル・アクション処理には影響を及ぼしません。つまり、その特定のアクションに必要なダウンロードを保持するだけの十分なディスク容量がないことが原因で引き続きシングル・アクションまたはサブアクションの実行が制約とブロックの対象になる可能性があります。

BigFix Client のデフォルトの設定は、**_BESClient_Download_PreCacheStageContinueWhenDiskLimited=0** です。これは、グループ・アクションを開始できるのはすべてのサブアクション・ダウンロードがシステム上で同時に使用可能になった後ということです。

動的ダウンロードのホワイトリスト

動的ダウンロードにより、関連句を使用して URL を指定する機能が追加され、アクション・スクリプトの柔軟性が高まります。

静的ダウンロードの場合と同様に、動的ダウンロードでも、サイズまたは sha1 の確認を含むファイルを指定する必要があります。ただし、URL、サイズ、および sha1 は、アクション・スクリプトの外部にあるソースから取得されたものであってもかまいません。この外部ソースは、変動する新規ダウンロードのリストを含むマニフェストである場合があります。この手法により、アンチウィルスまたはセキュリティー・モニターなどの、短時間で変更されるファイルや、スケジュールに従って変更されるファイルにアクセスしやすくなります。

このような柔軟性があるため、十分な注意が必要になります。どのクライアントでも動的ダウンロードを使用してファイルを要求することができるため、無差別にファイルをホストするために自分のサーバーを他の人が使用する機会を与えることになります。これを回避するために、動的ダウンロードではホワイトリストを使用します。URL (アクション・スクリプト内のリテラル URL を使用して明示的に認可されたものではない) からのダウン

ロード要求はすべて、以下のファイルに含まれている URL のホワイトリストで指定された基準のいずれかを満たす必要があります。

Windows システムの場合:

```
<Server Install Path>\Mirror Server\Config
\DownloadWhitelist.txt
```

Linux システムの場合:

```
<Server Install Path>/Mirror Server/config/
DownloadWhitelist.txt
```

このファイルには、Perl regex 形式を使用した正規表現のリストが改行で区切られて記述されています。例えば次のようになります。

```
http://.*\.site-a\.com/.*
http://software\.site-b\.com/.*
http://download\.site-c\.com/patches/JustThisOneFile\.qfx
```

最初の行は最も制限が緩く、site-a ドメイン全体のすべてのファイルのダウンロードを許可します。2 行目では、特定のドメイン・ホストを要求しており、3 行目では最も制限が厳しく、URL を「JustThisOneFile.qfx」という名前の 1 つのファイルに制限しています。要求された URL がホワイトリスト内の項目に一致しなかった場合、ダウンロードは直ちに失敗となります。失敗のステータスは NotAvailable です。合格しなかった URL を含む注記が、リレー・ログに書き込まれます。ホワイトリストが空であるか、存在しない場合、すべての動的ダウンロードは失敗します。「.*」(ドットとアスタリスク) というホワイトリスト項目があると、どの URL でもダウンロードできます。

カスタム・クライアント・ダッシュボードの作成

コンソール内のダッシュボードに類似したカスタム・クライアント・ダッシュボードを作成できます。ダッシュボードは、ローカル・コンピューターを分析し、現在の結果を表示することができる埋め込み関連句を含む HTML ファイルです。

ダッシュボードを持つクライアントには、結果のレポートを表示する追加のタブがあります。ダッシュボード・グローバル変数は、許可にかかわらず、他のオペレーターが所有するカスタム・ダッシュボードからアクセスできます。

クライアント・ダッシュボードを作成するには、__BESData フォルダー内に __UISupport (先頭の下線に注意してください) という名前の新規フォルダーを作成する必要があります。これは、クライアント・フォルダーのサブフォルダーであるため、最終的なパス名は次のようになります。

Program Files/BigFix Enterprise/BES Client/__BESData/__UISupport

このフォルダーに、ダッシュボード・ファイル (`_dashboard.html`) と付随するグラフィック・ファイルを置きます。次にクライアントを起動すると、クライアントはこれらのファイルをインターフェースに取り込んで **「ダッシュボード」** タブに追加します。このタブをクリックすると、ダッシュボードは各関連句の最新の値を計算し、それを表示します。

カスタム・ダッシュボードに対するそれ以上の変更は __UISupport フォルダーで行い、クライアント・コンピューターを再起動することによってプロモートする必要があります。

関連文は、次の形式を使用して、HTML で特殊なタグ内に埋め込まれます。

```
      <?relevance statement ?>
```

例えば、時刻を見つけて表示するには、次のようにします。

```
      <?relevance now ?>
```

クライアントは、この文を含むページを表示するとき、関連句「now」を評価し、タグの値を置き換えます。次のサンプル HTML では、「Date:」という単語を表示し、次に現在の日時を表示します。

```
<html>
 <body>
 Date: <?relevance now ?>
 </body>
</html>
```

関連度の評価を更新するには、次の行をファイルに追加します。

```
<html>
 <body>
 Date: <?relevance now ?>
```

```
 <A href="cid:load?page=_dashboard.html"> Refresh </A>
 </body>
</html>
```

この **「Refresh」** というリンクによって、ページが再読み込みされます。ページが再読み込みされると、関連句が再評価されます。他の関連式をこのページに追加する方法は、簡単に分かります。

例えば、オペレーティング・システムおよびコンピューター名を表示するには、以下の 2 行を追加します。

```
<html>
 <body>
 Date: <?relevance now ?>
 Operating System: <?relevance name of operating system ?>
 Computer Name: <?relevance computer name ?>
 <A href="cid:load?page=_dashboard.html"> Refresh </A>
 </body>
</html>
```

スタイル・シートを使用して、出力のフォーマットを設定できます。プリセット・フォーマット用の、デフォルトのスタイル・シートである **offer.css** を使用できます。以下の例のダッシュボードは、タイトル、ヘッダー、更新リンク、および取得プロパティー値のセクションで構成されます。

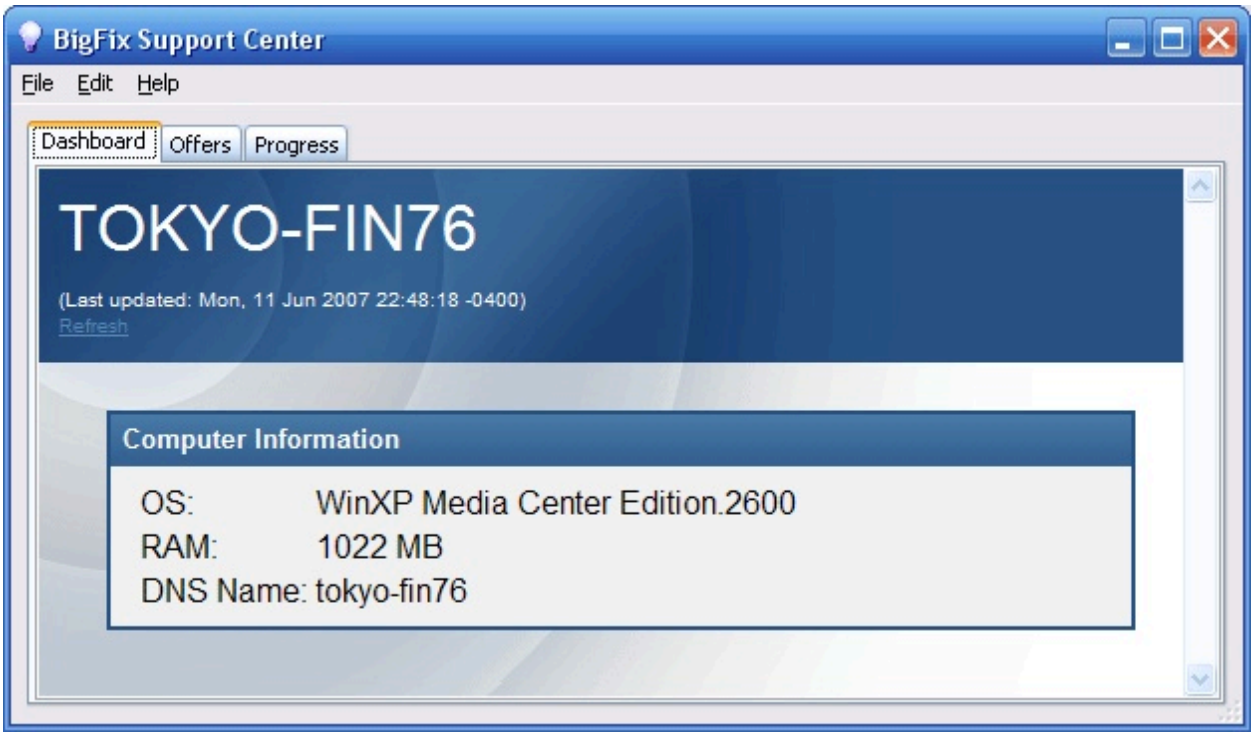
```
<html>
      <head>
         <link type="text/css" rel="stylesheet" href="offer.css"></link>
         <title>BigFix Dashboard Example</title>
      </head>
      <body>

       <div class="header">
         <div class="headerTitle">
             <font size="6"><?relevance computer name ?></font>
```

```
          </div>
          <div class="headerCategory">
              <font size="1">(Last updated: <?relevance now ?>)</font><BR>
            <div><font size="1">
               <a href="cid:load?page=_dashboard.html">Refresh</a></font>
            </div>
          </div>
        </div>

           <div class="section">
                 <div class="sectionHeader">Computer Information</div>
                 <div class="subsection">
                     <table>
                         <tr>
                              <td valign="top">OS: </td>
                              <td><?relevance operating system ?></td>
                         </tr>
                         <tr>
                            <td valign="top">RAM: </td>
                            <td><?relevance (size of ram)/1048576 ?> MB</td>
                         </tr>
                         <tr>
                            <td valign="top">DNS Name: </td>
                            <td><?relevance dns name ?></td>
                         </tr>
                     </table>
                 </div>
           </div>
    </body>
</html>
```

offer.css が正しく機能するには、以下のグラフィックス・ファイルが、クライアント・ディレクトリーから __UISupport ディレクトリーにコピーされなければなりません。

```
bodyBg.jpg,
bodyHeaderBg.jpg
bullet.gif
sectionHeaderBG.gif
```

クライアントから実行された場合、このダッシュボードは以下の出力を生成します。



関連式についてさらに調べるには、「Relevance Language リファレンス」を参照してください。

クライアントの地理的位置指定

クライアントは、多くの場合リモート・オフィスでインストールされるため、クライアントにその位置をレポートさせるプロパティーを作成すると役立ちます。

「ロケーション・プロパティー・ウィザード」 を使用して、 BigFix 内でロケーション・プロパティーを作成できます。

1. コンソールで、**「BigFix 管理」**ドメインに移動し、**「コンピューター管理」**をクリックし、次に**「ロケーション・プロパティー・ウィザード」**をクリックします。ウィザード文書が開きます。
2. ウィザードは、クライアントが自身のサブネット、IP 範囲、またはその他の情報に基づいて自身を識別できるようにする名前付きプロパティーを作成します。ウィザードの指示を読んで、プロパティーを作成します。

クライアントのロック

ネットワーク内の任意の BigFix クライアントのロック状態を変更することができます。これにより、Fixlet アクションの実行対象から、特定のコンピューターまたはコンピューター・グループを除外することができます。

これは、例えば、特定の開発用コンピューターを変更または更新から除外したい場合に役立ちます。また、限られた一連のロック解除されたコンピューターで新規 Fixlet アクションをテストし、ネットワークの残りの部分はロックされたままにするための強力な手法を提供します。クライアント・コンピューターは、(明示的にロック解除されるまで) 永続的にロックすることも、定義した期間だけロックすることもできます。

アクションを送信することにより、クライアントのロック状態に変更を加えます。結果として、コンソール・オペレーターは、どのコンピューターをロックまたはロック解除する場合も、正しい認証を提供する必要があります。クライアントがロックされている場合でも、 クロックの変更アクション、ロック解除アクション、BES サポート・サイトからのアクションなど、 一部のアクションについては、引き続き、クライアントで受け入れることができます。

コンピューターをロックまたはロック解除するには、以下の手順に従います。

1. ドメイン・パネルのナビゲーション・ツリーで**「コンピューター」**アイコンをクリックして、ネットワーク接続された BigFix クライアント・コンピューターのリスト・パネルを表示します。
2. ロックするコンピューターを選択します。

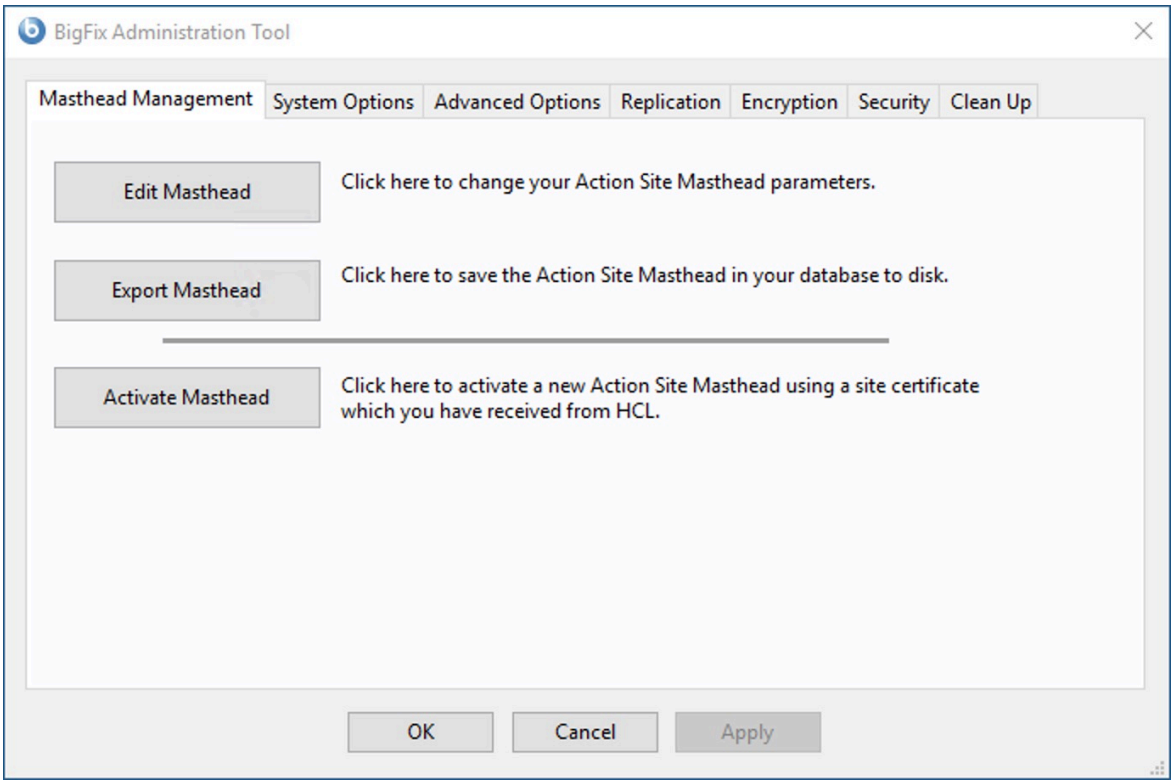
3. 右クリックし、ポップアップ・メニューから **「コンピューターの設定を編集」** を選択します (または **「編集」** メニューから **「コンピューターの設定を編集」** を選択します)。「設定の編集」ダイアログが開きます。
4. チェック・ボックスをクリックして、コンピューターをロックまたはロック解除します。

コンソールでは、ロックに有効期限を設定するための明示的なインターフェースは用意されていませんが、この操作を実行するカスタム・アクションを作成できます。詳しくは、BigFixDeveloper サイト を参照してください。

Windows システムでのマストヘッドの編集

「**BigFix管理ツール**」を使用して、マストヘッドに格納されているデフォルト・パラメーターを変更できます。

1. **「スタート」>「すべてのプログラム」>「BigFix」>「BigFix 管理ツール」** を選択してプログラムを起動します。
2. 秘密鍵 (`license.pvk`) を参照し、**「OK」** をクリックします。
3. **「マストヘッドの管理」** タブを選択し、**「マストヘッドの編集」** をクリックします。



4. デジタル署名を確認するために使用される公開鍵とともに構成およびライセンスの情報が含まれる、マストヘッド・ファイルのパラメーターを入力します。このファイルは資格情報フォルダーに保存されます。

   以下のオプションを編集できます。

   サーバーのポート番号:

   通常、この番号を変更する必要はありません。52311 が推奨ポート番号ですが、異なるポートの方が特定のネットワークでの利便性が高い場合は、異なるポートを選択できます。通常、ポートは、IANA が管理するプライベート・ポートの範囲 (49152 から 65535) から選択します。予約済みのポート番号 (ポート 1 から 1024) を使用できますが、トラフィックを正常にモニターする機能または制限する機能が低下する可能性があり、特定のアプリケーションについてポート番号を使用できなくなります。BigFix が正しく動作しないおそれがあるため、クライアントをインストールしてマストヘッドを作成した 後に サーバーのポート

番号を変更しないでください。追加情報については、次のセクションの「ポート番号の変更」を参照してください。

収集間隔:

このオプションは、サーバーからの通知がない状態でクライアントが待機する時間を決定します。この時間が経過すると、クライアントは、新規コンテンツが使用可能であるかどうかを確認します。一般にサーバーは、新規コンテンツを収集するたびに、UDP 接続を通じて新規コンテンツが提供されていることをクライアントに通知することを試み、この遅延を回避します。ただし、UDP がファイアウォールによってブロックされているか、またはネットワーク・アドレス変換 (NAT) によってサーバーの観点からクライアントの IP アドレスが再マップされる状況では、クライアントからタイムリーに応答を得るためには、間隔を短くすることが必要になります。収集レートが高くても、差分のみが収集されるため、サーバーのパフォーマンスへの影響はほんのわずかです。クライアントは、すでに保有する情報は収集しません。

初期アクション・ロック:

インストール後にクライアントが自動的にロックされるようにする場合、すべてのクライアントの初期ロック状態を指定します。ロックされたクライアントは、どの Fixlet メッセージがそのクライアントの適用対象であるかをレポートしますが、アクションは適用しません。デフォルトでは、クライアントがロックされないままにして、後から特定のクライアントをロックします。ただし、新規にインストールされたクライアントを制御しやすくするため、最初からクライアントをロックした状態にしておき、その後個別にロック解除したい場合もあります。あるいは、一定の期間 (分単位) だけ、クライアントがロックされるように設定することもできます。

アクション・ロック・コントローラー:

このパラメーターによって、誰がアクション・ロック状態を変更できるかが決まります。デフォルトは**「コンソール」**です。これは、管理権限を持つすべてのコンソール・オペレーターに、ネットワーク内の任意のクライアントのロック状態を変更することを許可します。ロックの制御

をエンド・ユーザーに委任したい場合は、**「クライアント」**を選択できますが、これは推奨されません。

アクションのロックから次のサイト URL を除外する:

まれに、特定の URL を、すべてのアクション・ロックから除外することが必要な場合があります。このボックスにチェック・マークを付け、除外する URL を入力します。指定できるサイト URL は 1 つのみであり、先頭を `http://` にする必要があります。

注: ベースライン・コンポーネントは、別のサイトから取得可能であるため、アクション・ロックから除外されません。

すべてのクライアントに対する最近のフォールバック・リレー (ルート・サーバーに置き換え)

クライアントが設定内で指定したいずれのリレーにも接続していない場合は、クライアントにフォールバック・リレーを定義する必要が生じることがあります。このチェック・ボックスを選択して、お使いの環境のフォールバック・リレーを次のいずれかのフォーマットに指定します。

- ホスト名。たとえば、*myhostname* です。
- 完全修飾ドメイン名 (FQDN)。たとえば、*myhostname.mydomain.com* です。
- IP アドレス。たとえば、*10.10.10.10* です。

このチェック・ボックスをオフにしてフォールバック・リレーを定義する場合は、ご自身の環境にあるルート・サーバーが使用されます。

注: フォールバック・リレーを指定する前に、ルート・サーバーに直接レポートするすべてのクライアントまたはリレーにリレーと定義されたルート・サーバーがあることを確認します。この設定により、エンドポイントでルート・サーバーを選択できなくなるということはありません。BES ルート・サーバーの *_BESRelay_Register_Affiliation_AdvertisementList*

を、DoNotSelectMe など、クライアントで設定されないグループ名に設定します。

FIPS 140-2 に準拠した暗号を使用する必要がある

ネットワークを連邦情報処理標準に準拠させるには、このボックスにチェック・マークを付けます。これにより、すべての BigFix コンポーネントが FIPS モードへの移行を試みるように、マストヘッドが変更されます。デフォルトでは、クライアントは、正しく FIPS モードに入ることができない場合、非 FIPS モードのままとなります。これは、特定のレガシー・オペレーティング・システムでは問題となる場合があります。このボックスにチェック・マークを付けると、クライアントの起動時間が 2 秒から 3 秒ほど長くなる可能性があります。

アーカイブでの Unicode ファイル名の使用を許可する (Allow use of Unicode filenames in archives)

この設定は、BigFix アーカイブでファイル名を書き込む際に使用されるコード・ページを指定します。ファイル名を UTF-8 コード・ページで書き込むには、このボックスにチェック・マークを付けます。
ローカル適用環境のコード・ページ (例えば、Windows-1252 や Shift JIS) を使用してファイル名を書き込む場合は、このボックスにチェック・マークを付けないでください。BigFix V9.5 のフレッシュ・インストールを実行する場合、デフォルトでは、ファイル名は UTF-8 で書き込まれます。

注: BigFix 環境を V9.5 にアップグレードすると、ファイル名はデフォルトでローカル・デプロイメントのコード・ページで書き込まれます。

5. **「OK」** をクリックして、変更を入力します。

注: マストヘッドの変更は、すでにデプロイされたクライアントには影響しませんが、管理ツール (**「マストヘッドの管理」** タブ) を使用してマストヘッドをエクスポートし、BigFix サーバーの BES Installers ディレクトリー (デフォルト・ディ

レクトリー:) 内のマストヘッドを置き換えることができます。`<drive>:\Program Files\BigFix Enterprise\BES Installers`これにより、新規にデプロイされた、あるいはインストールされたクライアントがこれらの変更を使用するようになります。

Linux システムでのマストヘッドの編集

マストヘッドを変更するには、スーパーユーザーとして以下のコマンドを実行します。

```
./BESAdmin.sh -editmasthead -sitePvkLocation=<path+license.pvk>
[ -sitePvkPassword=<password> ]
[ -display ] [ -advGatherSchedule=<0-10> ] [ -advController=<0-2> ]
[ -advInitialLockState=<0|2> | -advInitialLockState=1
 -advInitialLockDuration=<num> ]
[ -advActionLockExemptionURL=<url> ]
 [ -advRequireFIPScompliantCrypto=<true|false> ]
[ -advEnableFallbackRelay=0 | -advEnableFallbackRelay=1
-advFallbackRelay=<host> ]
```

各部の意味は以下のとおりです。

`-sitePvkLocation=<path+license.pvk>`

秘密鍵ファイル (*`filename`*`.pvk`) を指定します。管理ツールを実行するには、この秘密鍵ファイルとそのパスワードが必要です。サイト・レベルの署名鍵へのアクセス権およびパスワードを持つユーザーのみが、新規 BigFix オペレーターを作成できます。

注: コマンド構文で使用される表記 `<path+license.pvk>` は、*`path_to_license_file`*`/license.pvk` を表します。

`-sitePvkPassword=<password>`

秘密鍵ファイル (*filename*.pvk) に関連付けられたパスワードを指定します。この設定はオプションです。省略した場合は、コマンドを実行したときに対話式にパスワードの指定を求められます。

-display

マストヘッドの現行設定値を表示します。

-advGatherSchedule (optional, integer)

新規コンテンツが使用可能であるかどうかを確認する前に、サーバーからの通知なしでクライアントが待機する時間を決定します。一般にサーバーは、新規コンテンツを収集するたびに、UDP 接続を通じて新規コンテンツが提供されていることをクライアントに通知することを試み、この遅延を回避します。ただし、UDP がファイアウォールによってブロックされているか、またはネットワーク・アドレス変換 (NAT) によってサーバーの観点からクライアントの IP アドレスが再マップされる状況では、クライアントからタイムリーに応答を得るためには、間隔を短くすることが必要になります。収集レートが高くても、差分のみが収集されるため、サーバーのパフォーマンスへの影響はほんのわずかです。クライアントは、すでに保有する情報は収集しません。有効な値は以下のとおりです。

```
0=Fifteen Minutes,
1=Half Hour, 2=Hour,
3=Eight Hours,
4=Half day,
5=Day,
6=Two Days,
7=Week,
8=Two Weeks,
9=Month,
10=Two Months
```

-advController (optional, integer)

誰がアクション・ロック状態を変更できるかを決定します。デフォルトは**「コンソール」**です。これは、管理権限を持つすべてのコンソール・オペ

レーターに、ネットワーク内の任意のクライアントのロック状態を変更することを許可します。ロックの制御をユーザーに委任したい場合は、**「クライアント」** を選択できますが、これは推奨されません。有効な値は、以下のとおりです。

```
0=console,
1=client,
2=nobody
```

`-advInitialLockState (optional, integer)`

すべてのクライアントの初期ロック状態を指定します。ロックされたクライアントは、どの Fixlet メッセージがそのクライアントの適用対象であるかをレポートしますが、アクションは適用しません。デフォルトでは、クライアントがロックされないままにして、後から特定のクライアントをロックします。ただし、新規にインストールされたクライアントを制御しやすくするため、最初からクライアントをロックした状態にしておき、その後個別にロック解除したい場合もあります。あるいは、一定の期間だけ、それらがロックされるように設定することもできます。有効な値は、以下のとおりです。

```
0=Locked,
1=timed (specify duration),
2=Unlocked
```

`-advInitialLockDuration (optional, integer)`

クライアントをロックしなければならない期間を秒単位で定義します。

`-advActionLockExemptionURL (optional, string)`

まれに、特定の URL を、すべてのアクション・ロックから除外することが必要な場合があります。このボックスにチェック・マークを付け、除外する URL を入力します。

注: 指定できるサイト URL は 1 つのみであり、先頭を `http://` にする必要があります。

`-advRequireFIPScompliantCrypto (optional, boolean)`

連邦情報処理標準をネットワークに実装します。これにより、すべての BigFix コンポーネントが FIPS モードへの移行を試みるように、マストヘッドが変更されます。デフォルトでは、クライアントは、正しく FIPS モードに入ることができない場合、非 FIPS モードのままとなります。これは、特定のレガシー・オペレーティング・システムでは問題となる場合があります。このボックスにチェック・マークを付けると、クライアントの起動時間が 2 秒から 3 秒ほど長くなる可能性があります。

注: FIPS モードを有効にすると、プロキシーへの接続時に一部の認証方式が使用できなくなります。インターネットへのアクセスや BigFix サブコンポーネントとの通信にプロキシーを使用することを選択した場合は、プロキシー構成が `digest`、`negotiate` または `ntlm` 以外の認証方式を使用するようにセットアップされていることを確認してください。

`-advEnableFallbackRelay (optional,boolean)`

クライアントが設定内で指定したいずれのリレーにも接続していない場合は、クライアントに対するフォールバック・リレーを有効または無効にします。フォールバック・リレーを定義しないと、環境内のルート・サーバーが使用されます。

`-advFallbackRelay (optional, string)`

環境内のフォールバック・リレーのホスト名を次のいずれかのフォーマットに指定します。

- ホスト名。たとえば、*myhostname* です。
- 完全修飾ドメイン名 (FQDN)。たとえば、*myhostname.mydomain.com* です。
- IP アドレス。たとえば、*10.10.10.10* です。

注: フォールバック・リレーを指定する前に、ルート・サーバーに直接レポートするすべてのクライアントまたはリレーにリレーと定義されたルート・サーバーがあることを確認します。この設定により、エンドポイントでルート・サーバーを選択できなくなるということはありません。BES ルート・サーバーの *_BESRelay_Register_Affiliation_AdvertisementList* を、DoNotSelectMe など、クライアントで設定されないグループ名に設定します。

サーバーのパスワードの難読化

サーバーのパスワードを難読化されたパスワードに置き換えることができます。そのためには、置き換える元のパスワードのタイプと新規パスワードを指定します。

パスワードは難読化され、Windows システムの場合はレジストリーに保管され、Linux システムの場合は構成ファイルに保管されます。

サーバーのパスワードを難読化するには以下のコマンドを実行します。

Windows システムの場合:

```
BESAdmin.exe /updatepassword /type:<type> [ /password:<password>
 ]
/sitePvkFile:<path+license.pvk> [ /sitePassword:<pvk_password> ]
```

各部の意味は以下のとおりです。

`type:<type>`

以下のパスワードのタイプのいずれかを指定します。

server_db

難読化されて更新および記録されるパスワードは、サーバー・データベースとの接続に関連しています。

dsa_db

難読化されて更新および記録されるパスワードは、DSA データベースとの接続に関連しています。

password:<password>

難読化されてから記録されるパスワードを指定します。

sitePvkFile:<path+license.pvk>

秘密鍵ファイル (*filename*.pvk) を指定します。管理ツールを実行するには、この秘密鍵ファイルとそのパスワードが必要です。サイト・レベルの署名鍵へのアクセス権およびパスワードを持つユーザーのみが、新規 BigFix オペレーターを作成できます。

注: コマンド構文で使用される表記 <path+license.pvk> は、*path_to_license_file*\license.pvk を表します。

sitePassword:<password>

秘密鍵ファイル (*filename*.pvk) に関連付けられたパスワードを指定します。この設定はオプションです。省略した場合は、コマンドを実行したときに対話式にパスワードの指定を求められます。

Linux システムの場合:

各部の意味は以下のとおりです。

type=<type>

以下のパスワードのタイプのいずれかを指定します。

server_db

難読化されて更新および記録されるパスワードは、サーバー・データベースとの接続に関連しています。

dsa_db

難読化されて更新および記録されるパスワードは、DSA データベースとの接続に関連しています。

`password=<password>`

難読化されてから記録されるパスワードを指定します。

`sitePvkLocation=<path+license.pvk>`

秘密鍵ファイル (*`filename`*`.pvk`) を指定します。管理ツールを実行するには、この秘密鍵ファイルとそのパスワードが必要です。サイト・レベルの署名鍵へのアクセス権およびパスワードを持つユーザーのみが、新規 BigFix オペレーターを作成できます。

注: コマンド構文で使用される表記 `<path +license.pvk>` は、*`path_to_license_file/`*`license.pvk` を表します。

`-sitePvkPassword=<password>`

秘密鍵ファイル (*`filename`*`.pvk`) に関連付けられたパスワードを指定します。この設定はオプションです。省略した場合は、コマンドを実行したときに対話式にパスワードの指定を求められます。

グローバル・システム・オプションの変更

最小更新間隔や Fixlet の表示などの基本的なシステム・デフォルトを変更するには、以下の手順を実行します。

Windows システムの場合:

1. **「スタート」>「すべてのプログラム」>「BigFix」>「BigFix 管理ツール」**から管理ツールを起動します。
2. **「システム・オプション」** タブを選択します。
3. 上部で、グローバルな **「最小更新間隔 (Minimum Refresh)」** を設定できます。デフォルトは 15 秒です。この設定は、反応性もよく、ネットワーク負荷も低いため、バランスがとれています。これらの通信がネットワークに影響を及ぼす場合は、最小値を 60 秒以上に増やすことができます。
4. デフォルトでは、すべてのコンソール・オペレーターが外部サイトを表示できますが、**「デフォルトの Fixlet 表示」** というセクションでこれを変更できます。マスター・オペレーターだけが外部コンテンツを表示できるようにするには、下部のボタンをクリックします。マスター・オペレーターは、BigFix コンソールで、コンソールのツールバーにある **「非表示コンテンツの表示」** ボタンをクリックすることで、非表示になっている外部コンテンツ・サイトを表示できます。

Linux システムの場合:

1. `/opt/BESServer/bin` コマンド・プロンプトから、以下のようにコマンド・ラインを開始します。

```
./iem login --server=servername:serverport --user=username
 --password=password
```

2. `/opt/BESServer/bin` コマンド・プロンプトから、以下のコマンドを実行します。

```
./iem get admin/options  > /appo/options.xml
```

3. `/appo/options.xml` ファイルで、

```
<?xml version="1.0" encoding="UTF-8"?>
<BESAPI xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:noNamespaceSchemaLocation="BESAPI.xsd">
    <SystemOptions
 Resource="https://nc926065:52311/api/admin/options">
        <MinimumRefreshSeconds>15</MinimumRefreshSeconds>
```

```
        <DefaultFixletVisibility>Visible</DefaultFixletVisibility>


    </SystemOptions>
</BESAPI>
```

このファイルの以下のキーワードを編集して、最小更新時間を秒単位で設定し、外部サイトをすべてのコンソール・オペレーターが表示できるようにするか、マスター・オペレーターだけが表示できるようにするかを設定します。

```
<MinimumRefreshSeconds>15</MinimumRefreshSeconds>
<DefaultFixletVisibility>Visible</DefaultFixletVisibility>
```

これらのキーワードに割り当てられている値を変更した場合は、BigFix コンソールのアクティブ・セッションを再始動して、変更を有効にする必要があります。

注: マスター・オペレーターは、BigFix コンソールで、コンソールのツールバーにある **「非表示コンテンツの表示」** ボタンをクリックすることで、非表示になっている外部コンテンツ・サイトを表示できます。

4. 以下のコマンドを実行して、変更したファイルをアップロードします。

```
./iem post /appo/options.xml admin/options
```

BigFix ライセンスの拡張

初めてアクション・サイト・ライセンスを要求すると、特定の期間のライセンスが発行されます。ライセンスが失効する前に、BigFix が、ライセンス更新のために十分な時間が確保されるタイミングで警告を出します。

有効期限に近づくと、BigFix は Fixlet メッセージを使用してユーザーに通知します。同様に、ライセンスに割り振られているクライアント数を超過し始めた場合、BigFix が警告します。ライセンスの有効期限を延長するか、インストール済み環境に新規クライアント・ライセンスを追加するには、以下の手順に従います。

1. HCL 担当員に通知します (延長ライセンスに対する支払いを行っていない場合は、販売担当員または販売店に連絡し、延長ライセンスを購入する必要があります)。
2. サーバーは、ライセンスの新しいバージョンがあるかどうかを毎日チェックします。強制的にサーバーに直ちにチェックさせたい場合は、コンソールで、**「BigFix 管理」** ドメインに移動し、**「ライセンスの概要」** ノードをクリックし、**「ライセンスの更新を確認」** をクリックします。

ライセンスの管理方法について詳しくは、「管理ガイド」の『ライセンスの管理』の章を参照してください。

サイト認証情報の再作成

秘密鍵および公開鍵の暗号化では、BigFix ルートから、サイト管理者、各コンソール・オペレーターまでの署名権限のチェーンが作成されます。

サイト認証情報を失った場合、またはサーバーの IP アドレスを変更した場合、このチェーンは途切れます。この結果は深刻です。HCL に対してサイト証明書の新規要求を再び開始する必要があります。次に、すべてのクライアントを含むシステム全体を再インストールし (クライアントを新しいサーバーに移行する方法について詳しくは、サポート技術者にお問い合わせください)、すべてのユーザーを再作成する必要があります。これが発生した場合、サポート技術者にお問い合わせください。サイト証明書を保護するために、以下の重要なルールに従ってください。

- **サイトの証明書 (license.crt)** および**秘密鍵 (license.pvk)** は**なくさないでください。** 標準手順に従い、重要な機密情報をバックアップし、保護します。
- **サーバーの IP アドレスおよびホスト名やポート番号は変更しないでください。** これらがサイト証明書の主たる識別子であるためです。ライセンス要求時に指定した IP アドレスやポート番号を変更すると、そのライセンスは無効になり、BigFix システムをフレッシュ・インストールしなければならなくなります。サーバーを閉鎖する場合、必ず、同じ IP アドレスと同じポート番号を交換サーバーに適用してください。
- **パスワードは忘れないでください。** パスワードのメモおよび保存については、企業規定に従います。

注: BigFix サイト管理者は、サイト・レベルのキーの現在のパスワードを変更することができます (そのパスワードを知っている場合)。

名前が FileOnlyCustomSite で始まる特別なカスタム・サイトの作成

ファイルをエージェントに誤ってまたは意図的に伝播するために生成された (例えば、収集リセットを実行する場合)、名前が FileOnlyCustomSite で始まる特別なカスタム・サイトを削除することができます。このサイトは、`PropagateFiles.exe` ツールを使用して再作成できます。

`PropagateFile.exe` を使用してログインするには、**「コンソールを使用できます」**権限を有効にする必要があります。

次の手順は、名前が `FileOnlyCustomSite` で始まる特別なカスタム・サイトを再作成する方法を示しています。これらの手順に従うことで、`Unable to find site id for URL` などのエラーを回避することができます。

1. 空のファイルを含むダミー・ディレクトリーを作成します。例:`C:\dummy` と `foo_file.txt`。
2. BigFix サーバー・ディレクトリーに移動します。たとえば、`C:\Program Files (x86)\BigFix Enterprise\BES Server` です。
3. ユーザーに特別なカスタム・サイトへの書き込みを許可するには、次のように `PropagateFiles.exe` ツールを実行します。

   ```
   PropagateFiles.exe CreateFileOnlyCustomSiteUserAuthorization <site
    admin pvk> <site admin password> <serverURL> <operator
    name> <operator password> <site name>
   ```

   各部の意味は以下のとおりです。
   - *site admin pvk*:BigFix ライセンス・ファイル (`license.pvk`) への絶対パスです。
   - *site admin password*: ライセンス・ファイルのパスワードです。
   - *serverURL*: マストヘッド・ファイルからコピーされたサーバー URL です。

- *operator name*: 既存のユーザーの名前です。
- *operator password*: ユーザーのユーザー・パスワードです。
- *site name*: 特別なカスタム・サイトの名前です。サイトにはファイルのみが含まれ、名前は `FileOnlyCustomSite` で始まる必要があります。

たとえば、`PropagateFiles.exe CreateFileOnlyCustomSiteUserAuthorization C:\licenses\licence.pvk lcs_password http://bigfixserver:52311 bigfixuser bfu_password FileOnlyCustomSite_FOO` です。

Windows 資格情報を使用して認証することもできます。

```
PropagateFiles.exe CreateFileOnlyCustomSiteUserAuthorization
 Windowsauthentication <site admin pvk> <site admin
 password> <serverURL> <site name>
```

4. ダミー・ディレクトリーのコンテンツをカスタム・サイトに伝播するには、次のコマンドを実行します。

```
PropagateFiles.exe UpdateFileOnlyCustomSite <serverURL> <operator
 name> <operator password> <pvk file location> <directory to
 propagate> <site name>
```

たとえば、`PropagateFiles.exe UpdatefileOnlyCustomSite http://bigfixserver:52311 bigfixuser bfu_password C:\licenses\licence.pvk C:\dummy FileOnlyCustomSite_FOO` です。
Windows 資格情報を使用して認証することもできます。

```
PropagateFiles.exe UpdateFileOnlyCustomSite
 Windowsauthentication <serverURL> <directory to propagate> <site
 name>
```

重要: この操作により、カスタム・サイトのコンテンツはディレクトリーのコンテンツに置き換えられます。

5. カスタム・サイトとそのファイル・コンテンツをターゲットに配布するには、アクション・スクリプトでカスタム・アクションを実行します。

```
custom site subscribe CustomSite_<site name> as "<site name>" on
 "{parameter "action issue date" of action}"
```

例:

```
custom site subscribe CustomSite_FileOnlyCustomSite_FOO as
 "FileOnlyCustomSite_FOO" on "{now}"
```

`CustomSite_FileOnlyCustomSite_FOO` という特別なカスタム・サイトが作成されます。

特別なカスタム・サイトは、BigFix コンソールには表示されません。サイトが正しく作成されたことを確認するには、次のファイルをチェックします。

- `C:\Program Files (x86)\BigFix Enterprise\BES Server\Mirror Server\Inbox\bfemapfile.xml`
- `C:\Program Files (x86)\BigFix Enterprise\BES Server\Mirror Server\Inbox\GatherState.xml`

クライアント CPU 使用率の構成

BigFix クライアントがエンドポイント・マシンで使用する CPU の量を構成する方法。

適用先

BigFix プラットフォーム

問題

BigFix クライアントによる CPU 使用率および BigFix クライアントが使用する CPU の量を制御する方法。

解決方法

評価サイクル中にエンドポイント・マシンで BigFix クライアントが使用する CPU の量は、以下の 2 つのクライアント設定によって制御されます。

- _BESClient_Resource_WorkIdle
- _BESClient_Resource_SleepIdle

デフォルトでは、_BESClient_Resource_WorkIdle 設定は 10 に設定され、_BESClient_Resource_SleepIdle 設定は 480 に設定されます。

BigFix クライアントは、指定された時間だけ動作してから (関連度の評価)、指定された時間だけスリープ状態に入ります。WorkIdle 設定では各サイクルでスリープ状態になるまでの時間をミリ秒単位で制御し、SleepIdle 設定ではサイクルで作業を実行した後にスリープする時間をミリ秒単位で制御します。WorkIdle 設定が SleepIdle 設定と比べて高い場合、BigFix クライアントは Fixlet の関連度を迅速に評価しますが、CPU 使用率は高くなります。デフォルトでは、WorkIdle は 10 ミリ秒で、SleepIdle は 480 ミリ秒です。10 は 480 の 2% であるため、BigFix クライアントが使用する CPU は最大でも 2% になると見込まれます。WorkIdle と SleepIdle の両方とも、最大値は 500 に設定されています。

カスタム設定でエージェントが使用する CPU の量の上限を決定するには、以下の式を使用します。

最大エージェント %CPU = workidle / (workidle + sleepidle)

例 (デフォルト設定): 10 / (10 + 480) = 2%

BES サポート・サイトのタスク番号 168 により、容易にこれらの設定を管理し、BigFix クライアントが使用する CPU の量を調整できるようになります。

このタスクでは、クライアント CPU 使用率を 5 種類のモードのいずれかに設定できます (アクション・スクリプトで 5 種類のアクションとプリセット値を使用します)。

_BESClient_Resource_WorkIdle	_BESClient_Resource_SleepIdle	CPU モード	CPU 上限	Note
2	500	非常に低い	< 0.5%	推奨されません
10	480	デフォルト	< 1-2%	「デフォルト」

25	460	中	< 5%	推奨されません
50	450	高	< 10%	推奨されません
100	400	非常に高い	< 20%	推奨されません

LPAR を使用する AIX コンポーネントの場合:

- 上限なしモードおよび上限付きモードでライセンス済みキャパシティーがある AIX LPAR の最大エージェント % CPU 計算。
- - 上限なしモードで AIX LPAR に _BESClient_Resource_Entitlement=100 を使用するというのは、ライセンス済みキャパシティーによる最大エージェント CPU% の削減を排除するために使用したと考えることができます。

pSeries ハイパーバイザーの動作については、以下の例を参照してください。

モードあり: 上限なし、ライセンス済みキャパシティー: 0.10、AIX LPAR が 0.10 CPU コアを使用する場合は、100% の使用率。物理 CPU コアが

pSeries プールで完全には使用されていない場合、pSeries ハイパーバイザーにより、AIX LPAR が CPU コアをより多く使用できるようになります。また、pSeries LPAR 重み付けを使用して、実稼働など特定の AIX LPAR が、開発など他の AIX LPAR よりも、CPU リソースに対する優先度が高くなるようにすることもできます。

BigFix コンピューター設定
_BESClient_Resource_WorkNormal、_BESClient_Resource_SleepNormal (設定可能なタスクを BigFix エージェントが実行している場合)。

質問および回答:

質問 1:

Red Hat、SLES、Windows の場合:

最大エージェント % CPU = workidle / (workidle + sleepidle)

0.0476 = 20 / (20 + 400)

0.0476 x 100 = 4.76%

...

_BESClient_Resource_WorkIdle ステートメント「ライセンス済みキャパシティー: 0.10」というのは、CPU が 1/10 であるという意味です。そのため、workidle/sleepidle を計算する場合には常に計算値に 0.1 を乗じて、CPU 全体の 10% であるとシステムに通知されるようにする必要があります。

AIX LPAR の場合、以下のように計算されますか?

最大エージェント % CPU = (workidle / (workidle + sleepidle)) x ライセンス済み

キャパシティー

0.00476 = (20 / (20 + 400)) x 0.10

AIX LPAR に使用可能な CPU のうち、0.00476 x 100 / = 0.476% を使用できます。

回答:

BigFix 最大エージェント % CPU に AIX LPAR ライセンス済みキャパシティーを乗じると、上限なしモードまたは上限付きモードで AIX LPAR の BigFix 最大エージェント % CPU をさらに削減できます。BigFix 最大エージェント % CPU は、単一の物理 CPU コアに基づいています。

質問 2:

上限なしモードの AIX LPAR の場合:BigFix AIX コンピューター _BESClient_Resource_Entitlement を 100 に設定することで、lparstat -i ライセンス済みキャパシティーをオフにして、最大エージェント %CPU を削減できますか?

回答:

はい

質問 3:

_BESClient_Resource_Entitlement が 100 である AIX LPAR 上の BigFix エージェントの場合、最大エージェント %CPU を単一の物理 CPU コアに制限するための動作は Intel 上の Linux および VMware 上の Windows と同じですか?

回答:

はい。

また、BigFix コンピューター設定 _BESClient_Resource_WorkNormal、_BESClient_Resource_SleepNormal もあります (設定可能なタスクを BigFix エージェントが実行している場合)。

質問 4:

BigFix クライアント・プロセスの完了後も引き続き CPU% が高いままなのはなぜですか?

回答:

BigFix 実行するアクションがなくなり、アクション・サイトの評価を完了しても、クライアントの CPU% は高いままです。

注:

- これらの設定は、エンドポイント・マシンの限られたセットでテストしてから、デプロイメント・エンドポイント・マシンのほとんどにデプロイしてください。
- 実際には、CPU 使用率は通常、この比率よりも低くなります (エージェントが IO を待機することがよくあり、それで CPU 時間が生じるため)。
- これらの計算はすべて単一のプロセッサーに適用されるため、複数のプロセッサーを使用している場合、エージェント CPU の全体の割合は、プロセッサーの数で除算されるため、大幅に減少します。例えば、エージェントの CPU 使用率が 5% 未満で、4 個のプロセッサーが搭載されている場合、workidle を 100、sleepidle を 400 に設定する必要があります ([100 / (100 + 400)] / 4 = 5%)。
- CPU 使用率をデフォルトとは別の値に調整し、それで問題が発生した場合は、CPU 使用率をデフォルト値に戻してください。
- これらの CPU 設定は、クライアントのサイクルの評価モードでコンテンツを評価する BES クライアントを厳密に制御します。BigFix エージェントは、

操作の実行部分 (つまり、インストールの開始) の際に、CPU の最大 50% を使用することがあります。これらの CPU スパイクは通常のことであり、長くは続かず、これが顕著になることは通常ありません。BES クライアントで CPU スパイクが続く場合は、実行中のクライアントでアクションに問題があるか、システム上の問題が発生している可能性があります。この点を調査し、BigFix クライアント・ログや BigFix クライアント・デバッグ・ログを分析して問題を特定する必要があります。

ダウンロード操作を完了するためにクライアントが追加の CPU を使用できるようにする

ダウンロード・アクションの後、BigFix クライアントはダウンロード・ファイルに対して SHA コード評価を実行します。大きいファイルの場合、評価にかかる時間が非常に長くなる可能性があります (特にデフォルトの CPU 2% に制限された BigFix クライアントの場合)。BigFix クライアントがこの操作中に追加の CPU を一時的に使用できるようにすることで、この時間を最適化できます。

パッチ 2 以降では、追加の CPU を一時的に使用するよう BigFix クライアントに指示することで、ダウンロードしたファイルの SHA コードを評価する操作を高速化できます。これにより、使用される CPU が増加するにつれて評価に必要な時間が減少するため、ダウンロード・フェーズの時間の最適化が一貫して行われます。

`_BESClient_Resource_WorkFastHashVerify` および `_BESClient_Resource_SleepFastHashVerify` 構成設定により、BigFix クライアントが使用できる CPU の量を管理できます。

これらの設定について詳しくは、CPU 使用率 (（ページ）　　)を参照してください。

`_BESClient_Download_FastHashVerify` 構成設定を使用すると、BigFix クライアントはハッシュ・コードの評価のために追加の CPU を一時的に使用できます (デフォルトでは 25% に増加します)。

設定について詳しくは、「ダウンロード (（ページ）　　)」を参照してください。

クライアント UI メッセージ・ストリングのカスタマイズ

クライアント UI メッセージ・ストリングをカスタマイズできます。クライアント UI の XLAT ファイルを使用して、英語のテキスト出力を変更したり、ローカライズ可能なストリング変換を変更したりすることも可能です。64 ビット・マシンのクライアント UI XLAT ファイルのデフォルトの場所は `C:\Program Files (x86)\BigFix Enterprise\BES Client\BESLib\Reference` です。

以下のように実行します。

1. `C:\Program Files (x86)\BigFix Enterprise\BES Client\BESLib\Reference` で `<lang>.xlat` というファイルを探します。ここで、`<lang>` は対処する言語の識別子です (例えば、ITA)。
2. `C:\Program Files (x86)\BigFix Enterprise\BES Client\BESLib\Reference\<lang>.xlat` を参照し、変更する変換ストリングがある行を識別します。
3. この行を、変更した変換とともに `C:\Program Files (x86)\BigFix Enterprise\BES Client\__BESData\__UISupport\_brand<lang>.xlat` に保存します。
4. BigFix クライアントを再起動します。

注: ファイル `C:\Program Files (x86)\BigFix Enterprise\BES Client\BESLib\Reference\ENU.xlat` (English xlat) は空です。ただし、他の言語ファイルのいずれかで英語ストリングを探して、上記の手順に従うことができます。

注: バージョン 10 へのアップグレード後、カスタム XLAT ファイルの英語の文字列をアップグレードされた `C:\Program Files (x86)\BigFix Enterprise\BES Client\BESLib\Reference\<lang>.xlat` ファイルの文字列に揃える必要があります。

第 22 章. BigFix サーバーのマイグレーション (Windows/MS-SQL)

このセクションでは、BigFix サーバーを既存のハードウェアから新しいコンピューター・システムにマイグレーションするために必要なステップと操作手順について詳しく説明します。

このステップの一般的なユース・ケースは次のとおりです。

- ハードウェアの更新
- OS または SQL Server のアップグレード
- 32 ビットから 64ビット・アーキテクチャーへのマイグレーション
- リモート SQL Server のマイグレーション

このステップは、以下の Windows 版 BigFix サーバーのバージョンに適用されます。

- 9.2
- 9.5
- 10.0

BigFix サーバーのマイグレーションは複雑でリスクがあるため、BigFix サーバーのマイグレーション・プロセスを実行する際は、BigFix 技術者のサポートを受けることを強くお勧めします。

マイグレーションに関する考慮事項

このセクションでは、BigFix ルートまたはアプリケーション・サーバーのマイグレーションに関するいくつかの注意事項とガイドラインを提供します。

一般的な注意事項とガイドライン

- 可能であれば、最初にマイグレーションを隔離されたテストや開発環境で実行し、テストを行う必要があります。
- BigFix 災害対策サーバー・アーキテクチャー (DSA - 災害対策サーバー・アーキテクチャー) を利用する場合は、プライマリー BigFix サーバーの前に、レプリカ・サーバーやセカンダリー・サーバーをマイグレーションする必要があります。
- BigFix サーバーに適用されたカスタム設定は、マイグレーション後に再度実装する必要があります。典型的な例には、Web レポート HTTPS 構成、ダウンロード収集キャッシュ・サイズなどがあります。
- ダウンロード・プラグインとその他の拡張機能やアプリケーションも、新しいインストール場所に再インストールする必要があります。
- 典型的な例には、Unmanaged Asset Importer、Wake on LAN Medic、アップロード・サービス、自動化プラン・エンジンなどがあります。

事前マイグレーション・チェックリスト

- マストヘッドで指定された GatherURL に従って、クライアントを新しい BigFix サーバーに引き続き接続できるようにする戦略が決定されていることを確認します (上記の前提 #1 に対応)。
- BFEnterprise データベースと BESReporting SQL データベースをバックアップします。
- license.crt、license.pvk、マストヘッドなどのサイト・レベルの資格情報をバックアップします。8.1 より前のバージョンを使用している場合は、publisher.pvk や publisher.crt などのユーザーやオペレーターの資格情報もバックアップする必要があります。
- MSSQL データベースへの認証方法を文書化します (SQL と NT)。
- NT 認証を使用する場合は、BigFix サーバー・サービスに使用される NT ドメインやサービスのアカウントを文書化します。
- SQL 認証を使用する場合は、SQL 認証レジストリー値に使用する SQL アカウントを文書化します。
- 次の ODBC 接続を文書化します (スクリーンショットを撮ることを検討します)。bes_BFEnterprise、bes_EnterpriseServer、enterprise_setup、LocalBESReportingServer。

ビット Windows システムの場合、32 ビット・バージョンの ODBC ツール (C:\Windows\SysWOW64\odbcad32.exe) を使用して、システム DSN を構成します。

- プライマリー BigFix サーバーをマイグレーションする場合、ダウンタイムを削減するために、マイグレーションの前に以下を適用することを検討してください。
  - すべてのクライアントで次の BigFix クライアント設定を変更します。
    - _BESClient_Report_MinimumInterval = 3600

      この設定により、エンドポイントから受信するデータ量が減少するため、システムをすばやく復旧し、潜在的なダウンタイムを削減できます。
    - _BESClient_RelaySelect_ResistFailureIntervalSeconds = 21600

      この値は、BES リレーがダウンしてから BES リレー選択を実行するまでの BES クライアントの待機時間を表します。これにより、マイグレーション中の不要な自動リレー選択を防ぐことができます。
  - BigFix コンソールのハートビートを 6 時間に変更します。コンソール・プリファレンス

    注: これは、エンドポイントから受信するデータ量を削減するもう 1 つの方法です。
- マイグレーション手順をよく確認してください。

BigFix ルート・サーバーのマイグレーション

ルート・サーバーをマイグレーションする方法。

BigFix サーバーのマイグレーションを実行する前に、以下のシナリオが当てはまると想定されています。

- プライマリーやマスターの BigFix サーバーをマイグレーションする場合、新しい BigFix サーバーは、マストヘッドやライセンス (https://support.hcltechsw.com/csm?id=kb_article&sysparm_article=KB0023184) で指定されているのと同じ DNS 名やエイリアスまたは IP アドレスを利用する必要があります。それ以外の場合、BigFix インフラストラクチャーは新しい BigFix サーバーと通信できません。これが不可能な場

合は、新しいライセンスを取得する必要があり、サーバーのマイグレーションではなくインフラストラクチャーのマイグレーションを実行する必要があります。これはマイグレーション戦略の重要な要素であり、適切な計画が必要です。

- マストヘッドが IP アドレスを利用する場合、新しいサーバーは同じ IP アドレスを利用する必要があります。
- マストヘッドがホスト名を利用する場合、新しいサーバーは同じホスト名を利用する必要がある場合があります。
- マストヘッドが DNS 名やエイリアスを利用する場合 (ベスト・プラクティスに従って)、マイグレーション・プロセスの一部として、エイリアスを新しい BigFix サーバーに再度指定する必要があります。マストヘッド内で DNS 名やエイリアスを利用する場合は、DNS 名の DNS スイッチを実行すると、エイリアスが新しい BigFix サーバーを指定するようになります。DNS スイッチが伝達されるまで待ちます (DNS サービスやインフラストラクチャーによっては時間がかかる場合があります)。

- 既存の BigFix サーバーは、マイグレーション前に正常に動作しています。
- 新しい BigFix サーバーが構築され、BigFix サーバーの要件を満たし、BigFix サーバーとして機能するように適切に構成されています。特に、OS とデータベース・プラットフォームは、マイグレーションする特定の BigFix バージョンでサポートされている必要があります。
- インストール・フォルダーは、元の BigFix /DSA サーバーと新しい BigFix /DSA サーバーの同じ場所とパスにあります (そうでない場合は、以下の手順で説明するように、ファイルを手動で変更する必要があります)。
- マイグレーションは勤務時間外に実行され、潜在的な影響やダウンタイムを最小限に抑えます。

1. マイグレーションの検証を容易にするために、現在のアクションサイトのバージョンをメモします。
   - BigFix サーバー・バージョンの場合:https://support.hcltechsw.com/csm?id=kb_article&sysparm_article=KB0023338。
   - v8.2 以降では、サーバーの診断ページ (http://<BigFixServer:port>/rd) からアクションサイトのバージョンを取得することもできます。「サイト収集情報」で

「現在のバージョンを取得」要求タイプを選択し、ドロップダウンからアクションサイトの URL を選択し、「送信」をクリックしてアクションサイトのバージョンをメモします。

2. 元のサーバーですべての BES サービスを停止して無効にすることを検討してください。
3. サーバー・バックアップ (（ページ）) の説明に従って、サーバーのバックアップ手順を実行します。
4. サーバーのリカバリー (（ページ）) の説明に従って、サーバーのリカバリー手順を実行します。
5. リストア結果の検証 (（ページ）) の説明に従って、リストア結果の検証手順を実行します。
6. 新しい BigFix サーバーによってホストされているアクションサイトのバージョンが、手順 1 で説明した同じ手順を使用して、手順 1 でメモしたバージョンと一致することを確認します。
7. すべての最上位リレーのリレー選択設定を確認してください。IP アドレスまたはホスト名を使用して元の BigFix サーバーを指すように設定している場合、新しい BigFix サーバーを指すように再設定する必要がある場合があります。
8. 古い BigFix サーバー・コンピューターから BigFix サーバー・ソフトウェアをアンインストールします。このコンピューターで BES サービスを再起動しないでください。古い BigFix サーバーを使用しようとすると、新しい BigFix サーバーでエラーが発生する可能性があります。
9. BESAdmin.exe /resetDatabaseEpoch を実行して、コンソールで新しいサーバーのキャッシュを強制的に更新します。
10. BigFix サーバー・サービスをシャットダウンする前に、クライアント設定とハートビート設定をリセットします。

データベースのマイグレーション

この手順は、リモート・データベースのインストールに適用されます。

始める前に

- BFEnterprise データベースと BESReporting SQL データベースをバックアップします (現在のバックアップは移動の直前に行う必要があります。バックアップ・データベースと運用データベース間に差異があってはなりません)。
- MSSQL データベースへの認証方法を文書化します (SQL と NT)。
  - NT 認証を使用する場合は、BigFix サーバー・サービスに使用される NT ドメインやサービスのアカウントを文書化します。
  - SQL 認証を使用する場合は、SQL 認証レジストリー値に使用する SQL アカウントを文書化します。
- 次の ODBC 接続を文書化します (スクリーンショットを撮ることを検討します)。bes_BFEnterprise、enterprise_setup、LocalBESReportingServer。32 ビット情報と 64 ビット情報の両方を記録して複製します。

注: 64 ビット Windows システムの場合、32 ビット・バージョンの ODBC ツールを使用して、32 ビットのシステム DSN を構成する必要があります。

- ルート・サーバーで ODBC ウィザードを使用して、新しいデータベースの場所 (新しい MS-SQL Server) への基本接続をテストします。
- ダウンタイムを削減するために、マイグレーションの前に以下を適用することを検討してください。
  - すべてのクライアントで次の BigFix クライアント設定を変更します。
    - _BESClient_Report_MinimumInterval = 3600 * この設定により、エンドポイントから受信するデータ量が減少するため、システムをすばやく復旧し、潜在的なダウンタイムを削減できます。
    - 1_BESClient_RelaySelect_ResistFailureIntervalSeconds = 21600 * この値は、BES リレーがダウンしてから BES リレー選択を実行するまでの BES クライアントの待機時間を表します。これにより、マイグレーション中の不要な自動リレー選択を防ぐことができます。

- BigFix コンソールのハートビートを 6 時間に変更します。コンソール・プリファレンス

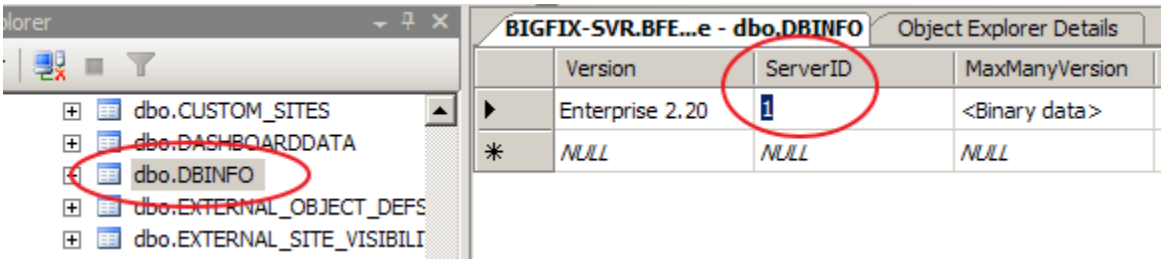
  注: これは、エンドポイントから受信するデータ量を削減するもう 1 つの方法です。

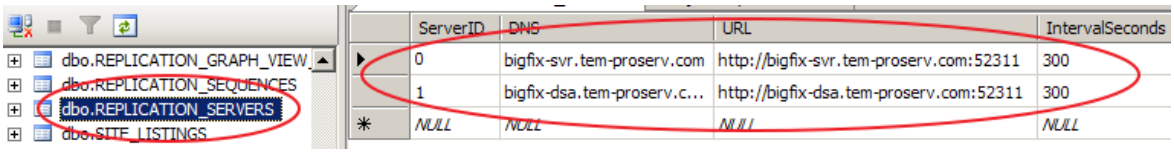
- マイグレーション手順をよく確認してください。

Procedure

1. すべての BES サーバー・サービスを停止します。
2. 現在の SQL Server インスタンス・データベースから BFEnterprise データベースと BESReporting データベースを切り離します。
3. BFEnterprise データベースと BESReporting データベースを新しい SQL Server インスタンスに移動します。
4. BFEnterprise データベースと BESReporting データベースを新しい SQL Server インスタンスに接続します。
5. ODBC システム DSN (bes_BFEnterprise、enterprise_setup、LocalBESReportingServer) を、新しい SQL Server インスタンスを指定するように変更します。この変更により、BigFix サーバー・アプリケーションの再インストールを回避できます。
   - ODBC 接続ウィザードを使用して接続をテストします。
   - 32 ビット構成と 64 ビット構成の両方を更新して検証します。

   注: 64 ビット Windows システムの場合、32 ビット・バージョンの ODBC ツールを使用して、システム DSN を構成する必要があります。
6. DSA を利用する場合は、SQL Server Management Studio を使用して BFEnterprise データベースに接続し、DBINFO テーブルと REPLICATION_SERVERS テーブルを調べます。





サーバーで DNS エイリアスを使用している場合、その名前は変更されません。ホスト名を使用していて、名前が変更されているときには、その列の値を手動で変更する必要がある場合があります。

注: 設定「HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\BigFix\Enterprise Server\UseRemoteDB」の値は、BigFix データベースがローカルの場合はデフォルトで「0」に設定され、BigFix データベースがリモートの場合は「1」に設定されます。

マイグレーションの確認

BigFix サーバーが正常にマイグレーションされたことを確認するには、このセクションで説明する手順を実行してください。

1. BigFix 診断ツールをチェックして、すべてのサービスが適切に開始されていることを確認します。
2. BigFix 管理ツールにログインします (ツールが正常に開いた場合は、データベース接続が確認され、ツールを閉じることができます)。
3. BigFix コンソールにログインし、ログインが正常に機能し、データベース情報が適切にリストアされたことを確認します。
4. BigFix クライアントと BigFix リレーは、サーバーが使用可能で、サーバーにデータがレポートされることをすぐに通知します。すべてのエージェント・レポートを含む完全リカバリーには、(適用環境のサイズとサーバーが使用できなかった期間によっ

て) 数分から長時間かかる場合があります。どのような状況でも、少なくともいくつかのエージェントは、1 時間以内に更新済み情報をレポートします。

5. いくつかのエージェントが適切にレポートしていることを確認したら、「空白のアクション」(「ツール」>「カスタム・アクションの実行」、「すべてのコンピューター」をターゲットにして「OK」をクリック) をすべてのコンピューターに送信します。空白のアクションはエージェント・コンピューターに何も変更を加えませんが、エージェントは空白のアクションを受け取ったことをレポートします。ほとんどのエージェントが空白のアクションに応答する場合、アクションの送信により BigFix の多くのコア・コンポーネントと通信パスがテストされるため、すべてが正常に機能していることが明確にわかります。
6. Web レポートにログインして、データが正しくリストアされたことを確認します。
7. 問題や質問がある場合は、BigFix サポートにお問い合わせください。

第 23 章. BigFix サーバーのマイグレーション (Linux)

このセクションでは、BigFix サーバーを既存の Linux ハードウェアから新しいシステムにマイグレーションするための基本情報ついて説明します。

以下の手順は、サーバー・コンポーネントのみを対象としています。設定でアプリケーションやサーバーをさらにカスタマイズしている場合は、バックアップとリストアを個別に行う必要があります。

注: BigFix サーバーのマイグレーションは複雑でリスクがあるため、BigFix サーバーをマイグレーションする際は、訓練を受けた BigFix 技術者のサポートを受けることを強くお勧めします。

始める前に

BigFix サーバーのマイグレーションを実行する前に、以下のシナリオが当てはまると想定されています。

- プライマリーやマスターの BigFix サーバーをマイグレーションする場合、新しい BigFix サーバーは、マストヘッドやライセンス (https://support.hcltechsw.com/csm?id=kb_article&sysparm_article=KB0023184) で指定されているのと同じ DNS 名やエイリアスまたは IP アドレスを利用する必要があります。それ以外の場合、BigFix インフラストラクチャーは新しい BigFix サーバーと通信できません。これが不可能な場合は、新しいライセンスを取得する必要があり、サーバーのマイグレーションではなくインフラストラクチャーのマイグレーションを実行する必要があります。これはマイグレーション戦略の重要な要素であり、適切な計画が必要です。
  - マストヘッドが IP アドレスを利用する場合、新しいサーバーは同じ IP アドレスを利用する必要があります。
  - マストヘッドがホスト名を利用する場合、新しいサーバーは同じホスト名を利用する必要がある場合があります。
  - マストヘッドが DNS 名やエイリアスを利用する場合 (ベスト・プラクティスに従って)、マイグレーション・プロセスの一部として、エイリアスを新しい BigFix サーバーに再度指定する必要があります。マストヘッド内で DNS 名やエ

イリアスを利用する場合は、DNS 名の DNS スイッチを実行すると、エイリアスが新しい BigFix サーバーを指定するようになります。DNS スイッチが伝達されるまで待ちます (DNS サービスやインフラストラクチャーによっては時間がかかる場合があります)。

- 既存の BigFix サーバーは、マイグレーション前に正常に動作しています。
- 新しい BigFix サーバーが構築され、BigFix サーバーの要件を満たし、BigFix サーバーとして機能するように適切に構成されています。特に、OS とデータベース・プラットフォームは、マイグレーションする特定の BigFix バージョンでサポートされている必要があります。
- インストール・フォルダーは、元の BigFix /DSA サーバーと新しい BigFix /DSA サーバーの同じ場所とパスにあります (そうでない場合は、以下の手順で説明するように、ファイルを手動で変更する必要があります)。
- マイグレーションは勤務時間外に実行され、潜在的な影響やダウンタイムを最小限に抑えます。

手順

1. マイグレーションの検証を容易にするために、現在のアクションサイトのバージョンをメモします。
    - BigFix サーバー・バージョンの場合:https://support.hcltechsw.com/csm?id=kb_article&sysparm_article=KB0023338。
    - v8.2 以降では、サーバーの診断ページ (http://<BigFixServer:port>/rd) からアクションサイトのバージョンを取得することもできます。「サイト収集情報」で「現在のバージョンを取得」要求タイプを選択し、ドロップダウンからアクションサイトの URL を選択し、「送信」をクリックしてアクションサイトのバージョンをメモします。
2. 元のサーバーですべての BES サービスを停止して無効にすることを検討してください。
3. サーバー・バックアップ (（ページ）) の説明に従って、サーバーのバックアップ手順を実行します。
4. サーバーのリカバリー (（ページ）) の説明に従って、サーバーのリカバリー手順を実行します。

5. リストア結果の検証 (（ページ）　　) の説明に従って、リストア結果の検証手順を実行します。
6. 新しい BigFix サーバーによってホストされているアクションサイトのバージョンが、手順 1 で説明した同じ手順を使用して、手順 1 でメモしたバージョンと一致することを確認します。
7. すべての最上位リレーのリレー選択設定を確認してください。IP アドレスまたはホスト名を使用して元の BigFix サーバーを指すように設定している場合、新しい BigFix サーバーを指すように再設定する必要がある場合があります。
8. 古い BigFix サーバー・コンピューターから BigFix サーバー・ソフトウェアをアンインストールします。このコンピューターで BES サービスを再起動しないでください。古い BigFix サーバーを使用しようとすると、新しい BigFix サーバーでエラーが発生する可能性があります。
9. ./BESAdmin.sh -resetDatabaseEpoch を実行して、コンソールで新しいサーバーのキャッシュを強制的に更新します。
10. BigFix サーバー・サービスをシャットダウンする前に、クライアント設定とハートビート設定をリセットします。

リモート・サーバーでのデータベースの再配置

データベースを再配置し、再配置後にアップグレードする方法。

一般ガイドライン

BigFix サーバーがインストールされているのと同じマシン上にあるローカル BigFix データベースを別のリモート・サーバーに移動する場合や、ローカル・データベースをリモート DB2 サーバーから別のリモート DB2 サーバーに再配置する必要がある場合は、以下のガイドラインを考慮してください。

現在の DB2 サーバーから BigFix データベースをバックアップして、新しい DB2 サーバーで復元する必要があるため、同じ DB2 レベルを使用してバックアップ/復元することを強くお勧めします。

必要に応じて、BigFix データベースを復元した後にのみ、DB2 アップグレードを実行します。

別のサーバーへの BigFix データベースの再配置は現在、同じインスタンス名 (デフォルトは db2inst1) を使用している場合にのみサポートされます。

DB2 のインストール要件および構成について詳しくは、以下のリンクを参照してください。

データベースの要件

DB2 のインストールと構成

BigFix データベースのマイグレーション

BigFix データベースをマイグレーションするには、以下の手順を実行します。

1. 開始および宛先の DB2 が同じレベルにあり、新しい DB2 システムが同じインスタンス (デフォルト: db2inst1) を使用していることを確認します。
2. すべての BigFix サービスを停止します。
3. DB2 データベースのバックアップ・コマンドを実行します。

   ```
   BACKUP DB BFENT COMPRESS
   BACKUP DB BESREPOR COMPRESS
   ```

4. 新しい DB2 システムで BigFix データベースを復元します。

   ```
   RESTORE DB BFENT
   RESTORE DB BESREPOR
   ```

5. BigFix サーバーに新しいデータベースをカタログします。BigFix サーバーから、以下の DB2 コマンドを実行します。

   ```
   UNCATALOG DATABASE BFENT
   UNCATALOG DATABASE BESREPOR
   UNCATALOG NODE TEM_REM
   CATALOG TCPIP NODE TEM_REM REMOTE {host} SERVER {port}
   CATALOG DATABASE BFENT AS BFENT AT NODE TEM_REM
   CATALOG DATABASE BESREPOR AS BESREPOR AT NODE TEM_REM
   ```

   各表記の意味は次のとおりです。

{host} は DB2 リモート・サーバー・ホスト名で、{port} は使用されている DB2 リモート・サーバー・ポートです。

6. `/var/opt/BESServer/besserver.config` で (必要に応じて) BigFix サーバー・データベース設定を更新します。

```
[Software\BigFix\EnterpriseClient\Settings\Client\_BESServer_Database_DatabaseAddress]
value = <new_hostname>

[Software\BigFix\EnterpriseClient\Settings\Client\_BESServer_Database_Port]
value = "<new_port_number>"
```

7. Web レポートがインストールされている場合は、`/var/opt/BESWebReportsServer/beswebreports.config` で (必要に応じて) 以下の設定を更新します。

```
[Software\BigFix\Enterprise Server\FillAggregateDB]
DatabaseAddress = <new_hostname>
Port = <new_port_number>
```

8. 「データベース・パスワードの変更」で説明されているように、BigFix サーバーで DB2 パスワードを更新します (新しい DB2 サーバー上の db2inst1 ユーザーのパスワードが古い DB2 サーバー上のものと異なっている場合)。
9. BigFix サーバー・サービスを開始します (WebUI を除く)。

```
/etc/init.d/besserver start
/etc/init.d/besfilldb start
/etc/init.d/besgatherdb start
/etc/init.d/beswebreports start
/etc/init.d/bespluginportal start (if installed)
/etc/init.d/besclient start
```

10. コンポーネントが開始され、新しい構成済みデータベースとの接続が機能していることを確認します。

11. WebUI がインストールされている場合は、BES コンソールから BES サポート Fixlet ID 2687 を実行して、「BigFix サーバー・データベース・ホスト」を新しいデータベース構成仕様で更新します (これで WebUI サービスも開始されます)。
12. REPLICATION_SERVERS テーブルの DNS フィールドを新しい DB2 サーバー・ホスト名で更新します。

再配置後のデータベースのアップグレード

DB2 アップグレードを実行する必要がある場合は、ローカルからリモートへの再配置を行った後に、以下の手順を実行できます。

1. BigFix サーバー・マシンからローカル DB2 サーバーを削除します。
   a. すべての BigFix サービスを停止します。
   b. BigFix サーバーから、以下の DB2 コマンドを実行します。

   ```
   UNCATALOG DATABASE BFENT
   UNCATALOG DATABASE BESREPOR
   UNCATALOG NODE TEM_REM
   ```

   c. DB2 サーバーのアンインストールを続行します。詳しくは、以下の IBM 資料を参照してください。DB2 データベース製品のアンインストール。
   d. DB2 のユーザーおよびグループを削除します。
2. サーバーをアップグレードする場合と同じバージョンの IBM Data Server をインストールします。
3. 新しい DB2 サーバーで、IBM の資料に従って、DB2 サーバーをターゲット・バージョンにアップグレードします。

   注: 新しい DB2 バージョンのインストール時に新規インスタンスを作成しないでください。

4. DB2 サーバーをアップグレードした後に、リモート・ノードを定義し、BigFix サーバーでリモート・データベースをカタログします。

   ```
   CATALOG TCPIP NODE TEM_REM REMOTE {host} SERVER {port}
   CATALOG DATABASE BFENT AS BFENT AT NODE TEM_REM
   CATALOG DATABASE BESREPOR AS BESREPOR AT NODE TEM_REM
   ```

5. BigFix サーバーからデータベース接続をテストします。
6. すべての BigFix サービスを開始します。

第 24 章. サーバー監査ログ

BigFix サーバーは、サーバー監査ログ・ファイルを生成します。このファイルには、アクセス情報 (ログイン/ログアウト) と、各ユーザーがコンソールまたは WebUI を通じて実行したアクションに関する情報が含まれます。

さらに、サーバー監査ログ・ファイルでは、コンソールまたは WebUI から BigFix サーバーを介してクライアントに送信され、その後、キャンセルされた特定のアクションも追跡されます。また、Web レポートまたは BigFix REST API を使用する場合は、BigFix サーバーへのアクセス情報も記録されます。

監査ログ・メッセージのフォーマット

監査項目は、単一の行で表示され、フィールド区切り文字の同じ番号が含まれます。
フィールド区切り文字は、その特定フィールドに値が存在しない場合でも表示されます。
監査フィールドのフォーマットは、将来変更される場合があるため、各行には、最初の項目としてバージョン番号が含まれます。

BigFix サーバー上の監査ログのデフォルトの場所を以下に示します。

- Windows コンピューター: `%ProgramFiles(x86)%\BigFix Enterprise\BES Server\server_audit.log`
- Linux コンピューター: `/var/opt/BESServer/server_audit.log`

BigFix 管理ツール上の監査ログのデフォルトの場所を以下に示します。

- Windows コンピューター:
  - BigFix 管理ツールを実行する各ユーザーの場合:

    `C:\Users\<USERNAME>\AppData\Local\BigFix\besadmin_audit.log`

    例:

    `C:\Users\Administrator\AppData\Local\BigFix\besadmin_audit.log`

    - BigFix 管理ツール (BESAdmin) が Fixlet によって呼び出される場合 (または LocalSystem ユーザーによって実行される場合):

      `C:\Windows\System32\config\systemprofile\AppData\Local\BigFix\besadmin_audit.log`

- Linux コンピューター:

  `/var/log/besadmin_audit.log`

BigFix バージョン 9.5.11 以降、監査ログ・メッセージは次のフォーマットです。

```
<format-version>|<timestamp>|<message-priority>|<username>|<event-source>|<event-label>|<event-type>|<ip-address>|<message>
```

“|” がフィールド分離文字です。

- `format-version`: メッセージ・フォーマットのバージョン。例えば、*1* です。
- `timestamp`: ログ・メッセージのタイム・スタンプ。サーバー・タイム・ゾーンまたは UTC を使用できます。
- `message-priority`: ログの優先度。
    - EMERG (緊急、システムが機能していない、または使用できない)
    - ERROR (エラー状態)
    - WARN (警告)
    - INFO (情報メッセージ)
- `username`: イベント・イニシエーターのユーザー名。ユーザー・イベントではない場合、このフィールドは *SYSTEM* に設定されます。
- `event-source`: イベントの発生源。可能な値:*CONSOLE*、*RESTAPI* , *WEBUI* , *WEBREPORTS*。
- `event-label`: 影響を受けるイベントまたは成果物。

  可能な値:USER、SITE、ACTION、ROLE、COMPUTER , AUTHZ, SETTING , DATABASE。
- `event-type`: イベントのタイプ。

  可能な値:CREATE、DELETE、UPDATE、LOGIN , LOGOUT , SEARCH。

- `ip-address`: イベント要求を開始したコンポーネントの IP アドレス。SYSTEM の場合、これはサーバーの IP アドレスです。
- `message`: 実際のログ・メッセージ。

BigFix バージョン 9.5.11 以降、サーバー監査ログには次の項目も含まれます。

- コンソールから、または API を通じてコンピューターが削除されたことについてのメッセージ
- アクションの削除に関するメッセージ

例

以下は、新しいフォーマットのログ・メッセージの例です。

```
1|Tue, 05 Sep 2017 10:57:06
 -0700|INFO|johndoe|CONSOLE|AUTHZ|LOGIN|172.28.128.5|user “johndoe “
(1):Successful log in. (Data Connection)
```

```
1|Tue, 05 Sep 2017 10:58:32
 -0700|INFO|johndoe|CONSOLE|ACTION|DELETE|172.28.128.5|
Action waitOverrideTest(50) was deleted
```

9.5.11 以降で導入された監査項目以外の監査項目の場合、メッセージのフォーマットは次のようになります。`<format-version>|<timestamp>|<message-priority>||||||<message>`. 例:

`1|Tue, 05 Sep 2017 10:57:06 -0700|INFO||||||user "johndoe" (1): Successful log in. (Data Connection)`

ログの管理

監査ログ・ファイルのデフォルト・サイズは 100 MB です。サイズが最大値に到達すると、ログ・ファイルは名前が変更され、新しいファイルが作成されます。名前が変更されたログ・ファイルが削除されることはありません。スペースを有効に使用するために、ログ・ファイルを別の場所に移動するか、定期的にパージする必要があります。

以下の設定を使用することで、この値を変更できます。

- BigFix サーバー上の *_Audit_Logging_LogMaxSize*。
- BigFix バージョン 10.0.8 で導入され、BigFix 管理ツール上にある *_BESAdminAudit_Logging_LogMaxSize*。

この _BESAdminAudit_Logging_LogDirectoryPath 設定を使用して、BigFix 管理ツールの監査ログ・パスを変更することもできますが、BigFix サーバーでは変更できません。

詳しくは、「ロギング (（ページ）　　)」および「BigFix ロギング・ガイド」を参照してください。

注: バージョン 9.5.11 にアップグレードすると、`server_audit.log` ファイルが強制的に `server_audit.YYYYMMDDHHMM` にローテンションします。これは 1 回限りのアクションで、ログ・ローテーションを構成している場合も、構成していない場合も適用されます。`server_audit.YYYYMMDDHHMM` ファイルには、古いフォーマットの監査ログのみ含まれ、一方 `server_audit.log` には新しいフォーマットの監査ログのみ含まれます。

注: バージョン 10.0.1 にアップグレードするときに、ユーザー名またはメッセージの内容に文字「|」 (パイプ) を挿入すると、文字は「%7C」に置き換えられ、ログ・ファイルの形式が正しくないことを回避するために自動ツールがログ・ファイルを解析します。

第 25 章. 詳細オプションのリスト

次のリストは、詳細オプションを示しています。

詳細オプションは、Windows システムの場合は、BigFix 管理ツールの「詳細オプション」タブで指定し、Linux システムの場合は、`BESAdmin.sh` コマンドで指定します。いずれの場合も、以下の構文を使用します。

```
./BESAdmin.sh -setadvancedoptions -sitePvkLocation=<path+license.pvk>
[-sitePvkPassword=<password>]
{ -list | -display
| [ -f ] -delete option_name
| [ -f ] -update option_name=option_value }
```

注: コマンド構文で使用される表記 `<path+license.pvk>` は、*`path_to_license_file`*`/license.pvk` を表します。

以下に示す各オプションは、通常は HCL ソフトウェア・サポートによって指定されます。

機能を無効にするための詳細オプション

これらのオプションは、コンソールの特定の機能を無効にする場合に使用します。

disableNmoSiteManagementDialog

「1」に設定した場合、マスター以外のオペレーター (NMO) は、サイト管理ダイアログを使用できなくなります。

disableNmoComments

「1」に設定した場合、NMO は、コメントを追加できません。NMO は、これまでどおり、コメントを表示することはできます。

disableNmoManualGroups

「1」に設定した場合、NMO は、マニュアル・グループのコンピューターの追加や削除ができなくなり、NMO のいずれのコンピューターもメンバーではないマニュアル・グループを表示できなくなります。

disableGlobalRelayVisibility

「1」に設定した場合、NMO は、コンソールのリレー選択ドロップダウンで、NMO に属さないリレーを表示できなくなります。NMO によって管理されていないリレーに報告するよう現在構成されているマシンを NMO が表示する場合は例外で、この場合はそのリレーもリストに表示されます。

disableNmoRelaySelModeChanges

「1」に設定した場合、NMO は、自動リレー選択のオンとオフを切り替えできなくなります。

disableDebugDialog

「1」に設定した場合、キーボード・シーケンス CTRL-ALT-SHIFT-D を使用してコンソールのデバッグ・ダイアログを開けなくなります。

disableComputerNameTargeting

「1」に設定した場合、「アクションの実行」ダイアログの「対象」タブで 3 番目のラジオ・オプション「コンピューター名のリストによる対象指定 (target by list of computer names)」が削除されます。

allowOfferCreation

「0」に設定した場合、「アクションの実行」ダイアログの「提案」タブが無効になります。Fixlet の提案プリセットはコンソールで無視されます。

disableNmoCustomSiteSubscribe

「1」に設定した場合、すべての NMO に対して「カスタム・サイト・サブスクリプションの変更」メニュー項目が無効になります。

パスワード・ポリシーの詳細オプション

これらの設定は、BigFix 環境でパスワード・ポリシーを適用するために使用します。

passwordComplexityRegex

オペレーター・パスワードを選択または変更するときには、パスワードの複雑性の要件として使用する *perl* 形式 の正規表現を指定します。以下に例を示します。

- 文字列「bigfix」と等しくない、6 文字以上のパスワードが必要。

```
(?![bB][iI][gG][fF][iI][xX]).{6,}
```

- 小文字、大文字、および句読点を含む 6 文字以上のパスワードが必要。

```
(?=.*[[:lower:]])(?=.*[[:upper:]])(?=.*[[:punct:]]).{6,}
```

- 小文字、大文字、句読点、および数値の 4 つの文字クラスのうちの 3 つを含む、8 文字以上のパスワードが必要。

```
((?=.*[[:lower:]])(?=.*[[:upper:]])(?=.*[[:punct:]])|
(?=.*[[:lower:]])(?=.*[[:upper:]])(?=.*[[:digit:]])|
(?=.*[[:lower:]])(?=.*[[:digit:]])(?=.*[[:punct:]])|
(?=.*[[:digit:]])(?=.*[[:upper:]])(?=.*[[:punct:]])).{8,}
```

注: サイト管理者パスワードは、この複雑性の要件の影響を受けません。

passwordComplexityDescription

パスワードの複雑性の要件の説明を指定します。この文字列は、パスワードの選択が、**passwordComplexity** オプションを使用して設定された複雑性の要件を満たさない場合にユーザーに表示されます。パスワードの複雑性の説明の例として、「パスワードには、6 文字以上を入力してください」などがあります。この値を設定せずに **passwordComplexityRegex** 設定を指定した場合は、**passwordComplexityRegex** で設定した説明がユーザーに表示されます。

passwordsRemembered

ユーザー・アカウントに設定可能な固有の新規パスワードの数を指定します。この数を超えると、古いパスワードを再利用できるようになります。デフォルト値は「0」です。

このオプションは、BigFix V8.2 で導入されました。

maximumPasswordAgeDays

パスワードを使用できる日数を指定します。この日数を超えると、システムがユーザーにパスワードの変更を求めます。デフォルト値は「0」です (最大値なし)。

このオプションは、BigFix V8.2 で導入されました。

minimumPasswordLength

ユーザー・アカウントのパスワードに含めることができる最小の文字数を指定します。デフォルト値は「6」です。以下に、このオプションの使用例を示します。

```
./BESAdmin.sh -setadvancedoptions -sitePvkLocation=LOCATION
-sitePvkPassword=PASSWORD -update minimumPasswordLenth=9
```

このオプションは、BigFix V8.2 で導入されました。

enforcePasswordComplexity

「1」または「true」に設定した場合、パスワードは、以下の最小要件を満たす必要があります。

- ユーザーのアカウント名、およびユーザーのフルネームの一部 (3 文字以上の連続する部分) を含めることはできません。
- 6 文字以上にする必要があります。
- 以下の 4 つのカテゴリーのうちの 3 つからの文字が含まれている必要があります。

```
English uppercase characters (A through Z)
English lowercase characters (a through z)
Base 10 digits (0 through 9)
Non-alphabetic characters (for example, !, $, #, %)
```

minimumPasswordLength 設定も指定した場合、有効なパスワード最小長は、6 と **minimumPasswordLength** の値のうち、どちらか大きい方の値になります。

複雑性の要件は、パスワードが変更または作成されるときに適用されます。デフォルト値は「0」です。

このオプションは、BigFix V8.2 で導入されました。

accountLockoutThreshold

あるユーザー名についての正しくないログオン試行の回数を指定します。この数を超えると、アカウントが **accountLockoutDurationSeconds** 秒の間ロックされます。デフォルト値は「5」です。

このオプションは、BigFix V8.2 で導入されました。

accountLockoutDurationSeconds

ログオン試行が **accountLockoutThreshold** 回失敗した後にアカウントがロックされる秒数を指定します。デフォルト値は「1800」です。

このオプションは、BigFix V8.2 で導入されました。

注: Web レポートにも類似のパスワード制御がありますが、こちらは別途設定する必要があります (「ユーザー」->「ユーザー・オプション」)。

対象指定制限の詳細オプション

これらの拡張オプションを使用して、対象指定制限をグローバルに指定します。特定のユーザーに対して設定する場合は、これらの設定をハイブ `HKEY_CURRENT_USER\Software\BigFix\Enterprise Console\Targeting` の下の BigFix コンソール・コンピューターのレジストリー・キーに DWORD として追加します。

以下の表にリストされているオプションは、対応するレジストリー・キーがコンソールで設定されていないか、キーがデフォルト値に設定されている場合にのみ有効になります。

targetBySpecificListLimit

個々の選択で対象として設定できるコンピューターの最大数を指定します。デフォルト値は 10000 です。

targetBySpecificListWarning

個々の選択で対象として設定できるコンピューター数のしきい値を指定します。この数を超えると、コンソールに警告メッセージが表示されます。デフォルト値は 1000 です。

targetByListSizeLimit

コンピューター名のテキスト・リストによって対象を設定する場合に、指定できる最大バイト数を指定します。デフォルト値は 100000 です。

拡張オプションの名前と関連するレジストリー設定の名前との対応は、以下のとおりです。

```
targetBySpecificListLimit => SpecificListLimit
targetBySpecificListWarning => SpecificListWarning
targetByListSizeLimit => ByListSizeLimit
```

以下の例では、SpecificListLimit 設定 (targetBySpecificListLimit 拡張オプションに対応) を 9000 = 0x2328 に制限します。

```
{[HKEY_CURRENT_USER\Software\BigFix\Enterprise Console\Targeting]
"SpecificListLimit"=dword:00002328}
```

注: デフォルト値は増やさないでください。

認証の詳細オプション

これらの設定は、コンソールに対するユーザー認証を管理するために使用します。

loginTimeoutSeconds

ある特定のアクションを実行するためにコンソールでユーザーの再認証が必要になるまでのアイドル時間 (秒単位) を指定します。ユーザーが認証するか、アイドル時間のしきい値までの間に認証が必要なアクションをユーザーが実行するたびに、タイマーがリセットされます。V8.2 より前の適用環境からのアップグレードの場合、デフォルト値はゼロです。V8.2 以降の新規インストールの場合、デフォルト値は無限大です。

loginWarningBanner

コンソールまたは Web レポートにログインした後にすべてのユーザーに表示されるテキストを指定します。ユーザーが次に進むには **「OK」** をクリックする必要があります。以下に、このオプションの使用例を示します。

```
./BESAdmin.sh -setadvancedoptions
 -sitePvkLocation=/root/backup/license.pvk
-sitePvkPassword=pippo000 -update loginWarningBanner='new messag
e'
```

このオプションは、BigFix V9.1 で導入されました。

timeoutLockMinutes

コンソールで再認証が必要になるまでのアイドル時間の経過分数を指定します。この設定は、**loginTimeoutSeconds** とは異なります。**timeoutLockMinutes** では、コンソール全体が非表示になり、他のユーザーはコンソールを表示することも使用することもできなくなります。アイドル時間は、キー・ボタン、マウス・クリック、マウス移動を含め、セッションにいかなる種類の入力も行われない状態を指します。

オペレーターが Windows セッションの資格情報を使用してアクセスした場合 (Windows 認証)、このオプションはコンソール上で有効になりません。

このオプションは、BigFix V9.1 で導入されました。

timeoutLogoutMinutes

コンソールが閉じるまでのアイドル時間の経過分数を指定します。この設定は、**timeoutLogoutMinutes** によってコンソールが完全に閉じられるため、**loginTimeoutSeconds** や **timeoutLockMinutes** とは異なります。アイドル時間は、キー・ボタン、マウス・クリック、マウス移動を含め、セッションにいかなる種類の入力も行われない状態を指します。

このオプションは、BigFix V9.5.11 で導入されました。

注: 非効率的な MIME 拡張オプションは、BigFix V9.5 サーバーではサポートされなくなりました。既存のアクションは引き続きクライアントで実行されますが、サーバーは非効率的な MIME アクションを生成することはできません。

コンピューターの削除をカスタマイズするための詳細オプション

デフォルトでは、非アクティブなコンピューターが BigFix によって自動的に管理されることはありません。非アクティブなコンピューターのエントリーをコンピューター・リスト・ビューから削除して削除済みのマークを付けない限り、そのコンピューターは引き続きコンソール・ビューに表示されます。また、非アクティブなコンピューターのデータはデータベース内に残るため、不要なデータによってテーブルが占有されることになります。

この動作を変更するには、非アクティブなコンピューターを削除済みとしてマークする詳細オプションを指定します。これにより、非アクティブなコンピューターがコンソール・ビューに表示されなくなり、そのコンピューターのデータが BigFix データベースから削除されます。

この方法により、指定された日数内に BigFix サーバーに対して応答したコンピューターだけがコンソール・ビューに表示されるようになります。また、ディスク・スペースが解放されるため、データベースの実行速度も上がります。

コンソールからコンピューターを自動的に削除し、そのコンピューターのデータをデータベースから削除するには、以下のオプションを使用します。

inactiveComputerDeletionDays

コンピューターに削除済みのマークを付けるまでの連続した日数を指定します。この日数が経過しても BigFix サーバーに応答しなかったコンピューターが、削除済みとしてマークされます。このコンピューターが再び応答を返した場合は、削除済みのマークが解除され、このコンピューターのエントリーがコンソール・ビューに表示されます。このオプションのデフォルト値は **0** です。この場合、非アクティブなコンピューターが自動的に削除済みとしてマークされることはありません。

inactiveComputerPurgeDays

コンピューターのデータを BigFix データベースから削除するまでの連続した日数を指定します。この日数が経過しても BigFix サーバーに応答しなかったコンピューターのデータが、データベースから削除されます。このコンピューターが再び応答を返した場合、システムはこのコンピューターに対し

て、データをデータベースに復元するための完全な更新情報を送信するように要求します。その後、このコンピューターの削除済みマークが解除されます。このオプションのデフォルト値は **0** です。この場合、削除済みのマークが付いたコンピューターのデータがデータベースから自動的に削除されることはありません。

inactiveComputerPurgeBatchSize

BigFix は、**inactiveComputerPurgeDays** で指定された日数が経過したコンピューターのデータをデータベースから削除する内部タスクを毎日実行します。このタスクにより、バッファー内のコンピューター・データ (コンピューターのホスト名など) が削除されるため、不要なデータがデータベースにロードされることはありません。**inactiveComputerPurgeBatchSize** の値により、各バッファー内のデータベースでクリーンアップされるコンピューターの数を指定します。このオプションのデフォルト値は **1000** です。コンピューターが再度レポートを返した場合、そのコンピューター ID を使用して、データベース内のそのエントリーとの突き合わせが行われます。

注: 0 以外の値を **inactiveComputerPurgeDays** に設定した場合は、**inactiveComputerPurgeBatchSize** オプションを指定してください。

BigFix Query をカスタマイズするための詳細オプション

オプションで、いくつかのパラメーターを設定して、BigFix Query 機能をカスタマイズできます。

BigFix Query 要求とその結果を格納するために、データベースで使用できるスペースを多く使用するのを防止するため、次の詳細オプションを、BigFix サーバーの管理ツールでカスタマイズできます。

queryHoursToLive

BigFix Query 要求をデータベースに保持する時間数を決定します。このオプションのデフォルト値は **1440** (60 日間) です。有効な値は 0 から 8760 (1 年間) です。

queryResultsHoursToLive

BigFix Query 結果をデータベースに保持する時間数を決定します。デフォルトの値は **4** 時間で、1 から 336 (2 週間) の範囲が有効です。この範囲外の数値を入力した場合は、デフォルトの値が使用されます。

queryPurgeBatchSize

queryHoursToLive または **queryResultsHoursToLive** が経過した要求や結果を表すデータベース内のエントリーは、バッファー内のデータベースから削除されます。この詳細オプションは、これらのバッファーそれぞれに含まれるデータベース・エントリーの数を決定します。このオプションのデフォルト値は **100000** バイト (100 KB) です。

BigFix Query 機能をカスタマイズするために使用できるその他の構成設定は次のとおりです。

queryPerformanceDataPath

BigFix Query に実行時の、FillDB とサーバーとの対話に関するパフォーマンス情報を格納するログ・ファイルのパスを定義します。このオプションのデフォルト値は *none* です。

Enterprise Server BigFix Query_MaxTargetsForGroups

グループごとに対象が設定される BigFix Query 要求をアドレス指定できる最大対象数を決定します。対象の数が指定した値を超えた場合、BigFix Query 要求はすべてのクライアントに送信され、それぞれのクライアントが、対象グループのメンバーであるかどうかを判別します。対象の数が指定した値を超えない場合、BigFix Query 要求はグループのメンバーのみに送信されます。この設定は、BigFix コンソールで、「コンピューター」リストからサーバーを選択し、「設定の編集」をクリックすることで構成できます。このオプションのデフォルト値は **100** です。

その他の詳細オプション

これらのオプションは、BigFix 環境の他の側面をカスタマイズするために使用します。

automaticBackupLocation

`root` およびデータベース・インスタンス所有者 (デフォルトでは `db2inst1`) の両方がアクセスできる既存のパスに設定された場合、このオプションによって、BigFix サーバーが、アップグレード・プロセスの実行の前後に BFENT データベースおよび BESREPOR データベースのバックアップを自動的に実行できるようになります。

このオプションは、Linux BigFix サーバー V9.5.3 以降のみで使用可能です。

詳しくは、『アップグレード時の自動データベース・バックアップ (（ページ）　　)』を参照してください。

clientIdentityMatch

この拡張オプションを使用することにより、BigFix サーバーでエンドポイントが複製の可能性があることが検出されたときに、コンピューター・エントリーの重複を避けることができます。BigFix サーバーは、既存のコンピューター情報を使用してクライアントの ID の照合を試行し、ロールバックまたはリストアされた可能性があるコンピューターに同じ `ComputerID` を再割り当てすることができます。

clientIdentityMatch=0 の場合、BigFix サーバーは厳密な複製検出を実行します。このことは、BigFix サーバーがロールバックまたはリストアされたクライアントから登録要求を受信した場合、サーバーは旧 `ComputerID` を無効にして、古いクライアント定義をリセットし、新しい `ComputerID` を登録クライアントに割り当てることを意味します。これはデフォルト動作であり、V9.5.7 より前の BigFix サーバーの動作方法と同じです。

clientIdentityMatch=100 の場合は、BigFix サーバーは、新しい `ComputerID` を登録クライアントに割り当てる前に追加の検査を実行して、複製されたコンピューター・エントリーが作成されないようにします。このことは、BigFix サーバーが、ロールバックされたクライアントに関する情報がその `ComputerID` に保持されているデータと十分に一致するかどうかの判断を

行うことを意味します。クライアントの ID が一致すると、クライアントは旧 `ComputerID` を引き続き使用し、その ID はリセットされません。

詳しくは、『クライアントのリストア時の重複回避 (（ページ）　　)』を参照してください。

includeSFIDsInBaselineActions

「1」に設定した場合、ベースライン・アクションが発行されたときにコンソールがソース Fixlet ID を含むことが必須になります。これらの ID を発行することは、5.1 クライアントと互換性がありません。

defaultHiddenFixletSiteIDs

このオプションを使用すると、デフォルトの Fixlet 表示をサイトごとに個々に選択して変更できます。このオプションは、グローバルなデフォルトの Fixlet の非表示が使用されていない場合にのみ有効です。デフォルトで非表示にするすべてのサイト ID をコンマ区切りリストで指定します。サイト ID のリストは、データベースの SITENAMEMAP 表にあります。

defaultOperatorRolePermissions

このオプションを使用すると、オペレーターおよびロールの作成時に適用されるデフォルト権限を変更できます。以下の値にすることができます。

- 0: オペレーターおよびロールは、BigFix V9.5.10 まで適用されていたデフォルト権限で作成されます。
- 1: オペレーターおよびロールは、最小のデフォルト権限で作成されます。値を設定しなかった場合でも、同じデフォルト設定が適用されます。
- 2: オペレーターおよびロールは、前の場合と同様に、最小のデフォルト権限で作成されます。ただし、**「他のオペレーターのアクションの表示」**が**「はい」**に設定されており、**「非管理資産」**が**「スキャン・ポイント別」** (オペレーター向け) に設定されている場合を除きます。ただし、ロールの場合は**「非管理資産」**は常に**「すべて非表示」**に設定されます。オペレーター向けの**「アクセス制限」**は、**「このユーザーのログ**

インを常に許可する」に設定されます。ログイン権限**「コンソールを使用できます」**は、オペレーターとロールの両方で**「はい」**に設定されます。

このオプションは、BigFix V9.5.11 で導入されました。

enableRESTAPIOperatorID

このオプションを使用すると、オペレーター名ではなく、オペレーター ID とともにオペレーター・リソース URL を表示できます。たとえば、`https://BigFix_Server_URL:52311/api/operator/<Operator_ID>` です。このオプションを有効にするには、true または 1 に設定します。

このオプションは、BigFix V9.5.10 で導入されました。

showSingleActionPrePostTabs

「1」に設定した場合、単一アクションの場合でも、「アクションの実行」ダイアログの「事前アクション・スクリプト (Pre-Action Script)」タブと「ポスト・アクション・スクリプト (Post-Action Script)」タブが表示されます。

propertyNamespaceDelimiter

取得したプロパティーの分離文字を指定します。デフォルトでは、取得したプロパティーは、文字シーケンス「::」によって名前空間に分離されます。分離文字を示すために使用される文字シーケンスを、この適用オプションを使用して変更することができます。

DefaultFixletVisibility

このオプションが設定されている場合、外部サイトから収集された Fixlet、タスクおよび分析をグローバルに表示するか、グローバルに非表示にするかを指定できます。デフォルトでは、すべてのコンソール・オペレーターに対してグローバルに表示されます。

注: Windows プラットフォームの場合のみ、このオプションは BigFix 管理ツールの「システム・オプション」タブでも使用できます。

MinimumRefreshSeconds

このオプションが設定されている場合、コンソール・オペレーターが自動最新表示間隔を設定できるようになるまでの最小時間を指定できます。この時間は、秒単位で指定します。デフォルトでは、5 秒に設定されています。

注: Windows プラットフォームの場合のみ、このオプションは BigFix 管理ツールの「システム・オプション」タブでも使用できます。

minimumConsoleRequirements

コンソールの接続先のデータベースを実行するマシンが満たす必要のある最小要件を指定します。この値は、以下の 1 つ以上の要件文字列のコンマ区切りリストで構成されます。

"RAM:<min MB MO ram>/<min MB NMO ram>"

これは、コンソールが、指定された物理 RAM 以上のマシン上で実行されることを要求します。マスター・オペレーター用およびマスター以外のオペレーター用の 2 つの異なる値を指定する必要があります。値は両方とも 2^32 より小さくする必要があります。例えば、"RAM:2048/1024" のように指定します。

"ClientApproval"

マシンがログインに適しているかどうかを BES クライアントが判別する必要があることを指定します。マシンがログインに適していると見なされるのは、以下の設定のいずれかがローカルに指定されている場合です。

- **"moConsoleLoginAllowed"**
- **"nmoConsoleLoginAllowed"**

"ClientApproval" オプションを使用しているときにログインするには、コンソールは HKEY_LOCAL_MACHINE の下に格納されたクライアント・レジストリー・キーの読み取り権限を持つアカウントとして実行されることが必要です。

このオプションは、BigFix V6.0.12 で導入されました。

actionSiteDBQueryTimeoutSecs

コンソールが照会を停止し (読み取りロックを解放して任意のデータベース・ライターの書き込みを許可するため)、その後、停止した場所から照会を再開するまで、どのくらいの期間にわたってアクション・サイト・データベース照会の実行が許可されるかを指定します。設定しない場合、デフォルト値は 60 秒です。「0」に設定した場合、アクション・サイト・データベース照会は、タイムアウトになりません。

このオプションは、BigFix V6.0.17 で導入されました。

usePre70ClientCompatibleMIME

「true」に設定した場合、コンソールは、7.0 より前のクライアントが理解できるアクション MIME 文書を作成できます。デフォルトでは、アップグレードの場合は「true」、フレッシュ・インストールの場合は「false」に設定されます。

このオプションは、BigFix V7.0 で導入されました。

disableRunningMessageTextLimit

「0」以外の値に設定した場合、コンソール・ユーザーは、「アクションの実行」ダイアログの実行中のメッセージ・テキストに 255 文字よりも多く入力できます。

このオプションは、BigFix V7.0.7 で導入されました。

useFourEyesAuthentication

「true」に設定した場合、コンソール・ユーザー・ドキュメントのユーザー・アクションに対して承認者を設定できます。承認者は、ユーザーがログオンしたのと同じコンソール上のアクションを確認する必要があります。

このオプションは、BigFix V8.2 で導入されました。

masterDatabaseServerID

デフォルトでは、サーバー ID が 0 のデータベースがマスター・データベースです。これは、BESAdmin が接続する必要のあるデータベースです。このオプションは、マスター・データベースを別のマシンに変更する場合に使用します。

このオプションは、BigFix V7.0 で導入されました。

enableWakeOnLAN

「1」に設定した場合、コンソールはコンピューター・リストに「右クリック: WakeOnLAN」機能を表示します。デフォルトでは、この機能は表示されません。

このオプションは、BigFix V7.1 で導入されました。

enableWakeDeepSleep

「1」に設定した場合、コンソールはコンピューター・リストに「右クリック: BES クライアント・アラート要求の送信」機能を表示します。デフォルトでは、この機能は表示されません。ディープ・スリープ中は、この特定のウェイクアップ・メッセージを除くすべての UDP メッセージは無視されます。

このオプションは、BigFix V8.0 で導入されました。

requireConfirmAction

「1」に設定した場合、アクションが実行されるたびに、アクションの詳細の要約が含まれた、確認のポップアップ・ウィンドウが表示されます。ポップアップ・ウィンドウにリストされる情報は以下のとおりです。

```
Action Title
Estimated endpoints targeted
Start time
End time
Originated by or Source
```

要約には、アクションが再始動またはシャットダウンを要求する場合、それを行う必要性がリストされます。デフォルトでは確認ウィンドウは表示されません。

このオプションは、BigFix V7.1 で導入されました。

このオプションを構成したら、BigFix コンソールを再始動してください。

第 26 章. セキュリティー構成シナリオ

BigFix から、拡張セキュリティー・オプションを構成して NIST セキュリティー標準に準拠する機能を提供しています。

この設定により、デジタル署名とコンテンツ検証のハッシュ・アルゴリズムとして SHA-256 を使用できるようになります。また、BigFix コンポーネント間の TLS 1.2 通信が可能となります。

すべての BigFix コンポーネントをインストールまたはアップグレードした後でのみ、拡張セキュリティー・オプションを設定できます。

注: このオプションを設定すると、厳しく制限されたセキュリティー環境を構成することになり、製品のパフォーマンスが低下する場合があります。このセキュリティー設定は、マストヘッド・ファイルを編集することでいつでも有効または無効にすることができます。追加情報については、構成ガイド」を参照してください。

拡張セキュリティー設定に加えて、SHA-256 アルゴリズムを使用してファイル・ダウンロードの整合性を検証するためのチェックを設定できます。このオプションを設定しない場合、ファイル・ダウンロードの整合性チェックは SHA-1 アルゴリズムを使用して実行されます。このオプションは、拡張セキュリティー·オプションを設定した場合のみ設定できます。したがって、すべての BigFix コンポーネントが V10 以上の場合のみ設定できます。

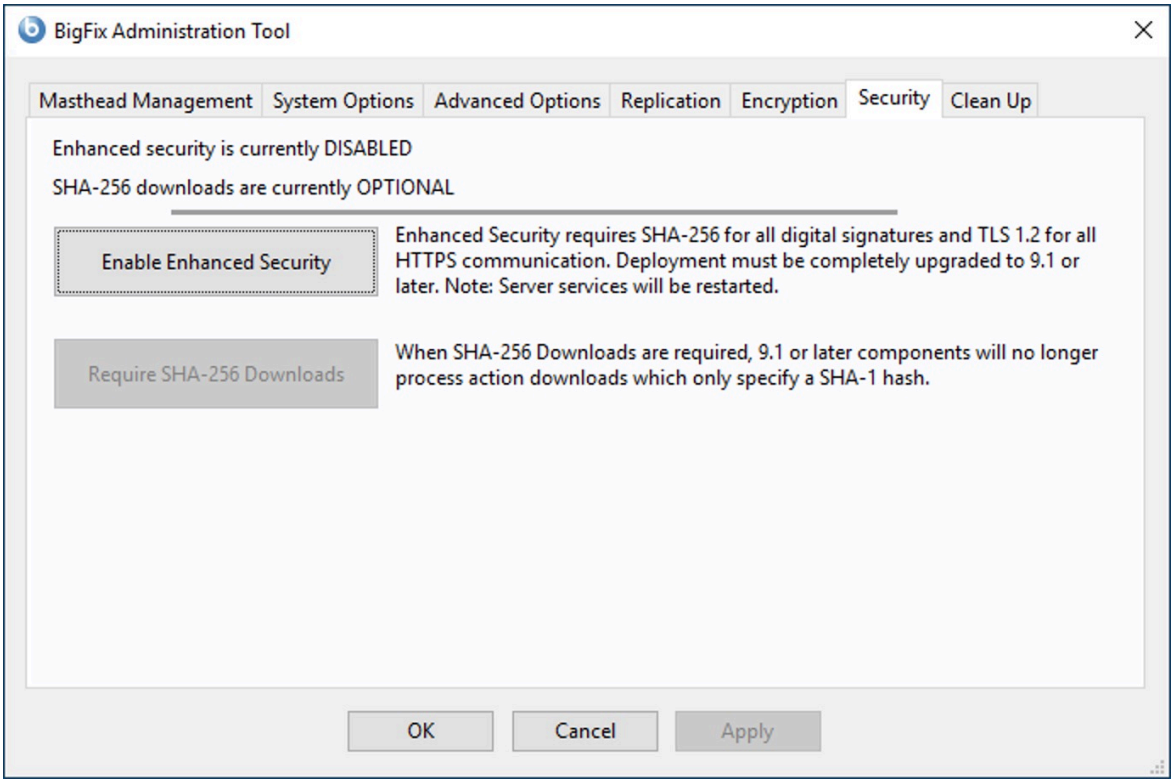
複合環境では、すべての DSA サーバーが BigFix V10 以上にアップグレードされ、新しいライセンスを取得した場合にのみ、拡張セキュリティー・オプションを有効にすることができます。

Windows システムの場合

拡張セキュリティー・オプションを設定するには、以下の手順を実行します。

1. **「スタート」>「すべてのプログラム」>「BigFix」>「BigFix 管理ツール」**をクリックして管理ツールを実行します。
2. サイト・ライセンス (`license.pvk`) のロケーションを参照し、**「OK」**をクリックします。

3. **「セキュリティー」**タブを選択します。以下のウィンドウが表示されます。



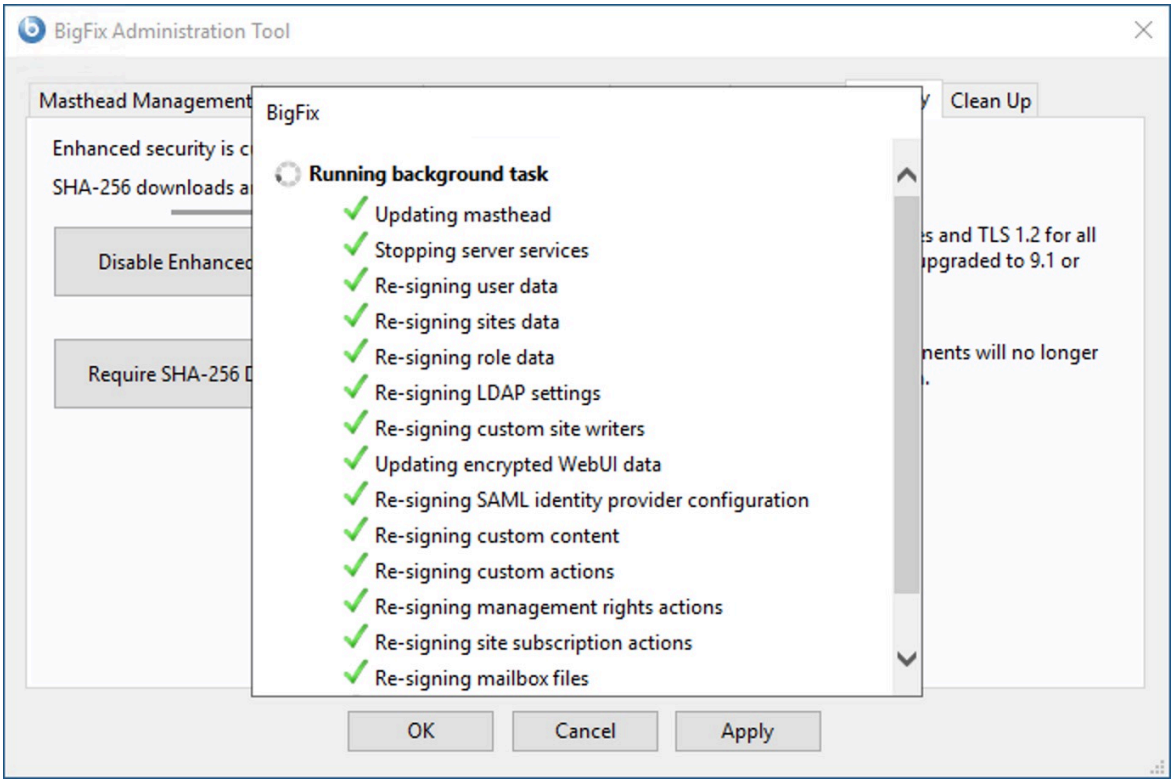
これで、拡張セキュリティー・オプションを有効にすることができます。

BigFix を旧バージョンからアップグレードしたときに、サブスクライブしていたサイトで拡張セキュリティー・オプションがサポートされている場合は、**「拡張セキュリティーをサポートしていないサイトのサブスクリプションを解除します」**が選択されません。

災害対応サーバー・アーキテクチャー (DSA) に関係するすべての BigFix サーバーがバージョン 10 になっていて、更新されたライセンスが使用されていることが確認されるまで、**「次の複製サーバー上で BESAdmin を実行してください」**チェック・ボックスにチェック・マークは付きません。

4. BigFix バージョン 10 に付属するセキュリティー機能拡張を使用する場合は、**「今すぐライセンスを収集」**をクリックします。クリックしない場合は、BigFix バージョン 9.0 によって提供されるセキュリティー動作を使用することになります。
「今すぐライセンスを収集する (Gather license now)」をクリックすると、更新されたライセンスが HCL サイトから収集され、BigFix クライアントに配布されます。こ

のステップを実行することにより、インストール・ステップ中に既存のライセンス・ファイルを指定した場合に、更新されたライセンス認証が確実に使用されます。



5. 3 つのチェック・マークが緑色の場合は、**「拡張セキュリティーを有効にする」**をクリックして、拡張セキュリティーを設定できます。
6. SHA-256 アルゴリズムを使用してダウンロードした後もデータが変更されていないようにするには、**「SHA-256 ダウンロードが必要」**をクリックします。このオプションを選択しない場合、ダウンロードされたファイルの整合性チェックは SHA-1 アルゴリズムを使用して実行されます。

注: 「拡張セキュリティーを有効にする」オプションを有効にした場合にのみ、**「SHA-256 ダウンロードが必要」**オプションを有効にすることができます。

Linux システムの場合

BigFix V10 をインストールしてから (または V10 にアップグレードしてから) セキュリティー・オプションを設定するには、スーパーユーザーとして以下のコマンドを実行します。

```
./BESAdmin.sh -securitysettings -sitePvkLocation=<path+license.pvk>
              -enableEnhancedSecurity -requireSHA256Downloads
```

注: コマンド構文で使用される表記 `<path+license.pvk>` は、*`path_to_license_file`*`/license.pvk` を表します。

`./BESAdmin.sh -securitysettings` の完全な構文は、次のとおりです。

```
./BESAdmin.sh -securitysettings -sitePvkLocation=<path+license.pvk>
   [-sitePvkPassword=<password>]
   { -status | {-enableEnhancedSecurity|-disableEnhancedSecurity}
   | {-requireSHA256Downloads|-allowSHA1Downloads} }
```

各部の意味は以下のとおりです。

状況

BigFix 環境のセキュリティー設定のステータスを示します。
例:

```
BESAdmin.sh -securitysettings -sitePvkLocation=/root/backup/lice
nse.pvk
-sitePvkPassword=mypassw0rd -status

Enhanced security is currently ENABLED
SHA-256 downloads are currently OPTIONAL
```

enableEnhancedSecurity | disableEnhancedSecurity

すべてのデジタル署名とコンテンツ検証に SHA-256 暗号ダイジェスト・アルゴリズムが採用され、BigFix コンポーネント間通信に TLS 1.2 プロトコルが採用される、拡張セキュリティーを有効または無効にします。

requireSHA256Downloads

SHA-256 アルゴリズムを使用してダウンロードした後もデータが変更されていないようにします。

注: requireSHA256Downloads の設定は、**enableEnhancedSecurity** も設定した場合にのみ可能です。

allowSHA1Downloads

必ず SHA-1 アルゴリズムを使用してファイル・ダウンロードの整合性チェックが実行されるようにします。

第 27 章. クライアント認証

最新の業界標準に準拠するため、製品バージョン 10.0.7 以降での BigFix エージェントのクライアント証明書の有効期間は 13 カ月となります。

以下のセクションで説明するいくつかの特定のケースを除き、13 カ月のクライアント証明書とその後の管理への移行は自動的に行われます。手動での対応は必要ありません。

クライアント証明書の有効期間

製品バージョン 10.0.6 までは、 BigFix エージェントには 10 年の有効期間を備えたクライアント証明書が付属します。

最新の業界標準では、製品バージョン 10.0.7 以降で最大 13 カ月の SSL/TLS 証明書が提供されるため、 BigFix エージェントは 13 カ月の存続期間標準に準拠するようクライアント証明書を自動的に更新します。

BigFix エージェントでの 13 カ月のクライアント証明書への自動移行

最初の登録試行でバージョン 10.0.7 にアップグレードされた後、エージェントが接続されている BigFix サーバーまでのリレー・チェーンが完全に 10.0.7 レベル (またはそれ以降) である場合、BigFix エージェントは自律的に 10 年のクライアント証明書から 13 カ月の証明書に切り替えます。

この条件が満たされない限り、 BigFix エージェントは 10 年の証明書を保持し、この制限を含めずに必要に応じて使用し続けます。あるいは、後で完全に 10.0.7 レベル (またはそれ以降) のリレー・チェーンを介して登録を実行できる場合、BigFix エージェントによる 13 カ月の証明書に自律的に切り替える機能は損なわれます。

BigFix エージェント 10.0.7 が 10 年のクライアント証明書から 13 カ月の証明書に切り替えると、次の 2 行が標準クライアント・ログ・ファイル (YYYYMMDD.log) に記録されます。

```
The current Client certificate validity (3650 days) does not match the
 value
specified in the masthead (398 days), starting the certificate update
 process now.
Completed Client certificate update.
```

BigFix エージェントでの 13 カ月のクライアント証明書の自動保守

13 カ月のクライアント証明書に切り替えると、BigFix エージェントは現在の証明書の有効期限が近づいた際に証明書の更新を要求して、自律的に最新の状態を維持します。これは通常、シャットダウン、障害、または予期しないイベントの可能性がある期間に対処するのに十分な期間を持つため、有効期限の **45 日前**に発生します。

10.0.7 レベルの BigFix エージェントが 13 カ月のクライアント証明書の更新を取得すると、次の 2 行が標準クライアントログファイル (YYYYMMDD.log) に記録されます。

```
Client Certificate expires in N days, HH:MM:SS, refreshing it now.
Completed Client certificate update.
```

BigFix エージェントが有効期限が切れる前に証明書を更新できない場合、認証リレーに接続されているのであれば、手動の手順を実行して、BigFix デプロイメントへの再接続を許可する必要があります。手順は 期限切れのクライアント証明書から復旧する方法 (（ページ） 474) で説明されています。

クライアント証明書のステータスを監視する方法

デプロイメントの BigFix エージェントにおけるクライアント証明書の状況を監視するには、BES サポートの**クライアント証明書情報**分析を有効にします。BigFix エージェントごとに、分析では以下の情報が提供されます。

- **クライアント証明書の有効期限**: クライアント証明書の有効期限。
- **クライアント証明書の全体的な妥当性**: クライアント証明書の全体的な妥当性。
- **クライアント証明書の有効期限**: クライアント証明書の残りの有効期間。

期限切れのクライアント証明書から復旧する方法

BigFix エージェントのクライアント証明書の期限が切れており、到達できるのがネットワーク上の認証リレーのみの場合、BigFix エージェントに対して以下のコマンドを手動で実行すると、エージェントは認証リレー経由で更新された証明書を取得できます。

BESClient -update-certificate <password> http://<relay>:52311

このコマンドには、認証リレーが更新要求を上方に転送する前に確認するパスワードが含まれています。認証リレーは 10.0.7 レベル (またはそれ以降) である必要があります。

認証リレーのパスワードは、以下のように構成できます。

- リレーで定義される、クライアント設定 **_BESRelay_Comm_KeyExchangePassword** 経由で設定された単一パスワード。
- **ManualUpdateCertificatePasswords** という名前のファイルに保管され、リレーのストレージ・ディレクトリー内で保存される、ワンタイム・パスワードの改行区切りリスト (HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\BigFix\Enterprise Server\GlobalOptions の値 **StoragePath**)。

注: パスワードは ASCII 文字のみを使用する必要があります。

注: 手動コマンドの一部として平文パスワードを指定したくない場合は、次の代替構文により、パスワードを表示せずに入力するよう求められます。

- **Windows**: cmd /c BESClient.exe -update-certificate http://<relay>:52311
- **Linux**: BESClient -update-certificate http://<relay>:52311
- **Mac**: BESAgent -update-certificate http://<relay>:52311

クライアント証明書の更新を強制する方法

次の新しい actionscript コマンドを使用するアクションをターゲットに設定すると、任意の時点でクライアント証明書を更新するよう 10.0.7 レベルの BigFix エージェントに強制できます。

client certificate refresh

BigFixエージェントはクライアント証明書を自律的に保守できるため、通常の条件下BigFixでは、オペレーターはこのコマンドを使用する必要はありません。ただし、このコマンドが役立つ場合があります。例えば、45 日の証明書更新ウィンドウが証明書の有効期限を保証するのに適していない特定の状況に対処するため、 BigFix オペレーターが証明書の更新を予想する場合などです。

注: このコマンドはリレー・チェーンのバージョンに関係なく、証明書の更新を要求するよう BigFix エージェントに強制します。その結果、リレー・チェーンが完全に 10.0.7 レベル (またはそれ以降) ではない場合、このコマンドにより BigFix エージェントは 10 年の有効期間を有する更新済みの証明書を取得します。この場合、エージェントは BigFix エージェントでの 13 カ月のクライアント証明書への自動移行 (（ページ） *473)* で 説明されているロジックの適用に再度切り替えます。このロジックにより、10 年の証明書から 13 カ月の証明書への移行が可能になります。

第 28 章. クライアント認証

クライアント認証 (バージョン 9 で導入) は、BigFix で使用されるセキュリティー・モデルを拡張して、信頼できるクライアント・レポートおよびプライベート・メッセージを実現します。

この機能には後方互換性がなく、バージョン 9.0 より前のクライアントは、認証を行うリレーまたはサーバーとは通信できません。

注: クライアント認証機能の一部のセキュリティー・オプションは、**minimumSupportedClient** サービスおよび **minimumSupportedRelay** サービスを設定することでも定義できます。詳しくは、Windows システムの場合は追加の管理コマンド (（ページ）)、Linux システムの場合はBigFix 管理ツールの実行 (（ページ）)を参照してください。

オリジナルのセキュリティー・モデルには、以下の 2 つの中心となる機能があります。

- **クライアントはサーバーからのコンテンツを信頼する。**クライアントが受信するすべてのコマンドと質問は、 クライアント上にインストールされた公開鍵に対して検証されるキーによって署名されます。
- **クライアントはプライベート・レポートをサーバーに送信できます。**クライアントはサーバーに送信するレポートの暗号化を選択でき、 レポートに含まれた内容を攻撃者が解釈できないようにすることができます。この機能はデフォルトでは無効になっており、 設定を使用してオンに切り替えます。

クライアント認証では、セキュリティー・モデルを拡張して、以上の 2 つの機能の次のようなミラー・イメージを提供します。

- **サーバーはクライアントからのレポートを信頼できます (否認防止)。**クライアントはサーバーに送信するレポートをすべて署名します。これにより、レポートが攻撃者からのものでないことを検証できます。
- **サーバーはクライアントにプライベート・データを送信できる (メールボックス)。**サーバーは、攻撃者がデータを解釈できないようにするために、個々のクライアントに送信するデータを暗号化することができます。

認証済みリレーを使用する通信は、SSL を使用してすべての通信を暗号化する、信頼できるプライベートな両方向の通信チャネルです。ただし、非認証リレーとその子の間の通信は、その通信が暗号化されたレポートまたはメールボックス宛のアクションやファイルでない限り、暗号化されません。

このレベルのセキュリティーは、さまざまな目的に役立ちます。企業において、インターネットに接続するノード、DMZ 内、または完全には信頼できないすべてのネットワーク接続に対して認証リレーを要求するセキュリティー・ポリシーを採用している場合があります。認証を行うことにより、まだご使用のデプロイメントに参加していないクライアントがデプロイメント情報を取得するのを防ぐことができます。

認証リレー

認証として設定されていないインターネットに接続されたリレーを含む BigFix デプロイメントは、セキュリティー脅威にさらされます。

このコンテキストでのセキュリティー脅威は、リレーとコンテンツまたは操作、それらに関連付けられているダウンロード・パッケージ、または機密情報 (例えば、ソフトウェア、脆弱性情報、パスワードなど) が含まれる可能性のある**「リレー診断」**ページへの無許可アクセスを意味します。

リレーを"認証"リレーとして構成し、エージェントを認証できます。これにより、信頼されたエージェントのみが、サイト・コンテンツの収集やレポートの通知を行うことができます。DMZ にある、インターネットに接するリレー用の認証リレー構成を使用します。エージェントを認証するように構成されたリレーは、サーバーが発行および署名した TLS 証明書を鍵交換中に提示する子エージェントまたはリレーとのみ、TLS 通信を実行します。

リレーが認証として設定されている場合、ご使用環境での BigFix クライアントのみがそれに接続でき、クライアント間の通信は TLS (HTTPS) を介して発生します。また、この設定は、リレーとサーバーの診断ページへの未承認アクセスを防止します。

注: 新しいクライアントをインストールする必要があり、認証リレーにのみ到達できる場合、手動でキーを交換する必要があります。詳しくは、手動での鍵交換 (（ページ） *480)*を参照してください。

リレー認証を有効にする方法

リレーを認証リレーにアップグレードするには、次のステップを実行します。

1. BES サポート Web サイトで **BES クライアント設定を探します。リレー認証** Fixlet を有効にします。
2. Fixlet を実行してアクションが完了するのを待ちます。

また、`_BESRelay_Comm_Authenticating` 構成設定を手動で更新することにより、認証のリレーを構成できます。設定のデフォルト値は `0` で、リレー認証が無効なことを示します。認証を有効にするには、値を `1` に設定します。詳細については、「認証 ((ページ))」を参照してください。

デフォルトにより、すべてのクライアントは 6 時間ごとにその親リレーで再登録されます。既存のクライアントはそれぞれをリレーで再登録しない限りレポートを送信できません。

関連情報

BigFix - インターネットに接しているリレーを簡単にセットアップ

鍵交換の処理

登録を試みたエージェントが鍵と証明書を持っていない場合、そのエージェントは、選択したリレーとの鍵交換を自動的に実行しようとします。

そのリレーが非認証リレーである場合、エージェントはリレー・チェーンをたどってサーバーに要求を転送し、サーバーがエージェントの証明書に署名します。エージェントは、後で認証リレーへの接続時にこの証明書を使用できます。

認証リレーでは、このような鍵交換の自動操作は拒否されます。一般的なシナリオを以下に示します。

新規の BigFix 9.5 環境をデプロイした場合や、既存の BigFix 環境を 9.5 にアップグレードした場合、すべてのエージェントは、リレーとの鍵交換を自動的に実行します。管理者が、インターネットに接しているリレーを認証リレーとして構成している場合、既存のエージェントはすでに証明書を保有しており、正しく稼働します。これ以上のアクション

は不要です。新規のエージェントを認証リレーに接続した場合、そのエージェントに対して手動の鍵交換 (（ページ） 480)手順を実行しない限り、そのエージェントは稼働しません。

手動での鍵交換

エージェントが証明書を保有しておらず、インターネットを介して接続されたネットワーク上の認証リレーにのみ到達できる場合、そのエージェントに対して以下のコマンドを手動で実行すれば、エージェントは認証リレーとの鍵交換を実行できるようになります。

```
BESClient -register [<password>] http://<relay>:52311
```

クライアントは認証リレーとの鍵交換にパスワードを含めます。これにより、鍵交換をその親に転送する前にクライアントが検証されます。

パスワードを省略してコマンドを実行すると、パスワードが対話式で要求されます。Windows システムでは、cmd /c 接頭部を使用して コマンドを実行します。

認証リレーへの手動登録を行うもう 1 つの方法は、クライアント設定 `_BESClient_SecureRegistration` に値を設定することです。この値は、認証リレーへの手動登録を行うために必要なパスワードを指定します。この設定は、クライアントの起動時にのみ読み取られます。リレーは `clientsettings.cfg` 構成ファイルに指定できます。この構成ファイルについて詳しくは、Windows クライアント (（ページ）)を参照してください。

リレーではパスワードは以下のように構成できます。

- リレーに対するクライアント設定 `_BESRelay_Comm_KeyExchangePassword` の単一パスワードとして。
- リレー・ストレージ・ディレクトリー内の `KeyExchangePasswords` という名前のファイルに保管されるワンタイム・パスワードの改行区切りリストとして (`HKEY\SOFTWARE\WOW6432Node\BigFix\Enterprise Server\GlobalOptions` の値 **StoragePath**)。

注: 使用できるのは ASCII 文字のパスワードのみです。非 ASCII 文字を含むパスワードは使用できません。

クライアント証明書の取り消し

クライアントの認証後に、クライアントの有効性を疑う何らかの理由がある場合は、クライアントの証明書を取り消すことができます。

証明書を取り消すと、そのクライアントは信頼できる通信に対して認証されなくなります。クライアントはコンソールから削除され、取り消しリストが更新されてすべてのリレーにより収集されることにより、そのクライアントの鍵は認証リレーとの通信には使用できなくなります。

コンピューターを取り消すには次のようにします。

1. コンピューターのリストで、コンピューターを右クリックします。

2. ポップアップ・メニューで、**「証明書の取り消し」**をクリックします。
3. コンピューターの証明書を削除して問題がなければ、確認ダイアログで**「OK」**をクリックします。

これにより、取り消しがリレーへと通知されます。取り消された後は、そのクライアントは自身の秘密鍵を使用して認証リレーからコンテンツを収集することはできなくなります。取り消されたクライアントは、コンソールのコンピューター・リストに表示されなくなります。

取り消されたクライアントの再登録

クライアントの取り消し手順では、クライアントがコンソールから削除され、クライアントの証明書失効リストが更新されます。

クライアントは、非認証リレーに接続可能な場合は自動的に新規の証明書を取得できます。

そのようなリレーを使用できない場合は、以下の手動クリーンアップを実行してクライアントを再度登録する必要があります。

1. クライアントを停止します。
2. KeyStorage クライアント・ディレクトリーとクライアント・コンピューター ID を削除します。
3. 手動による鍵交換手順を実行します。
4. クライアントを開始します。

この手順が完了すると、クライアントは新しい証明書と新規のクライアント・コンピューター ID を取得します。

注: 認証リレーに接続されているが、ルート・サーバーからブロックされている SuSe クライアントの証明書を取り消す必要がある場合は、パスワードを使用した手動での登録を実行する前に、以下のエントリーを `besclient.conf` ファイルに必ずコピーしてください。

```
Settings\Client\__Relay_Control_Server
Settings\Client\__RelayServer1
```

手動での登録は、実際には、構成ファイル内のこれらのエントリーを自動的に削除し、再度作成しません。そのため、登録が完了した後、クライアントがその認証リレーと再度通信できるように、それらのエントリーを手動で追加する必要があります。

メール・ボックス

クライアント・メールボックスを使用すると、暗号化されたアクションを、すべてのクライアントにブロードキャストする代わりに特定のクライアントに送信できます。

これにより、クライアントはすべてのアクションに対して資格を持つ必要がなくなるため効率性が向上し、ネットワーク・トラフィックが最小化されます。その結果、以下のようになります。

- クライアントは、ターゲットとなった場合にのみ中断される。
- クライアントは、自身に関連しないアクションについて、レポート、評価、収集、およびアクション処理を実行する必要がなくなる。

メッセージは受信者ごとに特別に暗号化されるためプライバシーが保証されます。ターゲット・クライアントのみがそのメッセージを復号できます。

クライアントのメールボックスは特別なアクション・サイトとして実装され、各クライアントは自動的にそのサイトをサブスクライブします。クライアントは、 このサイトに加えてマスター・サイトおよびオペレーター・サイトでアクションをスキャンすることを知っています。

暗号化されたアクションをクライアント・メールボックスに直接送信するには、以下の手順を実行します。

1. **「アクションの実行」** ダイアログを開きます (「ツール」メニューやその他のダイアログで使用可能)。

2. **「ターゲット」**タブをクリックします。
3. **「デバイスの選択」**または**「デバイス名の入力」**をクリックします。クライアントの静的なリストを指定した場合にのみ、メールボックスが使用可能になります。動的にターゲットとなったコンピューターは暗号化されず、代わりにオープンな状態でマスター・サイトまたは特定のオペレーター・サイトに送信されます。バージョン 9.0 より前のターゲット・クライアントを選択した場合も、アクションはマスター・サイトまたはオペレーター・サイトに送信されます。
4. **「OK」**をクリックします。コンピューター ID またはコンピューター名によりターゲット設定されたアクションが、暗号化されてクライアント・メールボックスに送信されます。

アクションをデプロイするオペレーターの ID がアクションとともに含まれています。アクションを実行する前に、クライアントはまず現在そのオペレーターにより管理されているかどうかを判別します。そうでない場合、クライアントはアクションの実行を拒否します。

第 29 章. メンテナンスおよびトラブルシューティング

Windows サイトのパッチをサブスクライブすると、SQL Server データベース・サーバーに最新のアップグレードとパッチを確実に適用することができます。

この場合、サーバー・コンピューターとコンソール・コンピューターを含むすべてのコンピューターにクライアントをインストールする必要があります。さらに、場合により、以下に示すその他のツールおよび手順も使用する必要があります。

- SQL Server がインストールされている場合、**MS SQL Server ツール** を使いこなす必要があります。これにより、データベースをスムーズに実行し続けることができます。
- 定期的にデータベースをバックアップすることが一般的に実践されていますが、これは BigFix データベースに対しても実践する必要があります。また、ときどきエラー・チェックを実行してデータを検証することもお勧めします。
- パフォーマンスが低下していることに気付き始めたら、フラグメント化していないか確認します。BigFix は多数の一時ファイルを書き出します。これにより、ディスクのフラグメント化が進む場合があるため、必要に応じてドライブのデフラグを実行します。定期メンテナンスでも、ときどきディスク・ドライブのエラー・チェックを実行します。
- BigFix**診断ツール**を使用すると、サーバー・コンポーネントを徹底的にテストできます。このツールは、問題が発生したときにいつでも実行できます。詳細については、BigFix 診断ツールの実行 (（ページ）　　)を参照してください。
- **「BigFix 管理」** ドメインをときどき確認します。BigFix コンポーネントに関する問題を検出できる、入手可能な Fixlet が多数あります。これにより、多くの場合、ネットワークに影響が及ぶ前に、問題を防ぐことができます。
- システム全体のパフォーマンスを向上させるためにリレーを追加し、それらに細心の注意を払います。適用環境を正常にするには、リレーが正常であることが重要です。
- **「BigFix 管理」**ドメインの **「適用状態チェック」** ダッシュボードで、最適化および障害について確認します。
- サーバーでモニタリング・アクティビティーをセットアップし、ソフトウェアまたはハードウェアの障害が発生した場合に、以下に関して通知が送られるようにします。

- サーバーの電源オフまたは使用不能
- ディスク障害
- サーバーのアプリケーションに関するイベント・ログ・エラー
- サーバーのサービスの状態
- FillDB バッファー・ディレクトリーのデータ・バックアップの状態

リレーの正常性のモニター

BigFix では、クライアントおよびリレーのセットアップをモニターして、それらが最適に動作していることを確認できます。

大規模なパッチをデプロイする前に、スムーズなロールアウトを確実に実行できるよう、リレーの状況を確認できます。

リレーの配置をモニターする際の推奨事項を以下に示します。

- **「BigFix 管理」**ドメインと**「分析」**ノードをクリックして、リレーの状況分析をアクティブにします。この分析には、リレーの正常性の詳細ビューを表示するためのさまざまなプロパティーが含まれています。
- 分析の**「結果」**タブをクリックして、リレー状況分析の「リレーへの距離」プロパティーをモニターし、ネットワークの正常な状態を確認します。トポロジーが突然変化したり、一部のクライアントがサーバーに到達するために余分なホップを使用していることに気付いたりした場合、リレーの障害を示している可能性があります。
- サーバーに直接報告するクライアントの数を最小化するようにしてください。これは通常、リレーを使用するよりも効率性が低くなります。この分析を調査することで、どのコンピューターがどのリレーに報告しているかが分かります。

リレーおよびサーバーの診断

BigFix 環境のセットアップと状況をモニターし、クライアント上でアクションを実行します。

以下の機能診断を使用して、サーバーおよびリレー設定の情報を取得したり、クライアント上でアクションを実行したりできます。V9.5.6 以降、リレー診断ページはデフォルトで無効になっており、有効に設定された場合はパスワードで保護されます。詳しくは次を参照してください。リレー診断 (（ページ）) からインストールします。

診断にアクセスするには、ブラウザーを開き、アドレス・フィールドに以下を入力します。

```
http://<computer_name>:52311/rd
```

または

```
http://<computer_name>:52311/RelayDiagnostics
```

各部の意味は以下のとおりです。

<computer_name>

検査するサーバーまたはリレーがインストールされているワークステーションのアドレスです。

診断ページは以下のセクションに分割されています。

リレーまたはサーバーの診断

このセクションでは、環境設定に関する情報を収集できます。`+` 記号をクリックして、設定の各種タイプを展開し、その値を確認します。

注: 項目**「照会設定 (Query Settings)」**は、BigFix 照会処理を参照しています。この機能について詳しくは、BigFix Query の使用によるクライアント情報の取得 (（ページ） *145)*を参照してください。

リレー状況情報

このセクションでは、リレー上で使用されるキャッシュの、FillDB 専用のキューおよび BigFix Query 要求および結果専用のキューの情報を表示できます。

- **FillDB ファイル・サイズ制限 (FillDB File Size Limit)**
- **FillDB ファイル・カウンター制限 (FillDB File Counter Limit)**
- **「キュー内の照会のタイムアウト」**は、BigFix Query 要求が削除されるまでのキュー内での滞在時間を示しています。

- **「キュー内の照会のサイズ」**は、BigFix Query 要求を保管するためにリレーで使用されるキャッシュのサイズを示しています。
- **「キュー内の結果のサイズ」**は、BigFix Query 結果を保管するためにリレーで使用されるキャッシュのサイズを示しています。

「照会キューを空にする」ボタンをクリックすると、リレー・キャッシュ内の BigFix Query 要求および結果を保管するキューがクリーンアップされます。

コンソール・ユーザー情報

このセクションでは、ユーザーが BigFix へのアクセスを許可されているかどうかを確認できます。このセクションは、サーバー診断にアクセスした場合にのみ使用可能です。

「ユーザー許可の確認 (Check User Authorization)」をクリックし、ユーザーの資格情報を入力して、そのユーザーが BigFix コンソールへのアクセスを許可されているかどうかを、実際にそれらの資格情報を使用してログインする必要なしに、検証できます。

サイト収集情報

このセクションでは、環境サイトに関連した情報を収集できます。

- **「収集状況ページ 」**をクリックして、サイト収集状況に関する情報を取得します。
- サイト・コンテンツの最新バージョンを取得するには、**「すべてのサイトを収集」**ボタンをクリックします。
- **「Fixlet サイト要求」**では、サイトに関連する各種タイプの要求に関する情報を収集できます。要求のタイプ、表示されたリスト内のサイトの URL、CRC を使用するかどうかを選択し、**「送信」**をクリックします。

クライアント登録

このセクションでは、単一のコンピューターまたは環境内のすべてのコンピューターに対する要求を実行できます。

- **「コンピューター ID の取得」**ボタンをクリックして、リレーのコンピューター ID を確認します。
- **「単一コンピューターの要求 」**では、リスト内の要求のいずれかを選択し、**「送信」**をクリックすることで、単一コンピューターに関連するさまざまなタイプの要求を選択できます。要求タイプによっては、1 つ以上のテキスト・フィールドの入力が必要な場合があります。必要なフィールドは自動的に有効になります。
- **「すべてのコンピューターの要求」**では、表示されたリスト内の要求のいずれかを選択し、**「送信」**をクリックすることで、環境内のすべてのコンピューターに関連するさまざまなタイプの要求を選択できます。要求タイプによっては、**「アクション ID」**の指定が必要な場合があります (有効になっている場合)。

ダウンロード情報

このセクションでは、システムで実行されているダウンロードに関する情報を収集できます。

- **「ダウンロード状況ページ 」**をクリックして、サーバーまたはリレー上でアクティブなダウンロードに関する情報を取得します。
- **「ダウンロード状況テキスト・ページ」**をクリックして、サーバーまたはリレー上でアクティブなダウンロードに関する情報を XML 言語で取得します。
- **「ダウンロード要求」**では、関連フィールドに**「アクション ID」**と**「サイト URL」**を指定することで、特定のサイトの特定のアクションに関する情報を収集できます。**「ダウンロード収集要求 」**ボタンをクリックして、要求を実行します。

仮想化環境および仮想マシン

オペレーティング・システムを複数の仮想マシンで実行できます。

BigFix では、オペレーティング・システムを複数のイメージで実行して、ハードウェアとソフトウェアのリソースを共有するメリットを得ることができます。特に、HCL z Systems では、z/VM 環境の中で Linux イメージが IBM z Systems サーバーの信頼性、可用性、保守

性と内部の高速通信のメリットを得ることができます。z/VM は、Linux ワークロードを単一の物理サーバーに統合して、数百から数千の Linux イメージを実行できるようにする理想的なプラットフォームです。

BigFix の設計では、BESClient エージェントは、ループで作動し、ディレクトリー `<BESClient_installation_path>/__BESData` の内容に基づいて、実行するアクティビティーを検査します。これらのアクティビティーと、z/VM 環境で一般的に見られる多数の並行仮想マシンにより、CPU 使用率が 100% になる可能性があります。この問題を防止して、プロセスへの CPU 割り当てを制御するには、CPU 使用率 (（ページ）) に記載されている構成設定を使用します。

いくつかの有用なパラメーターとしては、作業の量とアイドル時間の長さのバランスを取ることで CPU 使用量を制御する `_BESClient_Resource_WorkIdle` と `_BESClient_Resource_SleepIdle` が挙げられます。これらのデフォルト値は、それぞれ 10 ミリ秒と 480 ミリ秒です。デフォルト値が使用される場合、各仮想マシンの作業は約 2% になります。さらに低いパーセンテージを得る必要がある場合は、これらの値を変更できます。この場合のマイナス面は、新規アクティビティーを処理する必要があるときに、BigFix クライアントが低速になることです。新しい値を設定することで、仮想マシンの数を考慮に入れて、全 CPU の使用率が 100% になることを回避できます。

その他のパラメーターでは、エージェントを 1 日の一定時間にわたって静止状態にして、残りの時間はアクティブになるように設定することができます。静止期間中、CPU 使用量はほぼ 0% になります。この動作を制御するパラメーターは、`_BESClient_Resource_QuietEnable`、`_BESClient_Resource_QuietStartTime`、および `_BESClient_Resource_QuietSeconds` です。例えば、以下の値を設定します。

```
_BESClient_Resource_QuietEnable=1
_BESClient_Resource_QuietSeconds=43200
_BESClient_Resource_QuietStartTime=07:00
```

エージェントは、毎日 07:00 AM に静止モードになり、43,200 秒間 (12 時間) にわたってこの状態のままになり、07:00 PM にウェイクアップします。静止モード中、エージェントが使用する CPU 時間はほぼ 0% で、アクティビティーを処理しません。

クライアントがスリープ・モードになっている時間の長さを制御するためのその他の有用なパラメーターは、`_BESClient_Resource_PowerSaveEnable` および

`_BESClient_Resource_PowerSaveTimeout`*X* (0 から 5 の範囲の X)です。特に、バッテリーの低電力の問題がある場合や、CPU 使用率を下げる必要がある場合などに役立ちます。

上記のパラメーターやその他多数のパラメーターの詳細な説明については、上記のリンクの構成設定を参照してください。

関連資料

設定のリストと詳細な説明 (（ページ）　　)

関連情報

CPU 使用率 (（ページ）　　)

BES クライアント・ヘルパー・サービス (Windows のみ)

BES クライアント・ヘルパーは、BES クライアントのウォッチャー・プロセスであることが意図されており、BES クライアントが実行されていない場合にサービスの再起動を試行します。

BES クライアント・ヘルパーは、BES クライアント・サービスが適切なタイミングで開始されない場合に、いくつかのトラブルシューティング手順も実行します。

1. 再起動を試みます。
2. 失敗した場合は、失効ファイルの削除 (バックアップ・コピーの作成) を試行して、新しい再起動を試みます。
3. 失敗した場合は、BESData フォルダーを削除して、最後の再起動を試みます。

このツールは、サービスとしてインストールされることを意図しています。以下の Fixlet を使用してインストールおよびアンインストールできます。

- #591: BES クライアント・ヘルパー・サービスをインストールします
- #592: BES クライアント・ヘルパー・サービスをアンインストールします

このサービスは、デフォルトで BES クライアント・プロセスを 1 日に 1 回チェックし、ログ・ファイルは生成されません。インストール中に、別のチェック頻度を選択し、ロギング・アクティビティーを有効にすることができます。

インストール後に設定を変更する方法: 頻度

必要な頻度 (秒単位で指定) をレジストリー・キー `_[HKEY_LOCAL_MACHINE\SOFTWARE\BigFix\BESClientHelper\ServiceRunPeriod]_` に設定して、サービスを再起動します。

インストール後に設定を変更する方法: ログ

Fixlet を使用して、別の設定でヘルパーをアンインストールして再インストールします。

または、次のようにしてサービスを再インストールできます。

1. レジストリー・キー `_[HKEY_LOCAL_MACHINE\SOFTWARE\BigFix\BESClientHelper\ServiceInstallationParameters]_` を「-l」(引用符なし) に変更してロギングを有効にするか、空にして無効にします。
2. 実行 `_<path of BESClient>\BESClientHelper.exe -remove_`
3. 実行 `_<path of BESClient>\BESClientHelper.exe -install auto_`

BES ルート・サーバーおよび BES リレー・サービスのデバッグ/詳細ロギングの有効化

この手順では、BigFix サーバーまたはリレーでのデバッグ/詳細ロギングを有効にして、BigFix サーバーとリレーによって実行されるアクティビティーをログに記録するための手順について説明します。

BigFix サーバーまたはリレーでデバッグ/詳細ロギング・レベルを有効にするには、以下の手順を実行します。

ロギングを有効にする方法は複数あります。BigFix コンソールを使用して BigFix クライアント設定を作成するか、それをマシン上で手動で有効化します。

Fixlet を使用したロギングの有効化

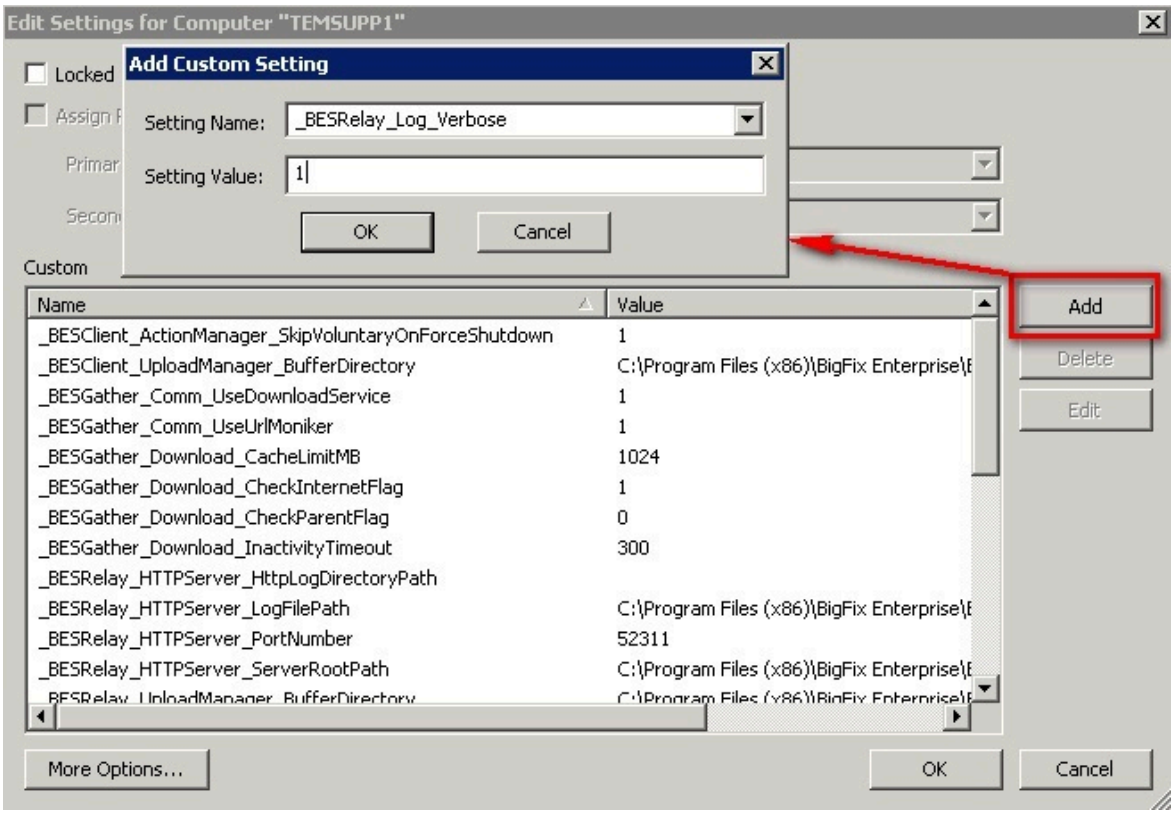
BigFix サーバーまたはリレーで詳細ロギングを有効/無効にするには、以下の BESSupport Fixlet を使用します。

- Fixlet ID: 4595 - サーバー詳細ログの有効化
- Fixlet ID: 4596 - 警告: サーバー詳細ログが有効になっています

- Fixlet ID: 4776 - リレー詳細ログの有効化
- Fixlet ID: 4777 - 警告: リレー詳細ログが有効になっています

BigFix コンソールを使用したロギングの有効化

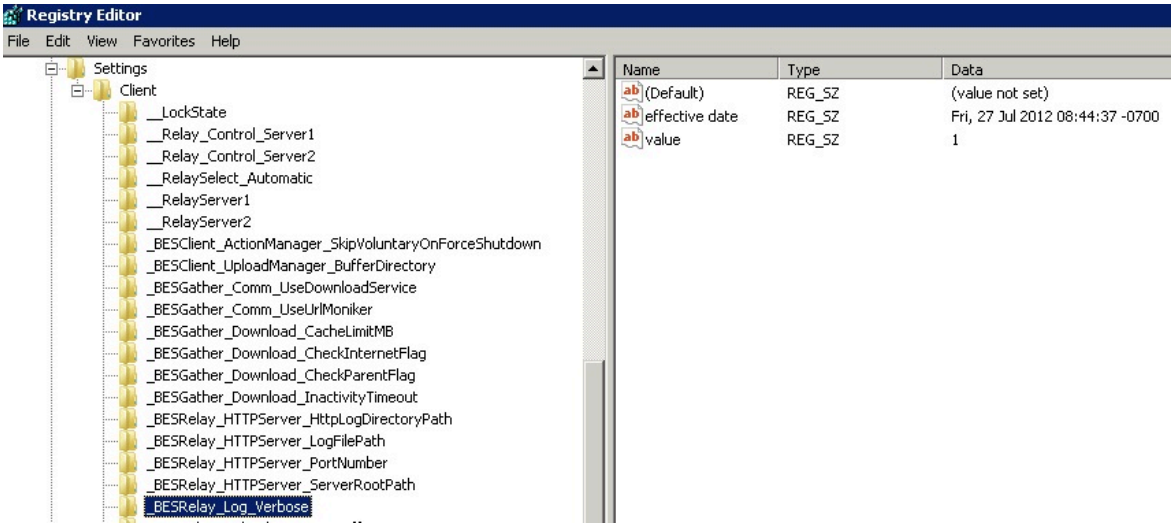
1. コンソールにマスター・コンソール・オペレーターとしてログインします。
2. コンソールで BigFix サーバーまたはリレー・コンピューターを右クリックします。
3. 「コンピューター設定の編集...」を選択します。
4. リストを調べて、_BESRelay_Log_Verbose 設定が既に作成されているかどうかを確認します。作成済みの場合は、「編集」ボタンをクリックし、その値を 1 に変更します (有効にします)。
5. 設定がまだ作成されていない場合には、「追加」ボタンをクリックして作成します。設定名に「_BESRelay_Log_Verbose」を入力し、設定値に「1」を入力して、詳細ロギングを有効にします。



6. 「OK」をクリックします。「Change '_BESRelay_Log_Verbose' Setting」という名前のアクションが、BigFix サーバーまたはリレー・マシンで対象となります。
7. アクションが正常に完了した後 (および設定が適用された後)、BES ルート・サーバー・サービスの場合には新しいロギング・レベルが有効になりますが、BigFix リレーの場合には BES リレー・サービスの再起動が必要です。タスク番号 447 に基づいてアクションを実行できます。これを行うには、BES サポート・サイトでサービスを再起動します。

レジストリーを使用した手動によるロギングの有効化 (Windows)

1. BigFix サーバーまたはリレー・マシンにログインします。
2. レジストリー・エディター (regedit) を開きます。
3. HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\BigFix\EnterpriseClient\Settings\Client にレジストリー・キー _BESRelay_Log_Verbose を追加します。
4. 「Value」という名前で REG_SZ 値を作成します。
5. 値を 1 に設定します。



6. リレーでは、BES リレー・サービスを再始動しますが、BigFix サーバーでは必要ありません。

詳細データは、サーバーでは <BigFix_Server_Installation_Folder>\BESRelay.log ファイルに出力され、リレーでは <BigFix_Relay_Installation_Folder>\logfile.txt に出力されます。

例えば、サーバーでの <BigFix_Server_Installation_Folder> は `C:\Program Files (x86)\BigFix Enterprise\BES Server` となります。

設定ファイルを使用した手動によるロギングの有効化 (Linux)

1. BigFix リレー・マシンにログインします。
2. 以下のコマンドを使用して、BESClient サービスを停止して、構成ファイルへの変更が上書きされないようにします。

```
service besclient stop
```

3. 構成ファイル /var/opt/BESClient/besclient.config を編集し、以下の行を追加または変更します。

```
[Software\BigFix\EnterpriseClient\Settings\Client\_BESRelay_Log_Verbose]
```

```
effective date = [Enter Current Date Time In Standard Format]
value = 1
```

有効な現在日時は、「Wed, 06 Jun 2012 11:00:00 -0700」と同様の形式にする必要があります。

4. 以下のコマンドを使用して BESClient サービスを開始します。

```
service besclient start
```

5. リレーの場合は、BESRelay サービスを再開します。

```
service besrelay stop
service besrelay start
```

注: 変更を有効にするために、BigFix サーバーの BESRootServer サービスを再開する必要はありません。

注: 詳細データは、サーバーとリレーの両方で `/var/log/BESRelay.log` ファイルに出力されます。

注: 詳細ロギングを無効にするには、BigFix クライアント設定を 0 に設定します。

警告: 詳細ロギングは、ディスク・スペースおよび処理リソースを節約するために、発生している問題のトラブルシューティングに必要な時間の分だけにしておきます。大規模な環境では、長期間にわたって詳細ロギングを有効にしておくと、BES ルート・サービスのパフォーマンスが大幅に低下して、コンソールのタイムアウトやサーバー・アクティビティーのデッドロックが発生することがあります。

注: アクティブ・ログ・ファイルの他に、ローテーションされた最大 10 個のログ・ファイルが保持され、その名前は logfile.txt_0、logfile.txt_1、...、logfile.txt_9 というようになります。デフォルト値は 50*1024*1024 (52,428,800) バイトです。

Appendix A. Support

For more information about this product, see the following resources:

- BigFix Support Portal
- BigFix Developer
- BigFix Playlist on YouTube
- BigFix Tech Advisors channel on YouTube
- BigFix Forum

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.