BigFix
Configuration Guide

Special notice

Before using this information and the product it supports, read the information in Notices (on page 1236).

Edition notice

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

Chapter 1. Introduction

This guide explains additional configuration steps that you can run in your environment after installation.

What is new in BigFix 11 Platform

BigFix Platform 11 delivers three major changes in the security area by adding support for OpenSSL3, SHA-384 and TLS 1.3. This releases also delivers new features, an updated list of supported platforms, and several upgraded libraries.

Patch 7

Custom content, managed like code

- **External Repository Sites:** BigFix can now connect directly to an external source control system, which is Git, and treat that repository as an “External Repository Site”. For details, see External Repository Sites *(on page 1191)*.

Bring Identity Provider users and computers information into BigFix

- **User-based management:** BigFix is now able to retrieve computer data from Active Directory and Microsoft Entra ID, such as the User associated to the Computer, the User Groups and Computer Groups it belongs to, and more. For details, see User-based management *(on page 93)*.

Performance at scale

- **Boosted BigFix Relay Scale:** 11.0.7 relays on Windows and RHEL implement architecture improvements to allow up to 20000 endopoints connected to a single relay, whether it is non-authenticated or authenticated. For more information please see BigFix Performance and Capacity Planning Resources.

- **Faster client performance on multi-core systems:** A new client setting (`_BESClient_Resource_ScaleWorkTimeWithCPUCores`, disabled by default) lets the BigFix Client take fuller advantage of available CPU cores, helping actions and evaluations complete faster on modern multi-core hardware. For details, see List of settings and detailed descriptions *(on page 404)*.
- **Faster onboarding and operation for large cloud deployments:** The Plugin Portal now evaluates Fixlets, properties, and actions concurrently across computers, decouples computer registration from execution, and batches registration requests to the BigFix Server.
- **Improved Console search engine:** Searching objects in the Console (Fixlets/Tasks, Baselines, Analyses, Actions) is now more efficient, contributing to improving user experience.

Compliance and security, reinforced

- **FIPS 140-3 readiness:** Organizations operating in regulated or government environments can now enable FIPS 140-3 mode. For details, see FIPS 140-3 cryptography in the BigFix environment *(on page 1071)* and Configuring FIPS 140-3 on the BigFix Server *(on page 1072)*. BigFix encryption module is also **FIPS 140-3 certified** https://csrc.nist.gov/projects/cryptographic-module-validation-program/certificate/5159.
- **Automatic protection for idle consoles:** A new setting lets you configure a BigFix Console lock timeout, so unattended, logged-in sessions are automatically locked. For details, see Setting the BigFix Console lock timeout.

- **Stronger download integrity:** Action scripts that fetch files (“add prefetch item” and “prefetch”) now support SHA-512. For details, see add prefetch item and prefetch.
- **Tighter control over MCP access:** A new "blockIncomingMCPRequests" option on the BigFix root server lets administrators disable incoming MCP requests. For details, see Installing and configuring BigFix Platform MCP Server.

More resilient administration

- **Preventing conflicting admin operations:** BigFix now blocks more than one instance of BESAdmin from running at the same time on a given computer. For details, see BESAdmin Windows GUI, BESAdmin Windows Command Line and BESAdmin Linux Command Line.

Expanded platform support

BigFix Agent now supports additional platforms, extending consistent management and visibility to more of your modern infrastructure footprint:

- New! Linux distribution on ARM processor
  - Ubuntu 24.04, 26.04 LTS ARM 64
  - Red Hat Enterprise Linux 9, 10 ARM 64
  - Debian 13 ARM 64
- macOS 27 ARM 64
- Ubuntu 26.04 LTS x86 64-bit
- VIOS 4.1.1, 4.1.2 PPC 64-bit on Power 11
- AIX 7.2, 7.3 PPC 64-bit on Power 11
- Red Hat Enterprise Linux 9, 10 PPC 64-bit LE on Power 11
- SUSE Linux Enterprise Server (SLES) 15, 16 PPC 64-bit LE on Power 11

Inspector changes

- New inspector types named "idp group" and "idp user" were added to return information about the groups and users associated with a computer joined to an Identity Provider (IdP). For details, see idp group and idp user.
- New properties were added to the "cryptography" inspector. These properties return whether BigFix components in the deployment should operate in FIPS 140-2 or FIPS 140-3 mode. List of added properties:
  - fips_140_2 mode of
  - fips_140_3 mode of

  For details, see cryptography.
- New properties were added to the "license" inspector. These properties return whether BigFix components in the deployment should operate in FIPS 140-2 or FIPS 140-3 mode. List of added properties:
  - fips_140_2 mode of
  - fips_140_3 mode of

  For details, see license.
- New properties were added to the "client" inspector. These properties return various information if the computer is joined to an Identity Provider (IdP). List of added properties:
  - idp directory type of
  - idp distinguished name of
  - idp group of
  - idp id of
  - idp display name of
  - idp sam account name of
  - idp user of

  For details, see client.

- A new cast named "as xml string" was added to the "bes action", "bes computer group", "bes fixlet" and "bes property" session inspectors to convert the specified BES object to an XML string. The new cast was introduced exclusively for the BigFix Console and BigFix Explorer. It is not supported in Web Reports. For details, see bes action, bes computer group, bes fixlet and bes property.

Upgraded libraries, binaries and 3rd party tools

- The boost library was upgraded to Version 1.91.0.
- The libcURL library was upgraded to Version 8.22.0.
- The libgit2 library was introduced (Version 1.9.7).
- The OpenSSL library was upgraded to Version 3.5.8.

Patch 6

Security and Authentication improvements

- **REST API Token Authentication:** BigFix Platform now supports authentication using tokens for the BigFix Server REST APIs, the IEM CLI, and the BigFix Explorer REST APIs. This provides a modern and secure approach to automation and integrations, leveraging bearer tokens and avoiding the need to share/store credentials. For details, see Configuring bearer token authentication *(on page 154)*.
- **Client Certificate Max Validity is now configurable:** With the `setClientCertificatePolicy` BESAdmin command, you can now customize the maximum validity period for newly issued client certificates - this will help you comply with company policies and changing industry standards. You can also define how long before expiration a BigFix Client should attempt the renewal. For details, see

Client certificate *(on page 1202)*, BESAdmin Windows Command Line and BESAdmin Linux Command Line.

- **Forcing Prefetch commands to use HTTPS on Relays:** You can now use the `_BESRelay_Download_UseHttps` configuration setting on Relays to force the use of HTTPS in prefetch statements. This capability was already available on the BigFix Root Server. For details, see Managing Downloads *(on page 1099)*.

PeerNest enhancements

- **Site Content can now be shared among peers:** Further leverage Peer-to-Peer benefits by sharing site content among peers, in addition to files, minimizing network traffic through Relays. For details, see Working with PeerNest *(on page 378)* and Peer to peer mode *(on page 915)*.
- **PeerNest Statistics:** BigFix now allows you to collect PeerNest statistics across your entire environment and export them for analysis. You can now measure effectiveness and efficiency of PeerNest and identify issues. In addition, a session relevance inspector and a Web Reports page have been added to allow monitoring key PeerNest metrics. For details, see PeerNest statistics: Exporting a JSON file with aggregated data *(on page 394)*, PeerNest, PeerNest and bes peer download.

64-bit BigFix Relay for Windows

On Windows, the BigFix Relay is now available as a 64-bit executable. This adds improved encryption and memory address space capabilities. The 32-bit Relay remains available for 32-bit operating systems. When performing a Relay upgrade on a Windows 64-bit OS, the 64-bit version of the Relay will automatically be installed.

Cloud Plugins new features

- **Discovery on Demand:** You can now manually trigger the discovery process for Cloud Plugins. This feature forces an immediate update of the discovered assets, avoiding the need to wait for the scheduled discovery interval. For details, see Discovery on Demand *(on page 217)*.
- **GCP Agent Installation:** You can now install the BigFix Client on virtual machines on Google Cloud faster via Google APIs, instead of using the Client Deploy Tool. For details, see BigFix Agent installation on cloud resources *(on page 353)*.
- **Cloud Tags generation using action commands:** The BigFix cloud plugins now have the capability to set or delete a tag on AWS, Azure, Google Cloud Platform, and VMware via action commands. For details, see Amazon Web Services Plugin Commands *(on page 337)*, Microsoft Azure Plugin Commands *(on page 340)*, Google Cloud Platform Plugin Commands *(on page 338)* and VMware Plugin Commands *(on page 341)*.

Expired or stopped actions automatic cleanup

You can now define a policy to automatically mark actions as deleted in a given number of days after they are stopped or they expire. You can configure this option either from the **Clean Up** tab of the BESAdmin Windows GUI or using the BESAdmin CLI option named `healthcleanup`. For details, see Health and Cleanup, BESAdmin Windows Command Line and BESAdmin Linux Command Line.

Toggle Fixlet visibility via API

Added new REST APIs under the Fixlet family to toggle and retrieve Fixlet global visibility status. For details, see RESTAPI Fixlet.

Management of the MS-SQL reindex job across platform upgrades

MS-SQL reindex job is no longer overwritten during a platform upgrade. In this way, modifications that were applied to it are not lost at every upgrade.

BigFix Scanner is now removed with the BigFix Agent

When removing the BigFix Client with the BigFix Removal Utility (BESRemove), the BigFix Scanner is also removed, if present.

Serviceability enhancements

- Analysis 5990 “Pending Restart information for Windows” was added to BES Support to help identify causes for “Pending Restart” status on clients
- The maximum allowable value for the _BESClient_Resource_InterruptSeconds setting was increased to 20 minutes
- Detailed logging was added in Root Server log for client re-registration failures, to expedite troubleshooting.

Removal of obsolete Fixlets and Tasks from BES Support

To streamline the publishing and gathering process and reduce the amount of content that agents must evaluate, obsolete Fixlets and Tasks were removed from the BES Support content:

- **Deprecation of Dynamic Bandwidth Throttling:** The Dynamic Bandwidth Throttling feature has been officially deprecated. In complex modern networking environments, dynamic bandwidth calculations are not always reliable or predictable. The use of Dynamic Throttling has been

discouraged for a long time in favor of Static Throttling and modern features like PeerNest, which manage bandwidth far more efficiently. Consequently, the following configuration Fixlets have been permanently removed:

  - 457 BES Client Setting: Dynamic Download Throttling
  - 458 BES Relay Setting: Dynamic Download Throttling
  - 459 BES Relay Setting: Dynamically Throttle Outgoing Traffic
  - 462 BES Server Setting: Dynamically Throttle Outgoing Traffic
  - 605 BES Client Setting: Enable/Disable Dynamic Throttling
  - 702 BES Relay/Server Setting: Enable/Disable Dynamic Throttling

- **BigFix 9.5 Content Cleanup:** Content used to deploy and manage older patch versions of BigFix 9.5 has been removed. BigFix v9.5 reached End of Support on June 30th, 2024. BES Support will only retain the content to deploy and manage the final patch of BigFix 9.5 (version 9.5.25).

Inspector changes

- New property named “metered connection of” was added to the “network adapter” client inspector. This returns information about the network adapter configuration as it relates to metered network connections. For details, see network adapter.
- New session inspector type named “bes peer download” was added to return information about files that were shared or downloaded using the PeerNest feature. For details, see bes peer download.

- New property named “evaluation period of” was added to the “bes fixlet” session inspector. This returns information about the frequency with which clients re-evaluate a Fixlet. For details, see bes fixlet.
- New properties were added to “bes user” session inspector. These properties return information about the explicit and effective permissions of a BigFix Operator. List of added properties:
    - effective can create actions flag of
    - effective can lock flag of
    - effective can send multiple refresh flag of
    - effective can submit queries flag of
    - effective custom content flag of
    - effective master flag of
    - effective restartandshutdown actionscript privilege allowboth flag of
    - effective restartandshutdown actionscript privilege allowrestartonly flag of
    - effective restartandshutdown actionscript privilege none flag of
    - effective restartandshutdown postaction privilege allowboth flag of
    - effective restartandshutdown postaction privilege allowrestartonly flag of
    - effective restartandshutdown postaction privilege none flag of
    - effective show other action flag of
    - effective stop other actions flag of
    - effective unmanagedasset privilege scanpoint flag of
    - effective unmanagedasset privilege showall flag of
    - effective unmanagedasset privilege shownone flag of

- explicit can create actions flag of
- explicit can lock flag of
- explicit can send multiple refresh flag of
- explicit can submit queries flag of
- explicit custom content flag of
- explicit master flag of
- explicit restartandshutdown actionscript privilege allowboth flag of
- explicit restartandshutdown actionscript privilege allowrestartonly flag of
- explicit restartandshutdown actionscript privilege none flag of
- explicit restartandshutdown postaction privilege allowboth flag of
- explicit restartandshutdown postaction privilege allowrestartonly flag of
- explicit restartandshutdown postaction privilege none flag of
- explicit show other action flag of
- explicit stop other actions flag of
- explicit unmanagedasset privilege scanpoint flag of
- explicit unmanagedasset privilege showall flag of
- explicit unmanagedasset privilege shownone flag of

For details, see bes user.

Added Support for Operating Systems and Databases

- Added support for BigFix Server running on:
  - Windows Server (2019 or later) with Microsoft SQL Server 2025 database
  - Red Hat Enterprise Linux 10 x86 64-bit with Microsoft SQL Server 2025 database

- Starting from Red Hat Enterprise Linux 9.4 x86 64-bit and later fix packs, the support was added for BigFix Server Version 11.0.6 with DB2 12.1.3. This DB2 version (12.1.3) is not compatible with Red Hat Enterprise Linux 10.
- Starting from Red Hat Enterprise Linux 10 x86 64-bit, the support was added for BigFix Server Version 11.0.6 with DB2 12.1.4.
- Added Support for BigFix Server on Windows and Linux with databases Azure SQL Managed Instance for Microsoft SQL Server 2025 and Google Cloud SQL for Microsoft SQL Server 2025:
    - Starting from Patch 6, BigFix Server on Windows Server (2019 or later) and on Linux Red Hat 9 supports Azure SQL Managed Instance for Microsoft SQL Server 2025 and Google Cloud SQL for Microsoft SQL Server 2025 databases.
- Added support for BigFix Agent running on:
    - Debian 13 x86-64
    - SUSE Linux Enterprise Server (SLES) Version 16 x86-64.

Upgraded libraries, binaries and 3rd party tools

- The Codejock library was upgraded to Version 24.2.0.
- The InstallShield was upgraded to Version 2025 R2.
- The libcURL library was upgraded to Version 8.18.0.
- The Microsoft ODBC Driver was upgraded to Version 18.5.2.1.
- The OpenLDAP library was upgraded to Version 2.6.10.
- The OpenSSL library (including FIPS module) was upgraded to Version 3.5.5.
- The SQLite library was upgraded to Version 3.51.0.

- The zlib library was upgraded to Version 1.3.2.
- The Microsoft Visual C++ Redistributable was upgraded to Version 14.44.35211.0.

Patch 5

Suspending downloads for expired actions

By default, if an action expires while the Relay is downloading some files for that action, the BigFix Relay completes the downloads.

Using the new setting named `_BESRelay_Download_SuspendExpiredAction` you can now choose to suspend these downloads.

If more than one action has requested to download the same file, the BigFix Relay suspends the download only when all these actions expire.

If a new action requires downloading a suspended file, the BigFix Relay resumes the suspended download from where it was left off.

For details, see Managing Downloads (on page 1099).

Forcing Prefetch commands to use HTTPS

You can now force a Prefetch command to download files using HTTPS protocol. By default, the URL provided in the Prefetch command will be used (whether it is HTTPS or HTTP).

You can use the new `_BESRelay_Download_UseHttps` setting to force a Prefetch statement containing a HTTP URL to use HTTPS instead or fail. As an alternative, you can configure it to use HTTPS or fallback to HTTP.

This setting applies to the Root Server only.

For details, see Managing Downloads (on page 1099).

Advanced information in the audit log

The following additional information is now available in BigFix audit log, related to actions performed by BigFix operators:

- Create/Update/Delete Content (Fixlets, Tasks, Analyses, Baselines)
- Create/Update/Delete Custom Sites
- Create/Update/Delete Computer Groups
- Globally Hide/Unhide Content
- Update Site Subscription (computers)

For details, see Configuring logs (on page 1154).

Web Reports- you can export and import reports within a BigFix deployment

Starting from BigFix Version 11.0.5 a Web Reports user can export reports (both public and private, with the exception of pre-installed ones) to a JSON file; the file can then be used to import the report in the same BigFix deployment.

For details, see Import Report, Exporting Reports in JSON format and Exported JSON Details.

Web Reports- at session timeout, hide UI and revert to login screen

As a security and data privacy measure, when a UI session timeout happens, the operator should be logged out and brought back to the login screen, to avoid someone else accessing the data that remains on the screen. This happens now also in Web Reports, as it was already happening in other interfaces.

Console SAML login may fail loading dynamic content (KB0124024)

The BigFix Console can now be configured to show the SAML login panel using a WebView2-based dialog. This allows to

overcome the problem described in the Defect Article Console SAML login may fail load dynamic content.

Inspector changes

- New client inspector types named:
  - yaml key
  - yaml value

  were added to represent YAML keys and values. For details, see yaml key and yaml value.
- New client inspector types named:
  - os log store
  - os log entry log

  were added to allow querying log entries from the local OS log store on macOS. For details, see os log store and os log entry log.
- New client inspector properties named:
  - last logon time of
  - last logon type of
  - last logon type number of
  - logon completion time of
  - logon session time of
  - logon session type of
  - logon session type number of

  were added to return information about the logon that started a Windows session. For details, see logged on user.
- New client inspector property named:
  - protocol type of

  was added to return the protocol type of the connected WiFi network. For details, see wifi network.
- The following client inspector properties named:

- pid of <service> : integer
- file of <service> : file
- image path of <service> : string

are now supported also on Red Hat and SUSE. For details, see service.

- New session inspector properties were added to return information about the remediated Fixlets and computers. New properties available on BigFix Explorer, Console and Web Reports:
  - remediated <bes fixlet> of <bes computer>
  - remediated <bes computer> of <bes fixlet>
  - remediated <(bes computer, bes fixlet)>
  - remediated <(bes fixlet, bes computer)>

  New properties available only on BigFix Explorer:
  - remediated computer set of <bes fixlet>
  - remediated computer of <bes fixlet>
  - remediated computer count of <bes fixlet>
  - relevant fixlet count of <bes computer>
  - remediated fixlet count of <bes computer>
  - remediated fixlet of <bes computer>
  - remediated fixlet set of <bes computer>

  Properties now available also on BigFix Explorer:
  - first became relevant of <bes fixlet result>
  - last became nonrelevant of <bes fixlet result>
  - last became relevant of <bes fixlet result>

  For details, see bes computer and bes fixlet.

Added Support for BigFix Relay

Added support for BigFix Relay running on Red Hat Enterprise Linux 10 x86 64-bit.

Added Support for BigFix Agent

Added support for BigFix Agent running on:

- macOS 26 ARM/x86 64-bit
- OpenSUSE Leap 16.0 x86-64
- Oracle Enterprise Linux 10 x86-64
- VIOS 3.1.4

Added Support for BigFix Plugin Portal

Added support for BigFix Plugin Portal running on Red Hat Enterprise Linux 10 x86 64-bit.

Library and driver upgrades

- The AWS Cloud plugin was upgraded to AWS SDK for GO v2.
- The OpenSSL library was upgraded to Version 3.5.2.
- The API version used by agents to retrieve information from Azure has been updated to 2023-07-01.

Patch 4

You can now secure HTTPS communication in BigFix with your preferred Certificate Authority (aka Bring Your Own CA)

With the `setcustomca` BESAdmin command, you can install an external Certificate Authority (CA) in the BigFix Platform. When that happens, BigFix components at 11.0.4 and later versions will start using certificates generated from the imported CA to secure HTTPS traffic. Back level components will continue to use the current BigFix CA.

For details, see Configuring the root server with a custom CA (on page 1207), BESAdmin Windows Command Line and BESAdmin Linux Command Line.

More console performance improvements

Prior releases have improved console load times and performance analytics. Our goal was to cut console load times in half, and customers generally meet or exceed this goal. With this release we have improved the general console performance, with a focus on process concurrency, efficiency, and device correlation impact. The net is through our cumulative improvements, you have a more responsive console.

More options for High Availability & Disaster Recovery (HADR)

The BigFix HADR solution until now has involved the Disaster Server Architecture (DSA) based on proprietary replication code. We now offer a full HADR solution based on a combination of database replication (e.g., and MS SQL Availability Group), cluster management, and shared file store. The solution aligns with industry best practices, has reference level data integrity, and is operationally more scalable and lighter weight than the DSA solution. This is especially important if you consider BigFix “mission critical” to your business.

For details, see HA and DR configurations.

Enhanced PeerNest - Streamlined functionality and reduced network traffic

Significant advancement in PeerNest efficiency: agents can exchange payloads among themselves without the need of Multicast UDP. Root Server and Relays will be used as trackers, sharing information about which clients are able to share which files. This means reduced network traffic, possibility to leverage the capability in diverse networking environments, and improved efficiency especially in the early phases of the communication.

For details, see Configuring PeerNest without the multicast function (on page 391) and Peer to peer mode (on page 915).

Higher Relay Scalability

A non-authenticating relay on Windows and Linux can now support up to 10000 endpoints.

Increased number of persistent connections

Persistent connections can be enabled on network configurations where UDP notifications are not available. The limit of endpoints that can be configured in this way is increased to 1000 per Relay.

BigFix Relay metrics collected and exposed

You can now monitor the status and efficiency of your relay infrastructure. The relay is now able to collect and expose metrics related to its HTTP server operations and lets you use Prometheus to collect them. The collected data can then be viewed and analyzed by a visualization application such as Grafana.

For details, see Exposed Metrics and Exported Metrics.

Data upload infrastructure enhancement - Pass-Through mode on Relays

In file upload scenarios, BigFix Platform allows you to enable the BigFix Relays to immediately forward received file chunks upstream, instead of storing files locally, using a new setting named `_BESRelay_PostFile_PassThrough`.

For details, see List of settings and detailed descriptions (on page 404).

VMware cloud plugin connection secured with vCenter certificates

BigFix Platform allows you to install the vCenter certificates on the system where you intend to install the VMware cloud plugin; this allows it to open secure connections.

For details, see Configuring cloud plugins (on page 279).

Azure cloud plugin is able to put the VMs in a deallocated state

The Azure cloud plugin has a new command named `deallocate instance`; by leveraging this feature you can more closely control your infrastructure costs on Azure.

For details, see Microsoft Azure Plugin Commands (on page 340).

Added BESAdmin command to convert BigFix operators from one Identity Provider to another

With the `converttoidpoperators` BESAdmin command, you can convert BigFix operators from one Identity Provider to another already configured Identity Provider. This is useful if, for example, you want to convert your BigFix operators from referencing MSAD to referencing Entra ID. This command is available on Windows platform only.

For details, see BESAdmin Windows Command Line.

Added BESAdmin command to set the priority in case of transaction deadlocks in the database

With the `databaseDeadlockPriority` BESAdmin command, you can set a "Low", "Normal" or "High" priority for the BigFix Administration tool Computer Remover and Audit Trail Cleaner so that you can influence how Microsoft SQL Server resolves the deadlocks. In particular, setting the `databaseDeadlockPriority` option value to "High" ensures that critical transactions are most likely not interrupted.

For details, see BESAdmin Windows Command Line, BESAdmin Linux Command Line, Computer Remover and Audit Trail Cleaner.

Added BESAdmin command to rotate the BigFix WebUI certificate

With the `rotatewebuicredentials` BESAdmin command, you can rotate one BigFix WebUI certificate associated to a given hostname, or the whole BigFix WebUI Certificate Authority (CA) and all BigFix WebUI certificates.

For details, see BESAdmin Windows Command Line and BESAdmin Linux Command Line.

Inspector changes

- New client inspector type named "cidr subnet" was added to represent the Classless Inter-Domain Routing (CIDR) value. For details, see cidr subnet.
- New client inspector properties named "first line of", "first line <integer> of <file>", "last line of" and "last line <integer> of <file>" were added to return specific lines of a given file. For details, see file line.
- New client inspector properties named "first rawline of", "first rawline <integer> of <file>", "last rawline of" and "last rawline <integer> of <file>" were added to return specific rawlines of a given file. For details, see file rawline.
- New client inspector properties named "enabled of", "global state of" and "stealth enabled of" were added to return specific firewall information. For details, see firewall.
- New "named" constructor for the process inspector on MacOS was added to return the process object corresponding to the specified name. For details, see process.
- The following client inspectors named:
  - firewall
  - process <string>
  - swap
  - free amount of <swap>
  - size of <swap>

- total amount of <swap>
- used amount of <swap>
- action of <firewall rule>
- application name of <firewall rule>
- service name of <firewall rule>
- rule of <firewall>

are now supported also on MacOS. For details, see firewall, firewall rule, process and swap.

Added Support for BigFix Server on Windows and Linux with database Amazon RDS for Microsoft SQL Server 2022

Starting from Patch 4, BigFix Server on Windows Server (2019 or later) and on Linux Red Hat 9 supports Amazon RDS for Microsoft SQL Server 2022 database.

Added Support for BigFix Server on Windows and Linux with database Azure SQL Managed Instance for Microsoft SQL Server 2022

Starting from Patch 4, BigFix Server on Windows Server (2019 or later) and on Linux Red Hat 9 supports Azure SQL Managed Instance for Microsoft SQL Server 2022 database.

Added support for BigFix Relay

Added support for BigFix Relay running on Tiny Core 16.

Added Support for BigFix Agent

Added support for BigFix Agent running on:

- Alma Linux 9 x86 64-bit
- Alma Linux 10 x86 64-bit
- Red Hat Enterprise Linux 9 on LinuxONE Systems
- Red Hat Enterprise Linux 10 x86 64-bit

Added Support for BigFix Explorer

Added support for BigFix Explorer running on Red Hat Enterprise Linux 10 x86 64-bit.

Library and driver upgrades

- The boost library was upgraded to Version 1.87.0.
- The InstallShield was upgraded to Version 2024 R2.
- The jQuery-UI library was upgraded to Version 1.14.0.
- The libssh2 library was upgraded to Version 1.11.1.
- The OpenSSL library was upgraded to Version 3.2.4.
- The Xerces-C++ XML Parser library was upgraded to Version 3.3.0.

Patch 3

Added filtering capability to BigFix Explorer component

In Patch 2 the new "BigFix Explorer" component was added to allow easier REST API access to BigFix data (see Explorer). When evaluating the Session Relevance using the BigFix Explorer, it is now possible to filter the Session Relevance results.

For details, see Session Relevance.

It is now possible to start/stop/restart VMs on AWS, Google and Azure via BigFix

The BigFix cloud plugins now have the ability to manage the cloud instances using power commands.

For details, see Cloud plugins Commands (on page 336).

AWS cloud plugin connection now available in FIPS mode

The Amazon Web Services (AWS) cloud plugin can be configured to leverage the FIPS mode connection, to comply with encryption algorithms prescribed by the FIPS standard.

For details, see Configuring cloud plugins (on page 279).

You can now configure a persistent connection for a Relay to Relay communication

BigFix Platform allows you to establish a persistent connection for a Relay with its parent Relay; this will facilitate the BigFix operations in complex network environments.

For details, see Relay-Relay persistent connection (on page 371).

Enabled Microsoft Control Flow Guard on BigFix Server

The BigFix Server leverages the Microsoft Control Flow Guard (CFG) security feature on Windows systems.

For details, see Enabling Microsoft Control Flow Guard on BigFix Server (on page 1189).

Added Subject Alternative Name field to the Client Certificate

The Subject Alternative Name X509v3 standard extension is added to the client certificate of the BigFix Agents. Its value corresponds to the hostname of the computer where the BigFix Agent runs. This allows the BigFix client certificates to adhere to industry standards as it relates to Subject Alternative Name.

For details, see Subject Alternative Name (on page 1204).

Added BESAdmin command to return the BigFix certificate bundle

With the `getcertificatebundle` BESAdmin command, you can export the complete BigFix certificate bundle. In the bundle, there are all the certificates for all authorized chains in the masthead. This allows you to provide the full certificate chain to tools or entities that request it for validation.

For details, see BESAdmin Windows Command Line and BESAdmin Linux Command Line.

CVEs details in the Server, Relay, Client and Console Upgrade Fixlets

Starting From BigFix Version 11.0.3, the Fixlets in the BES Support site which upgrade the BigFix Console and BigFix Platform components (Server, Relay and Client) will show a list of CVE IDs. These refer to the vulnerabilities affecting the previous level of the component, and resolved by the current one. This information will be found on the Details Tab, in the CVE ID field.

Microsoft Entra ID configuration using certificates

Starting from BigFix Version 11.0.3, BigFix Platform allows you to configure Microsoft Entra ID as Identity Provider using a certificate in addition to a client secret.

For details, see Integrating with Microsoft Entra ID (on page 75).

Enhanced Agent log file records with date/time stamps

BigFix Platform allows you to specify the desired Agent log file format, by using a new setting named `_BESClient_Log_TimestampsDetail`. This will include timestamps in every line of the agent log file.

For details, see List of settings and detailed descriptions (on page 404).

Limit number of targets when submitting action via REST API

When issuing an action via REST API, the maximum number of targets set in the “targetBySpecificListLimit” parameter (default value 10000) is considered; if exceeded, an “HTTP 413 Content Too Large” error response is returned. For details, see action.

On new clients, avoid the creation of client settings referring to deleted NMO's

New clients will no longer create settings that refer to Non-Master Operators that have already been deleted, reducing the

workload on these clients and avoiding useless references in their log file.

Audit Trail Cleaner update

The Audit Trail Cleaner tool was updated to allow you to remove the old files uploaded by the Archive Manager on the BigFix Server.

For details, see Audit Trail Cleaner.

Improved user experience in Web Reports as it relates to reauthentication

The number of reauthentication operations needed in Web Reports is now reduced. The behavior can be controlled via a new configuration setting named `ReAuthenticationEnabled`.

For details, see Performing the reauthentication.

Removal of obsolete Fixlets and Tasks from BES Support

Obsolete Fixlets and Tasks were removed from the BES Support content.

For details, see Removal of obsolete Fixlets and Tasks from BES Support content.

Inspector Updates

- New client inspectors named "case insensitive posix regex", "case insensitive posix regular expression", "posix regex" and "posix regular expression" were added to fully support POSIX compliant Regular Expressions. They will use the Boost library version 1.78.0, both for Windows and UNIX operating systems, since Boost is declared to be POSIX-Extended compliant. For details, see regular expression.

- New client inspector properties:
  - "rtt of": the round-trip time (RTT) of the TCP socket connections in the "ESTABLISHED" state. For details, see socket.

Added Support for BigFix Server

Added support for BigFix Server running on Windows Server 2025 x86 64-bit.

Added Support for BigFix Console

Added support for BigFix Console running on Windows Server 2025 x86 64-bit.

Added Support for BigFix Relay

Added support for BigFix Relay running on:

- Ubuntu 24.04 LTS x86 64-bit
- Windows Server 2025 x86 64-bit.

Added Support for BigFix Agent

Added support for BigFix Agent running on:

- macOS 15 ARM/x86 64-bit
- Ubuntu 24.04 LTS
- Raspberry Pi OS 12 32-bit
- Red Hat Enterprise Linux 9 on s390x
- Rocky Linux 10 x86 64-bit
- Windows Server 2025 x86 64-bit

Library and driver upgrades

- The libcURL library was upgraded to Version 8.9.1.
- The Microsoft ODBC Driver was upgraded to Version 17.10.6.
- The OpenSSL library was upgraded to Version 3.2.2.

Patch 2

New BigFix Explorer component to extend the power of BigFix

With this release BigFix Platform introduces BigFix Explorer.

This new component provides REST API access to BigFix data in an easier, more scalable and more resilient way.

It is designed to detach the datastore engine from BigFix Console and Web Reports and use it as a standalone service to query the BigFix Server data using Session Relevance expressions.

For details, see Introduction and Explorer.

Added Microsoft Entra ID as Identity provider

BigFix Platform now allows to use Microsoft Entra ID (formerly known as Microsoft Azure AD) as Identity Provider, to support the Single-Sign-On and Multi-Factor-Authentication use cases. This allows the user directory on Entra ID to be used for accessing the BigFix Console, Web Reports and WebUI.

For details, see Integrating with Microsoft Entra ID (on page 75) and Identity Provider Permissions.

IPv6 validation completion

Extensive validation scenarios were run, and problems addressed, to ensure that BigFix Platform components run both in IPv6-only and concurrent IPv6+IPv4 configuration.

For details, see IPv4 and IPv6 protocols concurrent support.

Removed init.d dependencies for RHEL and SUSE platforms

With this release BigFix services are no longer dependent on init.d for RHEL, SUSE and derived platforms.

For details, see Managing the BigFix Services.

Enhanced Archive Manager capabilities

A new set of REST APIs named "Archive Manager" is available to list, retrieve or remove the files uploaded by the Archive Manager on the BigFix Server. Additional customization is also possible with respect to Archive Manager file cleanup policy.

For details, see Archive Manager and Automatic Server Clean Up (on page 968).

VMware Plugin enhancements

The VMware Plugin has been extended with inspectors and action commands to improve the management capabilities for both host and guest systems.

For details, see Introduction to Cloud Plugins (on page 266), Configuring cloud plugins (on page 279), VMware Asset Discovery Plugin Inspectors (on page 313) and VMware Plugin Commands (on page 341).

Inspector Updates

- **Added client inspector type to support 128-bit signed integers**
  - A new client inspector type named "large integer" was created to support the 128-bit signed integers. For details, see large integer.
- **Added client inspector constructor to support BigFix Explorer**

    - A new client inspector constructor named "explorer service" was created to provide access to the BESExplorer service, if available on the local system. For details, see service.
- **Added client inspector properties**
    - New client inspector properties named "display name of" and "linux of" were added to return the OS name in a human readable format and to verify if the computer is running Linux. For details, see operating system.

Added Support for BigFix Server on Windows and Linux with Google Cloud SQL for Microsoft SQL Server 2022 database

Starting from Patch 2, BigFix Server on Windows Server (2019 or later) and on Linux Red Hat 9 supports Google Cloud SQL for Microsoft SQL Server 2022 database.

Added Support for BigFix Relay

Added support for BigFix Relay running on:

- Amazon Linux 2023 x86 64-bit.
- Rocky Linux 9 x86 64-bit.

Added Support for BigFix Agent

Added support for BigFix Agent running on OpenSUSE Leap 15.6 x86 64-bit.

Library and driver upgrades

- The jQuery library was upgraded to Version 3.7.1.
- The libcURL library was upgraded to Version 8.6.0.
- The Microsoft ODBC Driver was upgraded to Version 17.10.5.1.
- The OpenSSL library was upgraded to Version 3.1.5.

- The sqlite library was upgraded to Version 3.45.1.
- The zlib library was upgraded to Version 1.3.1.

Patch 1

Added support for BigFix Server on Linux with Microsoft SQL Server database

Starting from Patch 1, BigFix Server on Linux Red Hat 9 supports Microsoft SQL Server 2019 and 2022 (Enterprise and Standard are recommended). Only the SQL Authentication is supported. DSA environments are not supported on the BigFix Linux Server with Microsoft SQL database.

For details, see Installing the Server with MS SQL, Installing and configuring MS SQL, Installing Web Reports Standalone with MS SQL and Installing the WebUI Standalone with MS SQL.

Added support for BigFix Server on Red Hat Linux 9 with DB2 database

Red Hat Linux 9 is now supported with minimum BigFix Server version 11.0.1 and DB2 11.5.9.

Added support for BigFix Plugin Portal on Linux RHEL 9

Added support for BigFix Plugin Portal running on Red Hat Enterprise Linux (RHEL) 9 x86 64-bit.

License Overview dashboard "Device subscription by product" section changes

The "Device subscription by product" section of the dashboard now shows, for each product, the labels "Allocated" and "Actual". An Info Button is present next to each label.

For details, see License Overview dashboard.

Manual synchronization of the license updates

On Windows server, a new option named "syncmastheadandlicense" is provided to synchronize

the updated license with the masthead using the BigFix Administration Tool.

For details, see BESAdmin Windows Command Line.

Added client inspector type to support for unsigned integers

A new client inspector type named "uinteger" was created to support the unsigned integers.

For details, see uinteger.

Added support for the new "Device" and "User" metrics available for BigFix Workspace licenses

When BigFix Workspace licenses are allocated on any serial number, the "Device" and "User" tags are now reported in the license.crt file and properly managed in the License Overview Dashboard.

For details, see bes product.

Added support for BigFix Agent

Added support for BigFix Agent running on:

- Windows 11 ARM64
- Windows 11 23H2 ARM64
- Windows 11 24H2 ARM64
- Windows 11 25H2 ARM64
- Windows 11 26H2 ARM64
- VIOS 3.1.3

Library upgrades

- The ICU library was upgraded to Version 73.2 (54.2 on Tiny Core Linux).
- The libcURL library was upgraded to Version 8.4.0.

- The libssh2 library was upgraded to Version 1.11.0.
- The OpenSSL library was upgraded to Version 3.1.4.

Version 11

OpenSSL v3

BigFix Platform 11 uses OpenSSL v3 in all its components to ensure maximum protection of network traffic. More in detail, the version of the library is 3.1.1. Aside from the general benefits in the security area, the presence of OpenSSL v3 has the two following major consequences:

- Any HTTPS communication in which at least one of the two parties is represented by a BigFix Platform 11 component must use TLS 1.2 as minimum protocol version.
- SHA1 is no longer used as hashing signature algorithm to validate TLS communication as well as all BigFix content and actions (SHA1 is still supported as hashing for file downloads).

For more details, see BigFix Platform V11 Overview Page.

SHA-384 support

BigFix 11 uses a stronger hash based on SHA-384 as cryptographic digest algorithm for all digital signatures to validate TLS communication and all BigFix content and actions at every step. This change does not affect the hash used to verify downloaded files which can still be SHA-1 or SHA-256.

SHA-256 hash signatures are still supported but you have also the option of enforcing usage of SHA-384 only to comply with specific security requirements.

For more details, see BigFix Platform V11 Overview Page.

TLS 1.3 support

BigFix Platform now supports TLS 1.3 for HTTPS communications among the BigFix components, maintains the support of TLS 1.2 and no longer supports TLS versions lower than 1.2.

By default, BigFix Platform 11 supports both TLS 1.2 and TLS 1.3, while – due to the upgrade to OpenSSL v3 – it does no longer support TLS 1.1 or below.

For more details, see BigFix Platform V11 Overview Page.

Relay Drive Space Protection From Downloads

BigFix Platform adds now the capability to prevent the BigFix Relay ActiveDownloads folder from filling up, by using a new setting named `_BESRelay_Download_ActiveDownloadsMaxSizeMB`, which represents the maximum size, specified in MB, that the folder can reach.

For details, see Managing Downloads (on page 1099).

Perl Regular Expressions for non-Windows platforms support

The Perl Compatible Regular Expressions (PCRE) syntaxes, introduced with BigFix Platform 10.0.8 and available on the Windows client, are now also supported on several non-Windows platforms such as Debian, Mac, Raspbian, Red Hat, SUSE, Solaris Intel and Ubuntu.

For details, see regular expression.

Plugin Portal - Optimized devices data serialization

Plugin Portal optimization in terms of memory usage of the plugin portal machine as well as in the evaluation time of fixlet and analysis, with this leading to an increased responsiveness in returning data and executing actions on discovered devices.

New set of REST APIs

BigFix Platform 11.0 now supports a new set of Rest APIs that enable exploiters such as the BigFix WebUI to access the Download status of the actions. These Rest APIs allow also to re-submit failed downloads.

For details, see Action.

Added support for BigFix Console

The BigFix Console Version 11.0 adds support for:

- Windows 11 23H2
- Windows 11 24H2
- Windows 11 25H2

Added support for BigFix Relay

The BigFix Relay Version 11.0 adds support for:

- AIX 7.3
- Raspbian 11
- Tiny Core 13
- Tiny Core 14
- Windows 11 23H2
- Windows 11 24H2
- Windows 11 25H2

Added support for BigFix Agent

The BigFix Agent Version 11.0 adds support for:

- Debian 12 x86-64
- MacOS 14 ARM/x86 64-bit
- OpenSUSE Leap 15.4 x86-64
- OpenSUSE Leap 15.5 x86-64
- Windows 11 23H2 x86-64

- Windows 11 24H2 x86-64
- Windows 11 25H2 x86-64
- Windows 11 26H2 x86-64

Added support for new database level

- Microsoft SQL Server 2022 support
- Microsoft SQL Server 2022 deployed in a docker container

For details, see Installing a server with remote database deployed in a docker container and Database requirements.

Note also that, on BigFix Platform 11.0:

- The minimum supported SQL Server version is 2014 as Microsoft SQL Server 2012 is no longer supported.
- DB2 is a prerequisite for the installation of the BigFix Server on Red Hat Linux. DB2 is not distributed with BigFix 11. For existing BigFix 9 and 10 customers with a DB2 entitlement, the entitlement remains. For new customers on BigFix 11, a DB2 license must be acquired. The BigFix team is considering adding, in the near term, the possibility to utilize Microsoft SQL Server for BigFix deployments on Linux.

For information about database requirements, seeInstallation requirements for DB2 database products and Database requirements for information about the DB2 versions supported by BigFix.

Operating systems support matrix has been updated

In particular, for some platforms the minimum operating system version supported has changed. To see which operating system versions are supported, refer to the V11 system requirements page available at: BigFix Support Matrix.

Several libraries are upgraded to a newer version:

Library	V11 Library Version
jQuery	Version 3.6.4
libcURL	Version 8.1.2
Microsoft Visual C++ Redistributable library	Version 2019
OpenSSL	Version 3.1.1
OpenLDAP	Version 2.6.4
SQLite	Version 3.41.2
zlib	Version 1.2.13
AWS SDK	Version 1.44.165
Azure SDK	Version 1.0.0 (with azidentity v1.2.0)
GCP SDK	Version 0.105.0
VMWare SDK	Version 0.30.0

Chapter 2. BigFix Site Administrator and Console Operators

In BigFix there are two basic classes of users.

The Site Administrator

The Site Administrator is responsible for installing and maintaining the BigFix software, and to run administrative tasks that globally affects the environment such as site-level signing keys management. There is only on Site Administrator for a BigFix environment. For more information, see The Site Administrator *(on page 41)*.

The Console Operators

They are the user of BigFix who access the BigFix Console and, if authorized, the WebUI. They can be **Master Operators (MO)**, the user with Administrators of the BigFix Console, or **Operators (NMO)**, the day-to-day managers of their own domains. While, Master Operators can create other operators and assign management rights, Operators can not. For more information, see Introducing Operators.

Note: When defining an operator, ensure that the user name does not contain any of the following characters: `:`, `@`, and `\`.

The Site Administrator

The site administrator has the following primary responsibilities.

Obtaining and securing the Action Site Credentials

To install BigFix, the site administrator must generate a private key, receive a license certificate from HCL, and create a masthead with the digital signature and configuration information. This is a special key and must be used only for site-level tasks such as:

- Setting global system options
- Editing Mastheads
- Administering Distributed Server Architecture (DSA)

Preparing the Server

The BigFix Server must be correctly set up to communicate externally with the Internet and internally with the Clients. The Server also needs to be configured to host the BigFix database (or another computer can be used as the SQL Server database).

Installing the various components

The site administrator installs the BigFix Client, Server, Relay, and Console modules, and configures the credentials of the first master operator who will connect to the console to define the license subscriptions, gather content from subscribed sites, and define the BigFix network, the roles and the other operators.

The site administrator sets up and administers multiple BigFix Servers in a Disaster Server Architecture (DSA) for doing automatic BigFix server failover and failback.

Maintaining the Server

The BigFix server runs an SQL Server database and several specific services, such as running the Diagnostic Tool and the Administration Tool. Standard maintenance tasks such as upgrades or fixes are managed using Fixlet technology or can be performed manually by the site administrator.

For day-to-day console operations, the site administrator must create a master operator key.

The Site Administrator cannot:

- Access the BigFix Console.
- Create operators in addition to the one created during installation.
- Access the BigFix WebUI.
- Run BigFix Queries.

The Console Operators

There are two types of Console operators:

Master Operators (MO)

They are the administrative users of the Console. They have access to all the computers defined in the BigFix environment and the authority to create and manage other console operators. Any master operator can create, assign, and revoke management rights that allow operators to deploy actions.

Operators or Non-Master Operators (NMO)

They manage the day-to-day BigFix operations, including Fixlet management and action deployment, against a subset of computers they are allowed to manage by the master operator. They cannot create other operators and cannot assign management rights.

By default the Console operators cannot:

- Access the WebUI, unless the **Can use WebUI** permission is set to **YES**.
- Submit BigFix queries, unless both **Can use WebUI** and **Can Submit Queries** permissions are set to **YES**.

These and other permissions can be set by a master operator in the Permissions area of the Details tab of the operator's description. For more information about operators rights, see Mapping authorized activities with permissions *(on page 53)*.

Best practices

The following tables describe when to use a Master Operator (MO) or a Non-Master Operator (NMO) role.

Table 1. Master Operator

MO
They are the administrative users of the BigFix Console. They have access to all the computers defined in the BigFix environment and the authority to create and manage

Table 1. Master Operator (continued)

MO
other console operators. Any master operator can create, assign, and revoke management rights that allow operators to deploy actions.
They create users/operators/roles.
They create custom sites.
They create custom content that can be seen by all operators and will likely be used on most computers.
They issue certain policy actions that pertain to the entire BigFix environment. They keep number of actions to a minimum as this adds to the Master Action Site size.
They manage site subscriptions.
They create the retrieved properties.
They hide content globally.
They activate global analyses – for all master operators and non-master operators.

Table 2. Non-Master Operator

NMO
They manage the day-to-day BigFix operations, including Fixlet management and action deployment, against a subset of computers they are allowed to manage by the master operator. They cannot create other operators and cannot assign management rights.
They issue actions, such as deploying patches.
They make REST API calls.
They deploy/activate/deactivate Fixlets, Tasks, Baselines, Analyses.
They create custom content for a specific purpose.

Table 2. Non-Master Operator (continued)

NMO
They activate local analyses – based on the non-master operator's administered computers.

Different ways to define a Console Operator

There are different ways to add console operators, assigning them roles or granting permissions to view or manage specific computers and sites.

- You can add single operators at any time by selecting the **Tools > Create Operator** item or by right clicking in the operators work area and selecting **Create Operator** as described in Adding Local Operators *(on page 45)*.
- If you are using Active Directory or a generic LDAP, you can add previously defined users by selecting the **Tools > Add LDAP Operator** item or by right clicking in the operators work area and selecting **Add LDAP Operator** as described in Adding Identity Provider Operators *(on page 84)*.
- You can also associate an LDAP group to an existing role, in this way, with just one click, you add an operator for each user specified in the LDAP group and you associate that operator to the role. For more information about this capability, see Associating an Identity Provider Group *(on page 91)*.

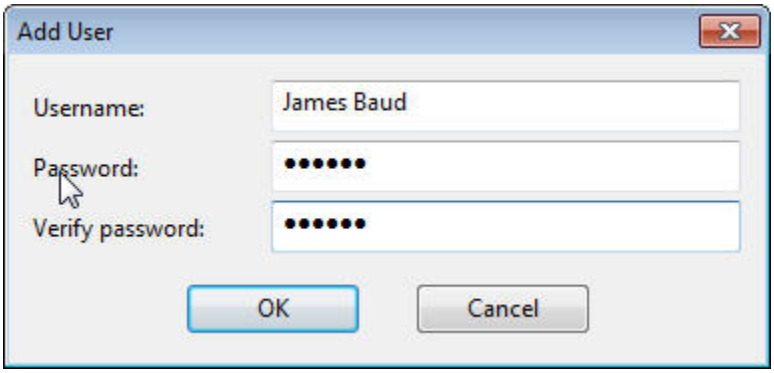
Note: For LDAP operator and LDAP Group an Active Directory or LDAP directory must first be added to BigFix.

Adding Local Operators

You can create accounts for operators that access the console using the local BigFix account.

To add a local operator perform the following steps:

1. Click the **Tools > Create Operator** menu item or right click in the operators work area and select **Create Operator**. The **Add User** dialog appears.



2. Enter the **Username** of the person you want to designate as a publisher or operator.
3. Create a **Password** and retype it for confirmation. When you give the keys to your operators, they can change their passwords if they want.
4. Click **OK**. The **Console Operator** window opens.
5. From the **Details** tab, assign operator permissions.



You can control the default settings by using the **defaultOperatorRolePermissions** option in **Advanced Options** of the BigFix Administrative tool. For details, see List of advanced options (on page 1169).



where:

Master Operator

Specifies if the operator is a Master operator or not.

Show Other Operator's Actions

Specifies if the operator can see the actions submitted by other operators.

Note: An operator with the **Show Other Operators' Actions** permission can see the action only in the following cases:

- If he is the owner of the action.
- If another operator submitted the action on at least one of his administered computers, and this computer

is administered by both operators. In this case, the information is available only when the computer reports back the data to the BigFix server.

Stop Other Operator's Actions

You can specify if a non-master operator (NMO) can stop the actions submitted by other non-master operators. For more details, see Stop Other Operator's Actions feature *(on page 50)*.

Can Create Actions

Specifies if the operator can create actions.

Note: The **Can Create Actions** permissions are required for a Non Master Operator to remove computers from the database.

Can Lock

Specifies if the operator can lock targets. This is a way to prevent other operators from running activities on those targets.

Can Send Refresh to Multiple Clients

Specifies if the operator can run a refresh on more than one target concurrently by clicking the **Refresh** button on the BigFix console.

Can Submit Queries

Specifies if the operator can submit BigFix Query requests from the WebUI user interface.

Custom Content

Specifies if the operator can run activities that require the creation of custom content.

Note: A Non Master Operator with the **Custom Content** and **Can Create Actions** permissions, can only edit/delete existing computer settings but cannot add new computer settings.

Unmanaged Assets

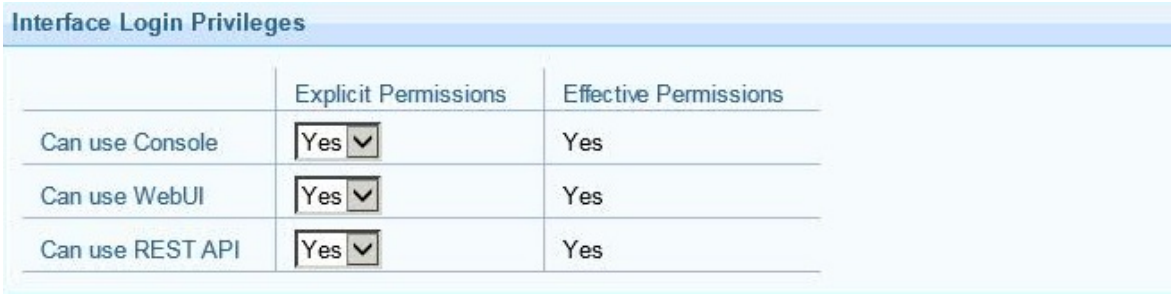
Specifies if the operator can manage assets on which no BigFix component is installed.

An **Explicit Permission** is a permission that you are assigning to the operator. An **Effective Permission** is a permission that is inherited from the roles that the operator is assigned to. If the values displayed in **Explicit Permission** and **Effective Permission** for the same permission are different, the less restrictive permission is applied.

You also decide to influence the ability of the operator to trigger restart and shutdown as Post-Action or to include them in BigFix Action Scripts.



Depending on the configuration that you set for a specific operator for shutdown and restart, the radio button in the Take action panel might be disabled for that operator. This configuration has no effect on actions with type other than BigFix Action Script. You can also set permissions to access the BigFix user interfaces.



6. From the **Administered Computers** tab, you see the list of computers that this operator can manage. This list is populated after the computers that satisfy the criteria specified in the **Computer Assignments** tab report back their information to the BigFix server.
7. From the **Assigned Roles** tab, select the roles to apply to this operator.
8. From the **Sites** tab, assign the sites you want this operator to have access to.
9. From the **Computer Assignments** tab, specify the properties that must be matched by the computers that the operator can manage. For master operators, all the computers are assigned.
10. From the **WebUI Apps** tab, specify the WebUI Applications that the operator is allowed to access.
11. To save the changes click **Save Changes**.

At any time, you can also convert a local operator to an LDAP operator. To do so, follow these steps:

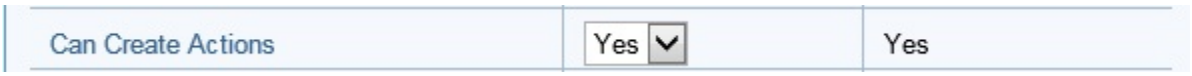
1. From any list of local operators, right click on the operator you want to convert.
2. From the context menu, select **Convert to LDAP Operator**.

Stop Other Operator's Actions feature

A non-master operator (NMO) can stop the actions submitted by other non-master operators if specific conditions are satisfied.

Requirements for the non-master operator (NMO) launching the action (the issuer)

This NMO must have at least the possibility to create and submit an action and some computers assigned, either inherited from an assigned role or explicitely assigned, but there are no other specific restrictions or requirements, related to this feature, for him.



Requirements for the non-master operator (NMO) stopping the action (the stopper)

1. This NMO must have both the **Show Other Operators' Actions** and the **Stop Other Operator's Actions** effective permissions set to **Yes**.



2. This NMO must have a set of assigned role names which is either identical or a superset of those of the issuer. If the issuer has no roles assigned, it is not required for the stopper to have roles assigned as well.

   Note: The role comparison is based only on the name of the assigned roles.

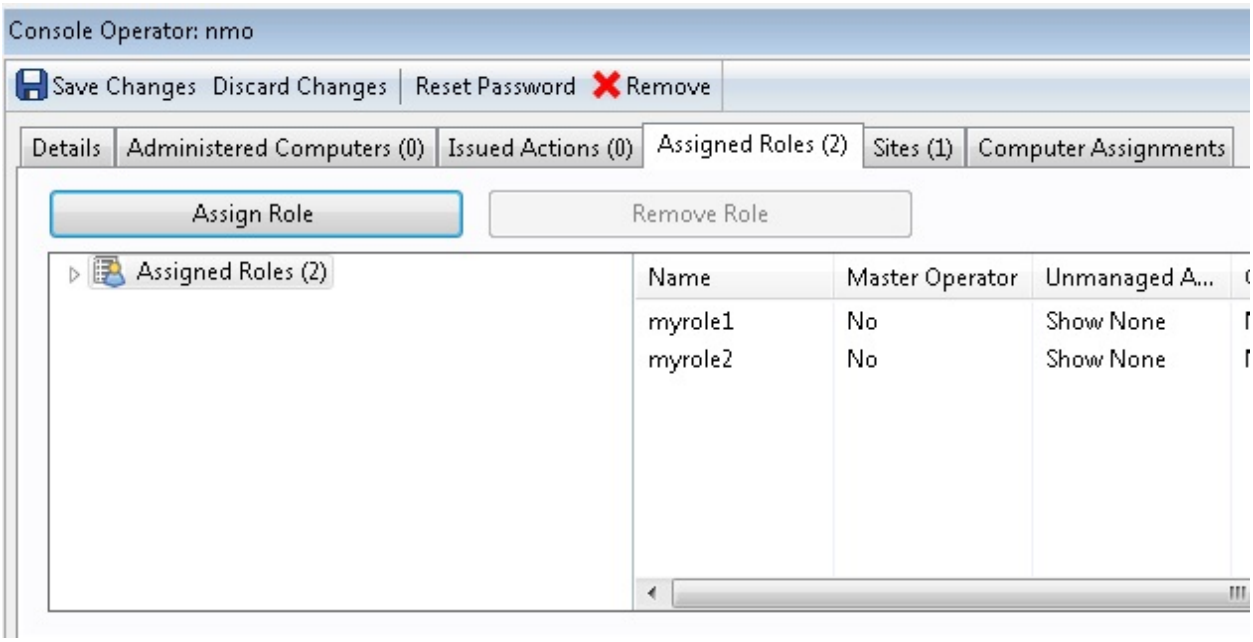
3. This NMO must have a set of Explicit Computer Assignments Definitions which is either identical or a superset of those of the issuer. The issuer can even have no computers explicitly assigned and, in this case, it is not required for the stopper to have explicit assignments as well. The same rule is valid also for roles.

   Note: The Explicit Computer Assignments Definitions is not the list of computers resulting from the computer assignment, but the definition of those computer assignments.

Note: The assignments inherited from the assigned roles (specified in the Assigned Roles tab of the NMO) and those explicitly assigned (specified in the Computer Assignments tab of the NMO) are evaluated separately.

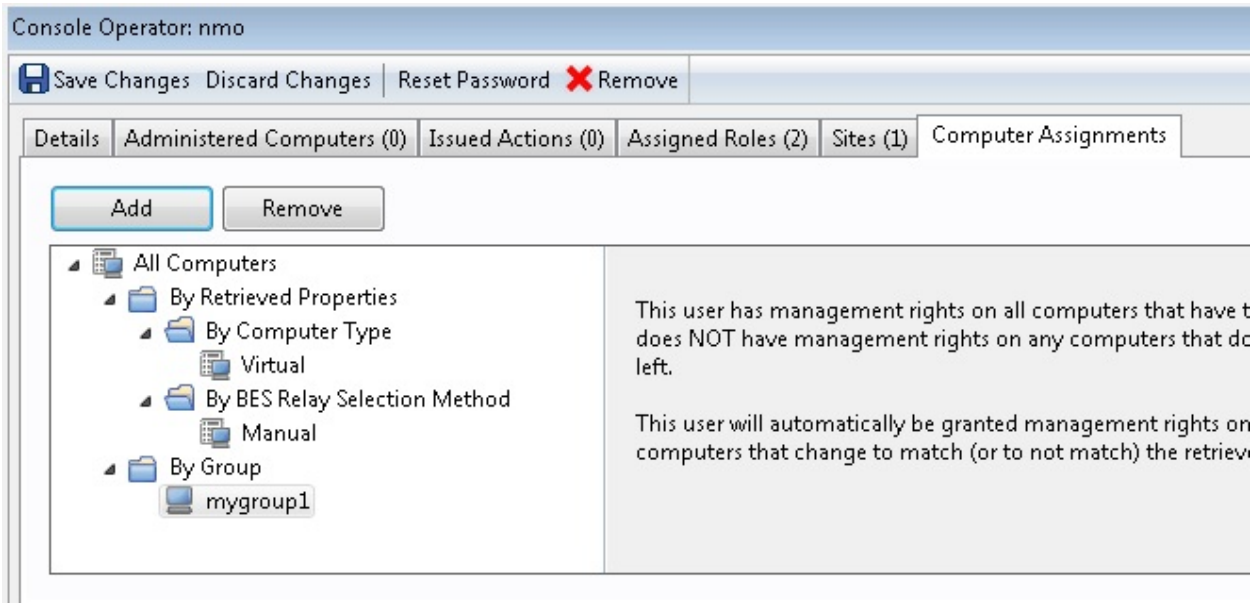
Now follow some examples of **Assigned Roles** and of **Computer Assignments**.

The following screenshot shows multiple roles (myrole1 and myrole2) assigned to an NMO.



The following screenshot shows multiple computer assignment definitions assigned to an NMO which are:

- By 'Computer Type' property -> Virtual
- By 'BES Relay Selection Method' property -> Manual
- By Group -> mygroup1



Mapping authorized activities with permissions

The following table shows which activities you can, cannot or could, under specific conditions, allow an operator to do by assigning permissions in the Details tab of the Operator Definition.

For more information about the operator specific permissions, see Adding Local Operators (on page 45).

Table 3. Mapping of authorized activities with operator permissions

Activities	Operator
Manage Fixlet Sites	No
Change Client heartbeats	No
Create Fixlets	If **Custom Content** is set to YES
Create Tasks	If **Custom Content** is set to YES
Create Analyses	If **Custom Content** is set to YES
Create Baselines	If **Custom Content** is set to YES

Table 3. Mapping of authorized activities with operator permissions (continued)

Activities	Operator
Activate/Deactivate Analyses	Administered
Take Fixlet/Task/Baseline Action	Administered
Take Custom Action	If **Custom Content** is set to YES and **Can Create Actions** is set to YES
Stop Actions	Administered
Manage Administrative Rights	No
Manage Global Retrieved Properties	No
View Fixlets	Administered
View Tasks	Administered
View Analyses	Administered
View Computers	Administered
View Baselines	Administered
View Computer Groups	Administered
View Unmanaged Assets	Administered
View Actions	Administered
Make Comments	Administered
View Comments	Administered
Globally Hide/Unhide	No
Locally Hide/Unhide	Yes

Table 3. Mapping of authorized activities with operator permissions (continued)

Activities	Operator
Use Wizards	If **Custom Content** is set to YES
Remove computer from database	If **Can Create Actions** is set to YES
Create Manual Computer Groups	If **Can Create Actions** is set to YES
Delete Manual Computer Groups	If **Custom Content** is set to YES and Administered
Create Automatic Computer Groups	If **Custom Content** is set to YES
Delete Automatic Computer Groups	If **Custom Content** is set to YES and Administered
Create Server Based Groups	If **Can Create Actions** is set to YES
Delete Server Based Groups	If **Can Create Actions** is set to YES and Administered
Create Custom Site	No
Modify Custom Site Owners	No
Modify Custom Site Readers/Writers	Site Owners
Create a Master Operator	No
Use the WebUI	If **Can use WebUI** is set to YES
Submit BigFix Query	If both **Can use WebUI** and **Can Submit Queries** are set to **YES**

Administered: The operator must own or have permissions.

Requires Custom Authoring: Granted by the site administrator through the console.

Operators and analysis

Operators have various rights and restrictions when activating and deactivating analysis.

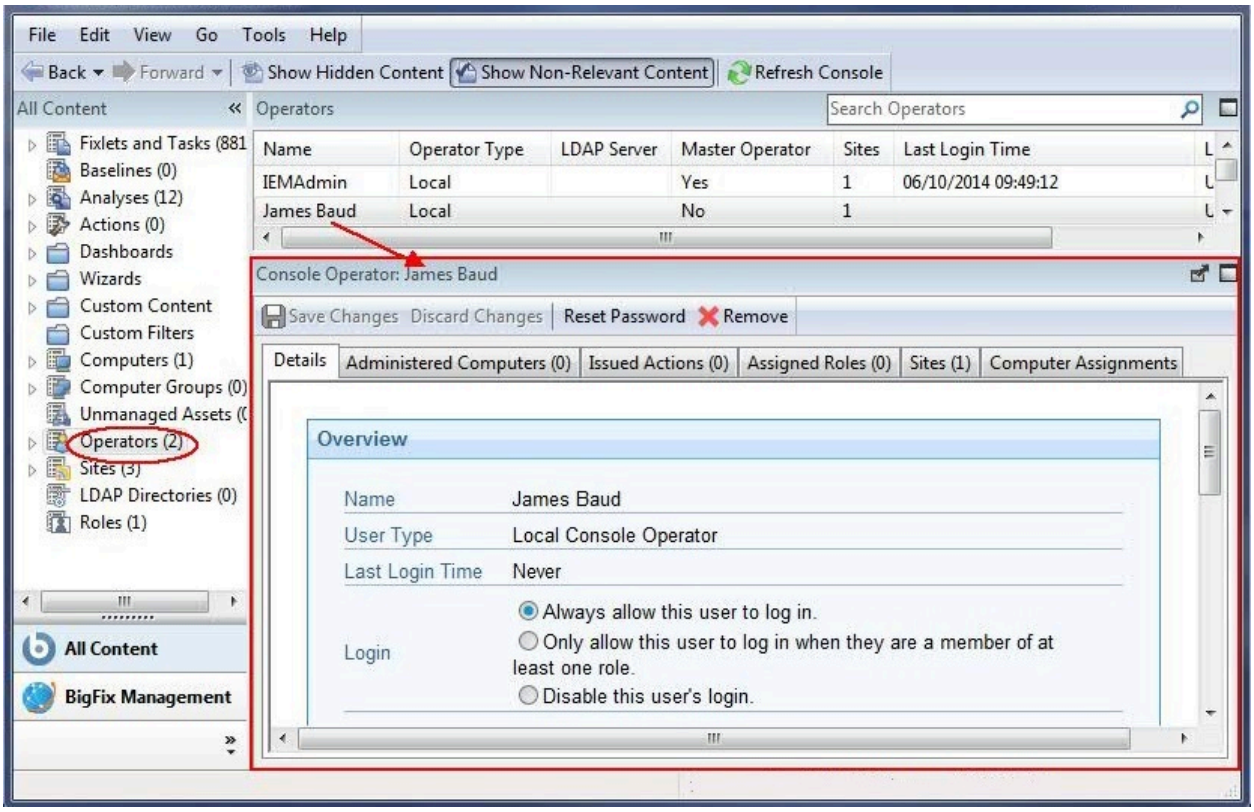
- Ordinary operators cannot deactivate an analysis activated by other operators on computers they administer.
- Master Operators cannot directly activate custom analysis authored by ordinary operators. They can, however, make a copy of an analysis and activate the copy.

Monitoring Operators

If you are a master Operator (you must have a correctly authorized user name created with the BigFix Administration Tool), you can monitor what other operators are doing and what computers they are authorized to administer.

Each operator is represented by, among other attributes, a **Name, User Type** and **Login type**. To view the list of Console Operators, select the **All Content** Domain and then click the node labeled **Operators** from the Domain Panel. In the List Panel on the right, all the current Operators are listed.

Click any operator from the List Panel to open the **Operator** work area.



There are several tabs to choose from:

- **Details:** Describes the operator by name and type and lets you select a login type. This is also where you can view and alter operator permissions.
- **Administered Computers:** Presents a list of computers that are currently assigned to the selected console operator.
- **Issued Actions:** Presents a list of actions that have been issued by the selected console operator.
- **Assigned Roles:** Displays the currently assigned roles, and lets you reassign them.
- **Sites:** Displays the sites currently assigned to this operator, and lets you reassign them. If the site is a custom site, you can also set Read/Write/Owner permissions.
- **Computer Assignments:** Lists the properties that must be matched by the computers that the operator can manage. If you specify a property to be matched, any time a computer is changed to match that property, it is added to the list of computers

assigned to the operator. On the other hand, if a computer is changed not to match that property, that computer is removed from the list.

This tab is available only for not-master operators.

Chapter 3. Integrating with LDAP or Microsoft Entra ID

You can add Identity Providers associations to BigFix.

That allows you and other users to log in to the console using those credentials. The same advantage applies also to Web Reports.

Follow the instructions provided in the next topics to learn how to integrate BigFix with a Generic LDAP or with Active Directory or with Microsoft Entra ID.

Note: If you are using SSL to integrate BigFix with a Generic LDAP server or with an Active Directory server, take into account that BigFix does not support the SSL connection to LDAP servers or Active Directory servers through a load balancer or a DNS alias.

Note: On Windows platforms, Kerberos authentication is only supported if you integrate BigFix with a Generic LDAP, otherwise NTLMv1 authentication will be used.

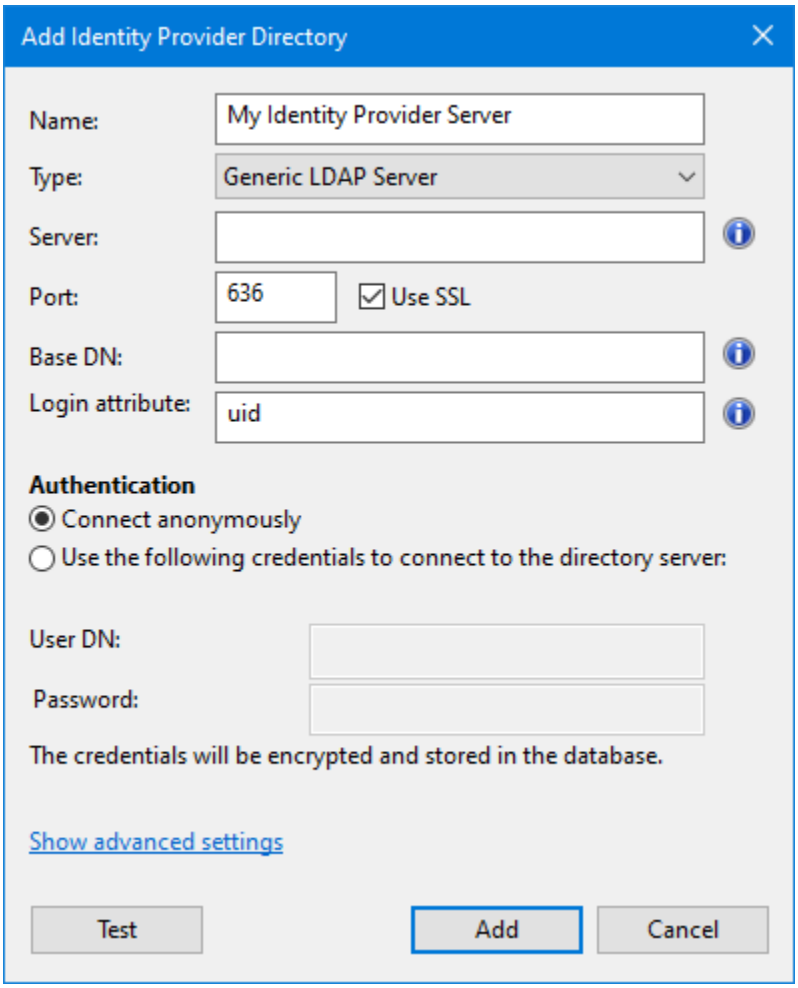
After you completed the steps to integrate with one of these two types of LDAP or with Microsoft Entra ID, you can associate Identity Provider operators or groups to BigFix Console operators or roles as described in Adding Identity Provider Operators (on page 84) and Associating an Identity Provider Group (on page 91).

Integrating with a Generic LDAP

How to configure the integration with a Generic LDAP by adding an existing LDAP domain to the console.

Perform these steps:

1. From the **Tool** menu, select **Add Identity Provider Directory** or right click in the work area and then select **Add Identity Provider Directory**. The **Add Identity Provider Directory** dialog appears.



2. Provide a name and from the Type pull-down, make sure **Generic LDAP Server** is selected. Note that no **global catalog** option is available on generic LDAP servers.
3. Fill in the information pertaining to your LDAP installation. Under **Server**, enter the host name or IP Address of the server.
4. Enter the port number, typically 636 if you are using Secure Sockets Layer (SSL).
5. Enter the base distinguished name (**Base DN**), of the form `dc=example,dc=com`.
6. Click the button to **connect anonymously** or to **use credentials**. If you choose to connect using credentials, enter your **User DN** and **password**.

7. Click **Test** to ensure you have entered your information correctly and a connection can be made to your LDAP.
8. If you want to include user or group filters, click the **Show advanced settings** link. After specified, all further LDAP searches will be subject to the appropriate filter.
9. Click **Add** to complete the LDAP setup.

Your LDAP Server is now configured and available for use in the console.

Integrating with Active Directory

You can use Microsoft Active Directory (AD) to handle authentication on BigFix.

That allows you and other users to log in to the console using your Active Directory credentials, taking advantage of your existing authentication policies. The same advantage applies also to Web Reports.

Starting from BigFix Platform Version 9.5 Patch 14, integration with Active Directory that is configured with LDAP channel binding and LDAP signing is supported.

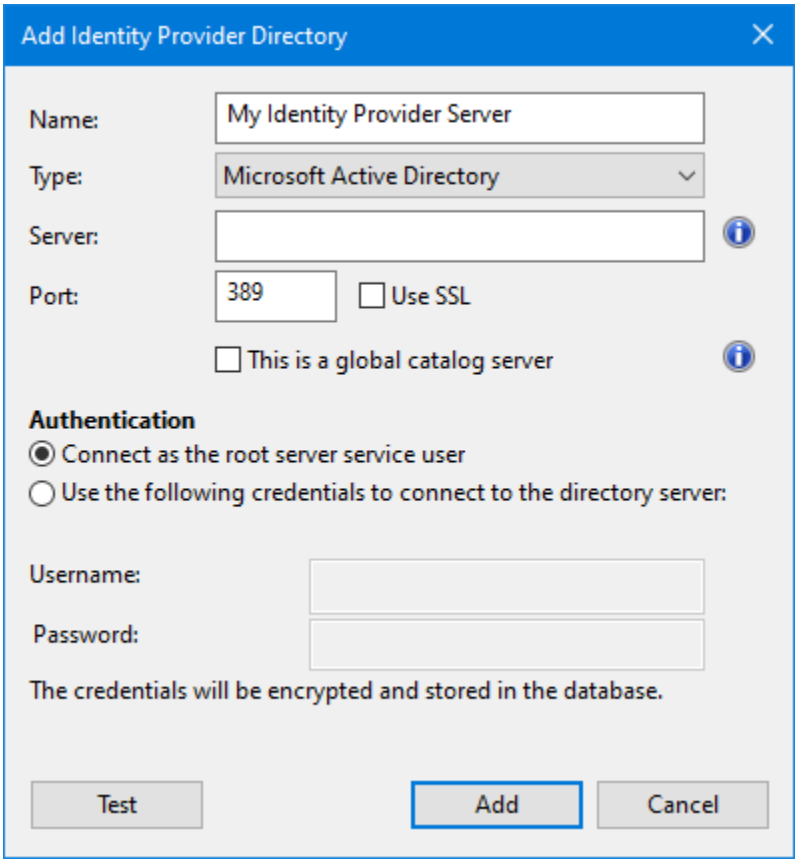
Note: On Windows platforms, the inspector that manages the calls to the Active Directory causes an ephemeral port to be allocated on the User Datagram Protocol (UDP), in addition to the 52311 port already required for the BESClient process. This port is visible in the output of the netstat -an command.

Integrating the BigFix Windows server with Active Directory

How to add an existing Active Directory to the console.

Follow these steps:

1. From the **Tool** menu, select **Add Identity Provider Directory**. The **Add Identity Provider Directory** dialog displays.



2. Provide a name for the Active Directory and from the Type pull-down, make sure **Microsoft Active Directory** is selected.
3. Under **Server**, enter the host name, IP Address or fully qualified domain name of the server.
4. Click **Use SSL** if you want to configure a secure connection (SSL).
5. To access an entire Active Directory forest, click **This is a global catalog server**. **Note:** If you plan on using the User-based management *(on page 93)* feature introduced in BigFix 11.0.7, you will get more complete data if you do not check that option.
6. Click the button to **connect as the root server service user** or to **use credentials**. If you choose to connect using credentials, enter your Active Directory **Username** and **Password**.

7. Click **Test** to make sure you have entered your information correctly and a connection can be made to your Active Directory server.
8. Click **Add** to complete the Active Directory setup.

Note: When you add an LDAP Server as **Microsoft Active Directory**, ensure that on the LDAP server you have defined the `UserPrincipalName` attribute corresponding to the **User logon name** of each user. This attribute value is used on the BigFix Console for each user authentication.

Your Active Directory Server is now configured and available for use in the console.

AD Domain/Forest Function level for BigFix Server running on Windows only

BigFix 10.0.7 is fully supported in an AD Domain/Forest Function environment with SSL in the following configuration:

- BES Server must be running on Windows only.
- Active Directories Windows 2016 or 2019 defined with 2016 Domain Functional Level and with patch level updated.
- Every Active Directory installed with Global Catalog.
- Enterprise Certification Authority installed on root domain.
- Creation of a certificate for the CA with the following characteristics:
  - In an AD forest with various domain extension (for example BIGFIX.ACME.COM for root domain and CHILD.BIGFIX.ACME.COM for child domain) consider the common part of domain name and create a certificate having a common name that starts with "*" (for example):

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

  - Then, on same certificate, define the DNS and list all AD servers, so the created certificate will have on the Details tab the field "Subject Alternative Name" with the values (for example):

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

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

The certificate must be loaded on All Active Directories of the SAN list.

The scenario was certified with DNS installed on the Active Directories.

User Base Management

If are using the User-based management *(on page 93)* feature, introduced in BigFix 11.0.7, and you are not getting a sufficient level of detail in your idp REST API results and inspector properties for your Active Directory, you can try the following:

1. In the BigFix Console, in the **Identity Provider Directories** view, select your directory and change its **Global Catalog** value to **No**.
2. Restart the BES Root Server service.

Integrating the BigFix Linux server with Active Directory

Configuring Kerberos authentication on Red Hat Enterprise Linux 7

To ensure a secure communication between Linux BigFix server and Active Directory, use the Kerberos protocol.

To integrate the Linux BigFix server with the Windows Active Directory domain using LDAP with Kerberos authentication, perform the following steps:

1. Ensure that the host names and the time service are set correctly in both the Linux BigFix server and the Active Directory server.
2. Install the NSS and PAM libraries.
3. Configure the Kerberos LDAP security and authentication.
4. Modify the local LDAP name.
5. Configure the NSS and PAM libraries.

Preliminary Checks

Before running the integration between the BigFix server running on a Red Hat Enterprise Linux 7 system and the Active Directory server, ensure the following.

- The DNS host names of the Red Hat Enterprise Linux 7 system and the Active Directory server are resolved correctly, by performing the following steps on the Red Hat Enterprise Linux 7 system:
  1. Open the file `/etc/host` and ensure that both DNS host names are specified as fully qualified domain names.
  2. Open the file `/etc/sysconfig/network` and ensure that the host name of the Red Hat Enterprise Linux 7 system is specified as fully qualified domain name.
- The time between the Active Directory and the Linux BigFix server is synchronized. If needed, you can synchronize the time service on the Red Hat Enterprise Linux 7 system and the Active Directory server with the time source server, by performing the following steps:
  1. In the file `/etc/ntp.conf` on the Red Hat Enterprise Linux 7 system, replace the following lines:

     ```
     server hostname
     ```

     with:

     ```
     server time_source_server_name
     ```

     where *time_source_server_name* is the server hostname or IP address of the time source server used to synchronize the time.
  2. When DNS lookups are not reliable, configure the Red Hat Enterprise Linux systems to perform DNS lookups from the Active Directory server by editing the `/etc/resolv.conf` file as follows:

     ```
     domain my.domain.com
     search my.domain.com
     nameserver1 ipaddress1
     nameserver2 ipaddress2
     ```

3. Activate the change on the Red Hat Enterprise Linux 7 system by:
   - Stopping the **ntp** daemon:

     ```
     service ntpd stop
     ```

   - Updating the time:

     ```
     ntpdate Red_Hat_server_IP
     ```

   - Starting the **ntp** daemon:

     ```
     service ntpd start
     ```

4. Synchronize the Active Directory server with the time source server by entering:

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

   where *time_source_server_name* specifies the list of DNS names or IP addresses for the NTP time source with which the Linux server synchronizes. For example, you can specify `time.windows.com` as the NTP time server. When you specify multiple peers, use a space as the delimiter and enclose the names of the peers in quotation marks.
5. On the Active Directory server, run the following command to ensure that the time is synchronized with the time source server

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

6. On the Red Hat Enterprise Linux 7 system configure the **ntpd** daemon to start at system boot:

   ```
   chkconfig ntpd on
   ```

Installing the NSS and PAM libraries

Ensure that the following NSS and PAM packages are installed.

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

Note: If you have a valid RHN subscription, run yum as shown in the following example:

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

Configuring Authentication

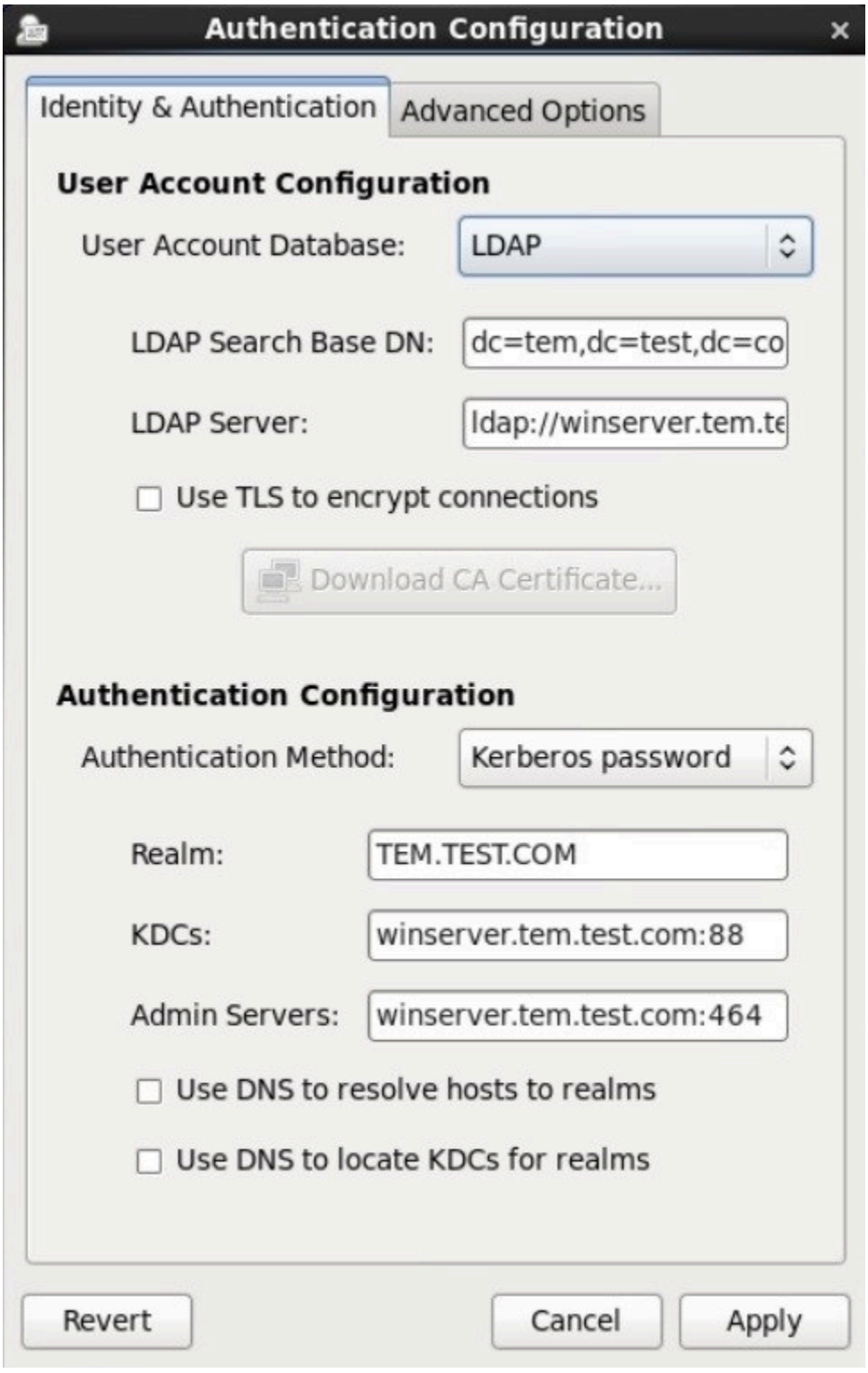
To configure the Kerberos protocol, the LDAP security and the authentication files for Active Directory integration, you can use one of the following methods.

- The **system-config-authentication** graphical tool.
- The **authconfig** command-line tool.

Using the system-config-authentication graphical tool

To configure the authentication with the system-config-authentication tool, perform the following steps.

1. Run the **system-config-authentication** graphical tool to define LDAP as the user account database for user authentication.
2. In **Identity & Authentication,** from the **User Account Database** drop-down list, select **LDAP.** Selecting the **LDAP** option allows the system to be configured to connect to the Windows Active Directory domain using LDAP with Kerberos authentication.



3. In **LDAP Search Base DN** specify to retrieve the user information using the listed Distinguished Name (DN), such as `dc=tem,dc=test,dc=com`.

4. In **LDAP Server** specify the address of the LDAP server such as `ldap://winserver.tem.test.com`
5. In **Authentication Method** select **Kerberos password**.
6. Configures the realm for the Kerberos server in **Realm**, such as `TEM.TEST.COM`. Ensure you enter the Realm name in uppercase.
7. Specify the *Key Distribution Center* (KDC) in **KDCs** for issuing Kerberos tickets, for example, `winserver.tem.test.com`
8. Specify the administration servers running `kadmind` in the **Admin Servers**, such as `winserver.tem.test.com`
9. Click **Apply**.

For more information about how to use this tool, see Launching the Authentication Configuration Tool UI.

Using the authconfig command-line tool

To update all of the configuration files and services required for system authentication, you can run the **authconfig** command-line tool.

As shown in the following example:

```
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
```

where:

--enableldap

Specifies to configure to connect the system with the Windows Active Directory domain using LDAP with Kerberos authentication.

--ldapserver

Specifies the address of the LDAP server such as `ldap://winserver.tem.test.com`

--ldapbasedn

Specifies to retrieve the user information using the listed Distinguished Name (DN), such as `dc=tem,dc=test,dc=com`

--enablekrb5

Enables the Kerberos password authentication method.

--krb5realm

Configures the realm for the Kerberos server, such as `TEM.TEST.COM`. Ensure you specify the realm name in uppercase.

--krb5kdc

Specifies the *Key Distribution Center* (KDC) for issuing Kerberos tickets, such as `winserver.tem.test.com`.

--krb5adminserver

Specifies the administration servers running `kadmind`, such as `winserver.tem.test.com`.

--update

Applies all the configuration settings.

For more information about how to use this command, see Configuring Authentication from the Command Line.

Modifying the local LDAP name

To modify the local LDAP name, perform the following steps.

1. Make a backup copy of the LDAP configuration file as follows:

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

2. Modify the value of the `base` and `uri` settings in the `/etc/nslcd.conf` file as in the following example:

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

3. Restart the local LDAP name service daemon:

```
service nslcd restart
```

4. Ensure that the local LDAP name service daemon (`nslcd`) is set to start with the server:

```
chkconfig nslcd on
```

Configuring the NSS and PAM libraries

How to use the LDAP database to authenticate users on a Linux system.

Edit the /etc/nsswitch.conf and change `passwd`, `shadow` and `group` entries from the SSSD daemon (**sss**) to LDAP:

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

to LDAP (**ldap**):

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

To configure the PAM libraries, edit the `/etc/pam.d/system-auth` and `/etc/pam.d/password-auth` files and add the `pam_krb5.so` library entries:

```
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
```

Note: Remove the entries for the SSSD libraries (`pam_sss.so`).

For additional information on RedHat integration see Integrating Red Hat Enterprise Linux 6 with Active Directory.

Configuring Kerberos authentication on Red Hat Enterprise Linux 8 or later

To ensure a secure communication between Linux BigFix server and Active Directory, use the Kerberos protocol.

To integrate the Linux BigFix server with the Windows Active Directory domain using LDAP with Kerberos authentication, perform the following steps:

1. Ensure that the host names are set correctly in both the Linux BigFix server and the Active Directory server.
2. Manually configure the `krb5.conf` file.

Preliminary checks

Before running the integration between the BigFix server running on a Red Hat Enterprise Linux 8 or later system and the Active Directory server, ensure the following.

- The Red Hat Enterprise Linux is at release 8.9 or higher.
- The DNS host names of both the Red Hat Enterprise Linux and the Active Directory server are resolved correctly.

Manually configure krb5.conf file

Edit the `/etc/krb5.conf` file and add the following lines, replacing "TEM.TEST.COM", "tem.test.com" and “myHostName.tem.test.com" with the domain and hostname in your environment (note that the case must be respected).

Remember to specify the ports for "kdc" and "admin_server" values in case you are not using the default values (88 and 749 respectively):

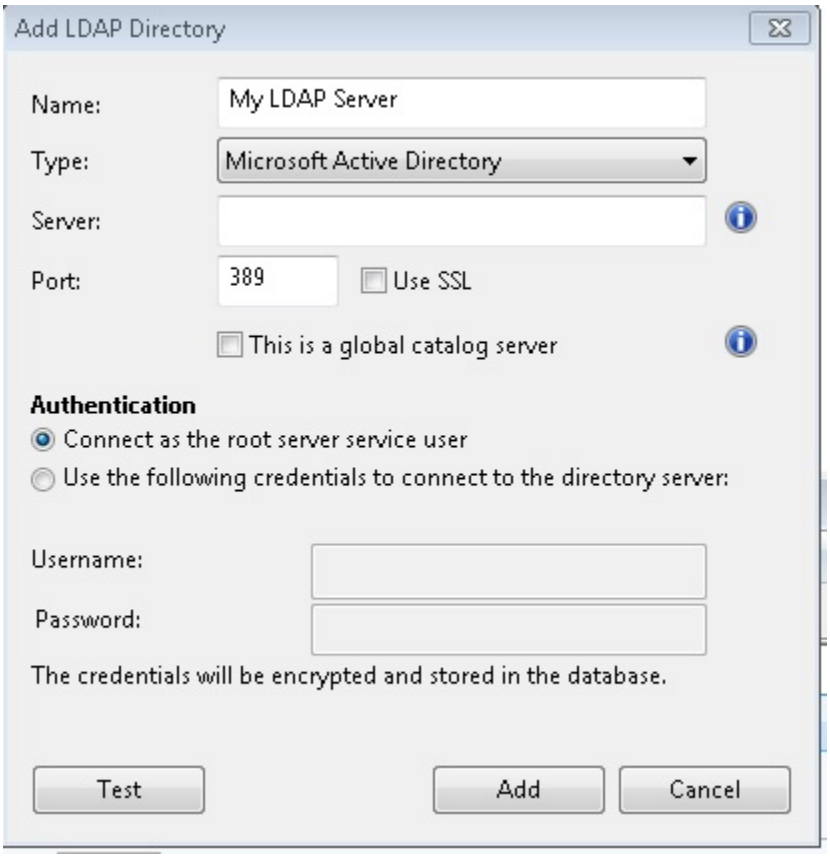
```
default_realm = TEM.TEST.COM
[realms]
 TEM.TEST.COM = {
  kdc = myHostName.tem.test.com
  admin_server = myHostName.tem.test.com
 }

[domain_realm]
 tem.test.com = TEM.TEST.COM
 .tem.test.com = TEM.TEST.COM
```

Integrating the server with Active Directory

How to integrate the BigFix server with Active Directory.

1. From the **Tool** menu, select **Add LDAP Directory**. The **Add LDAP Directory** dialog displays.



2. Provide a name for the Active Directory and from the Type pull-down, make sure **Microsoft Active Directory** is selected.
3. Under **Server**, enter the host name, IP Address or fully qualified domain name of the server.
4. Click **Use SSL** if you want to configure a secure connection (SSL).
5. To access an entire Active Directory forest, click **This is a global catalog server**.
6. Click the button to **connect as the root server service user** or to **use credentials**. If you choose to connect using credentials, enter your Active Directory **Username** and **Password**.
7. Click **Test** to make sure you have entered your information correctly and a connection can be made to your Active Directory server.
8. Click **Add** to complete the Active Directory setup.

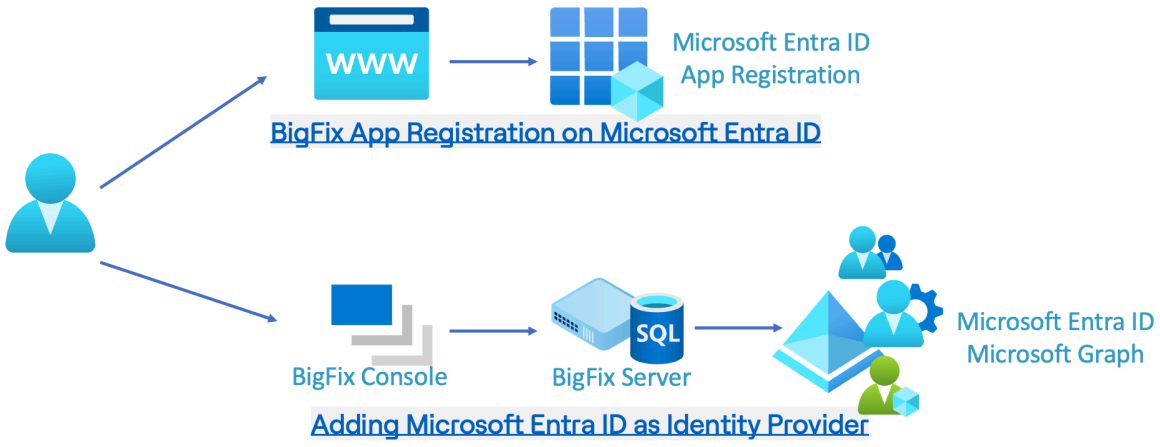
Note: When you add an LDAP Server as **Microsoft Active Directory**, ensure that on the LDAP server you have defined the `UserPrincipalName` attribute corresponding to the **User logon name** of each user. This attribute value is used on the BigFix Console for each user authentication.

Integrating with Microsoft Entra ID

You can use Microsoft Entra to handle authentication on BigFix.

With the release of BigFix Platform Version 11 Patch 2, you can now integrate Microsoft Entra ID as a new identity provider directly from the BigFix Console.

This integration allows for efficient authentication management using Microsoft Entra ID credentials, providing users with secure and convenient access to BigFix resources. With this integration, administrators can streamline access management processes, enhance overall security, and offer greater flexibility to users in how they access and utilize the BigFix Platform.



The configuration needed to integrate Microsoft Entra ID as a new identity provider is a two step process. First, you must configure the BigFix App from the Microsoft Entra Gallery as described in BigFix App Registration on Microsoft Entra ID (on page 76) and, then, perform the Identity Provider configuration in the BigFix Platform as described in Adding Microsoft Entra ID as Identity Provider (on page 81).

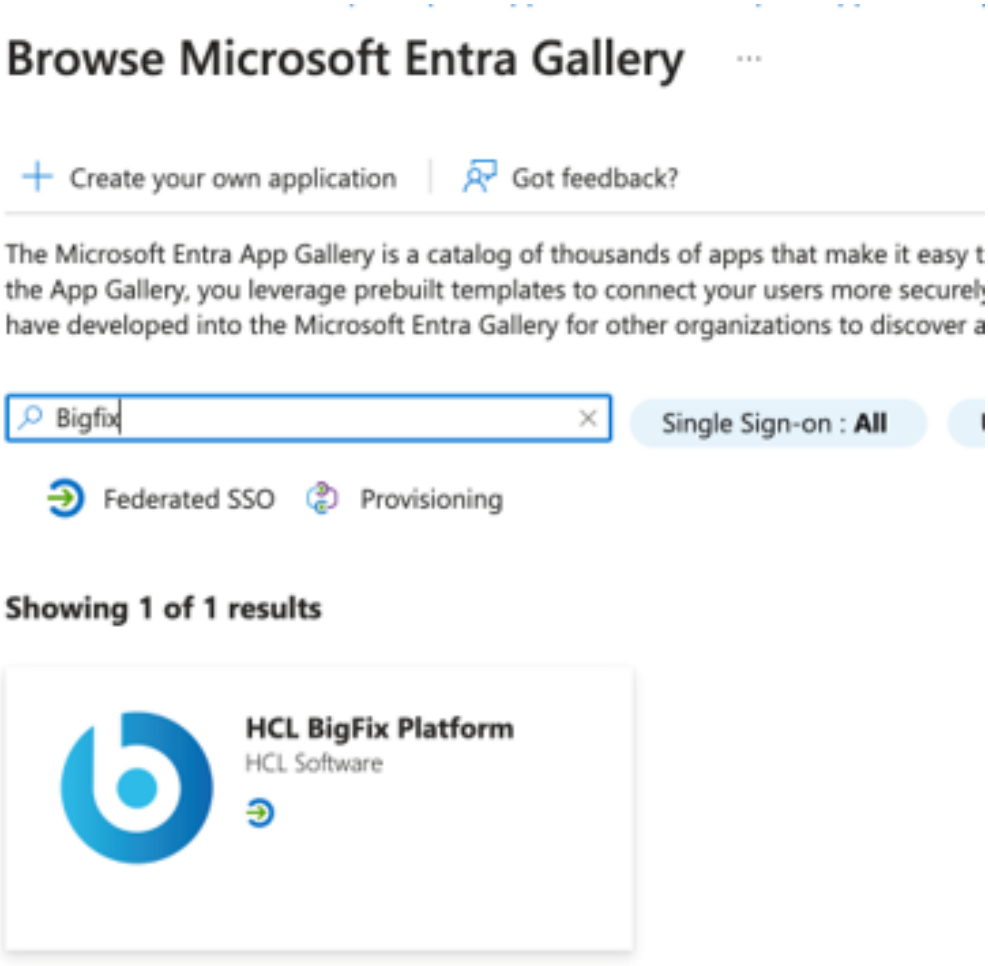
Note: If your Microsoft Entra ID provider requires multi-factor authentication (MFA), then you must enable SAML on the BigFix App, as described in Enabling SAML V2.0 authentication for identity provider operators *(on page 96)*.

BigFix App Registration on Microsoft Entra ID

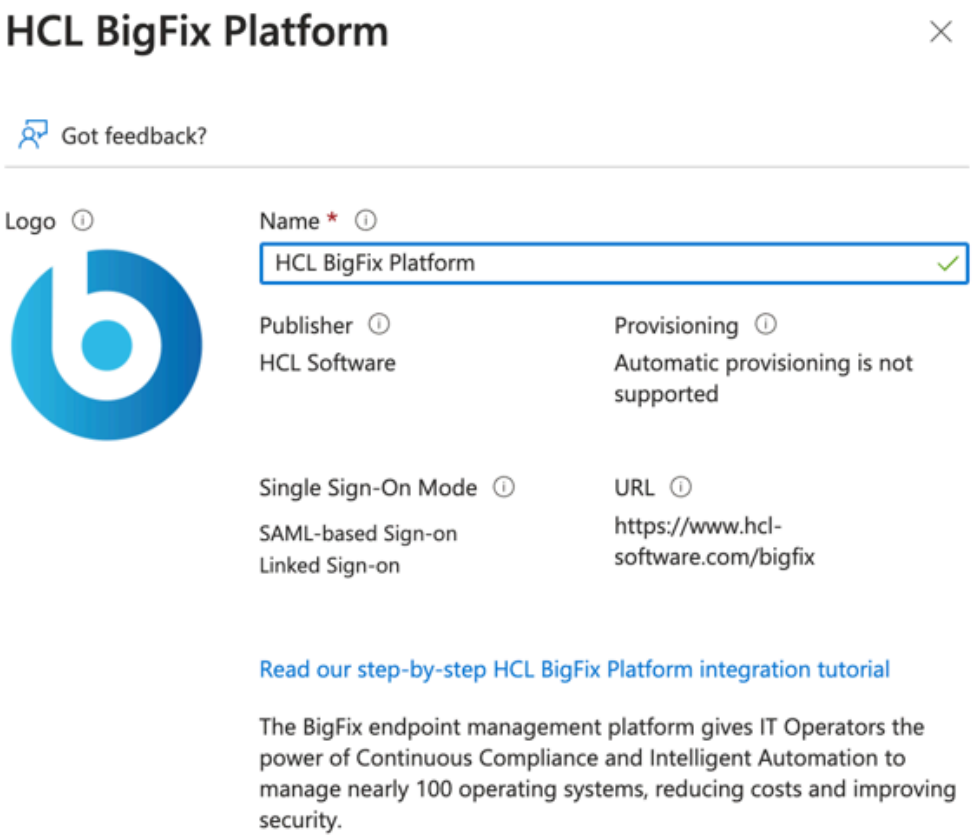
You can create your BigFix App from the Microsoft Entra Gallery.

To create your BigFix App on Microsoft Entra Gallery:

1. Select Microsoft Entra ID > New Enterprise Application.
2. Type “Bigfix” in the search area and click the BigFix tile in the Microsoft Entra Gallery.



3. Insert an application name and click Create.

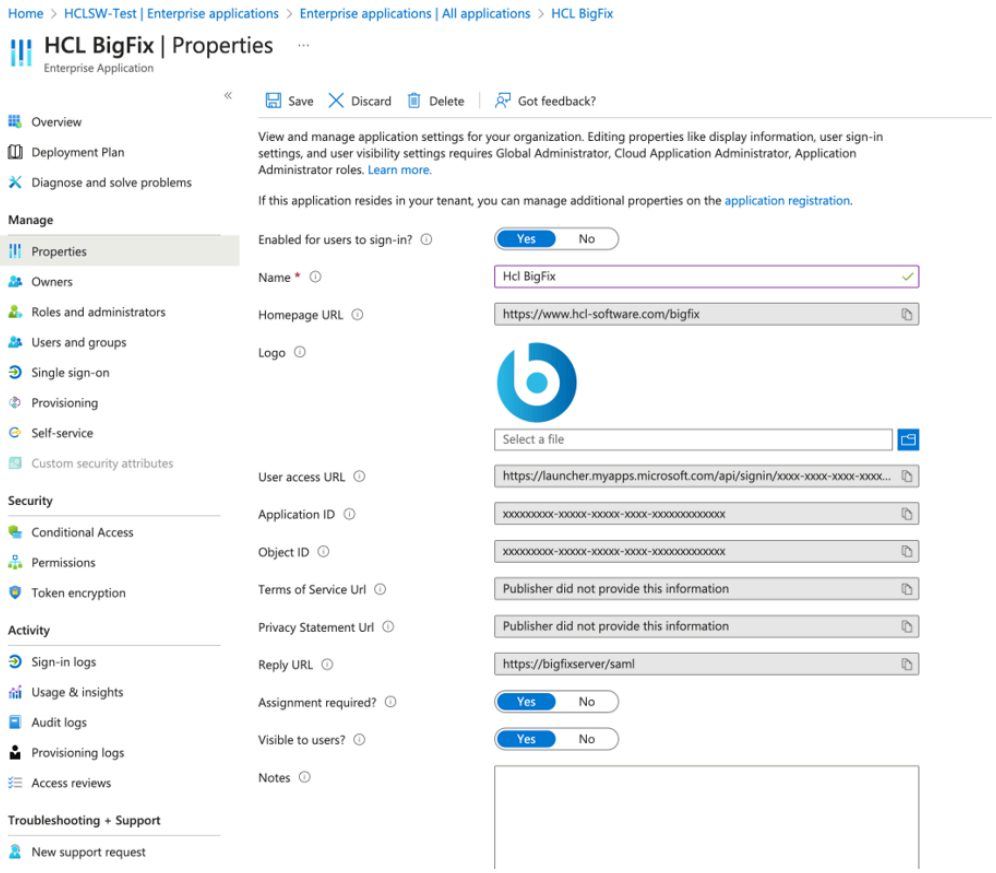


BigFix requires the following API permissions for Microsoft Graph related to user and group membership to retrieve the necessary information and enable Entra ID users to log in and access BigFix applications with the appropriate privileges as shown in the image below:

API / Permissions name	Type	Description	Admin consent requ...	Status
Microsoft Graph (3)				
GroupMember.Read.All	Application	Read all group memberships	Yes	Granted
openid	Delegated	Sign users in	No	Granted
User.ReadBasic.All	Application	Read all users' basic profiles	Yes	Granted

Note: To ensure only authorized users can log in to BigFix, set the **Assignment required?** option to **Yes**. This option is located under the **Properties** tab for the registered application, accessible by navigating to **Enterprise Applications** and selecting the app. This configuration restricts login access to users assigned to

the application or those belonging to a group assigned to it, provided they have the necessary permissions.



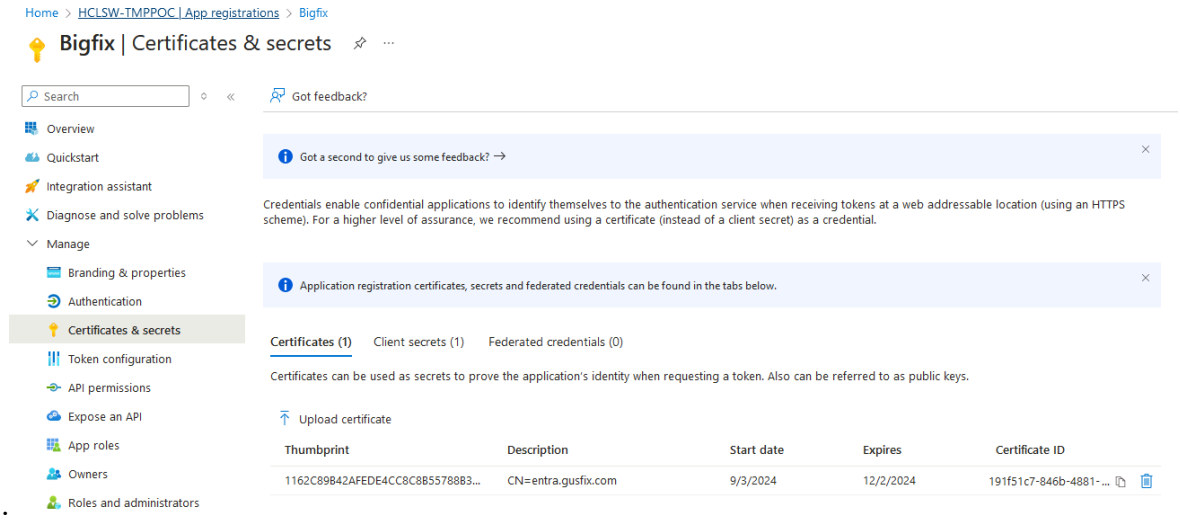
Certificate Configuration

Starting from BigFix Version 11.0.3, if you want to access the Microsoft Entra ID application using a certificate instead of a client secret, follow the steps below:

1. **Create a private key and public certificate pair** in PEM format. The certificate must comply with the X.509 standard and be based on a 2048-bit RSA key pair. The private key may or may not be password-protected.
   The output of this step should be two .pem files:
   a. The first file should contain the private key and a public certificate signed by that key.
   b. The second file should contain only the public certificate.

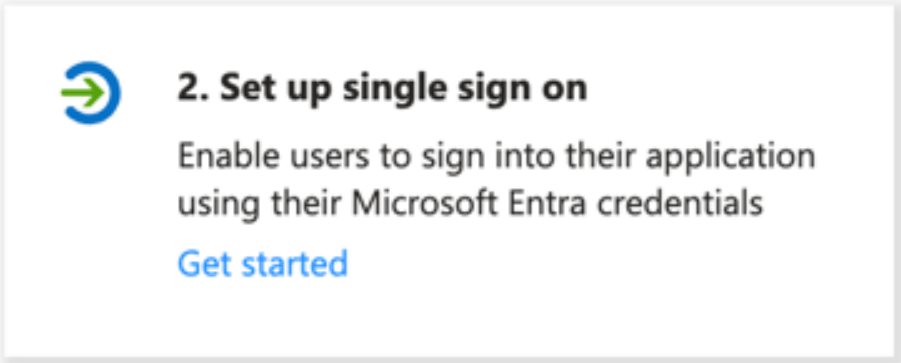
Note: It is possible to use a self-signed certificate, but it's recommended to use a certificate issued by a trusted Certificate Authority (CA).

2. Select **App registrations** and click on the application name to access its settings.
3. Click **Certificates & Secrets** and select the **Certificates** tab.
4. Click **Upload certificate** and browse to select your public certificate file.
5. Click **Add** to upload the certificate.
6. Ensure that the certificate appears on the list of the uploaded certificates, with its thumbprint and expiration date displayed.

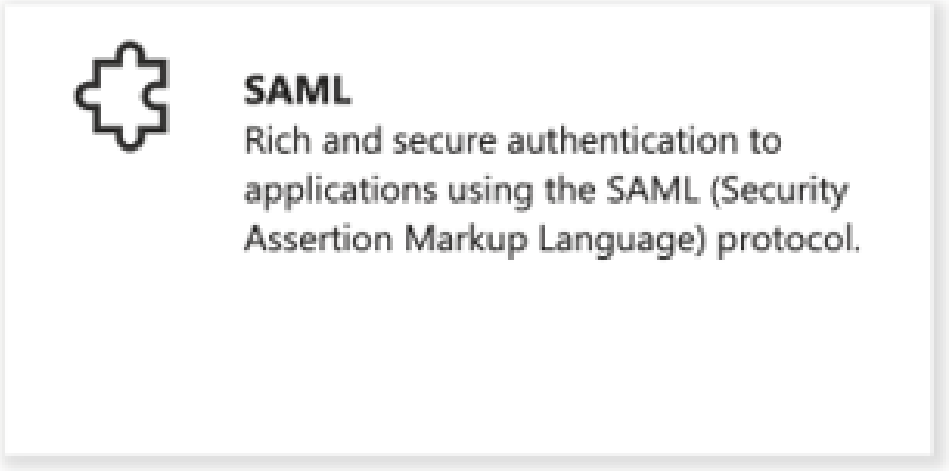


SAML configuration

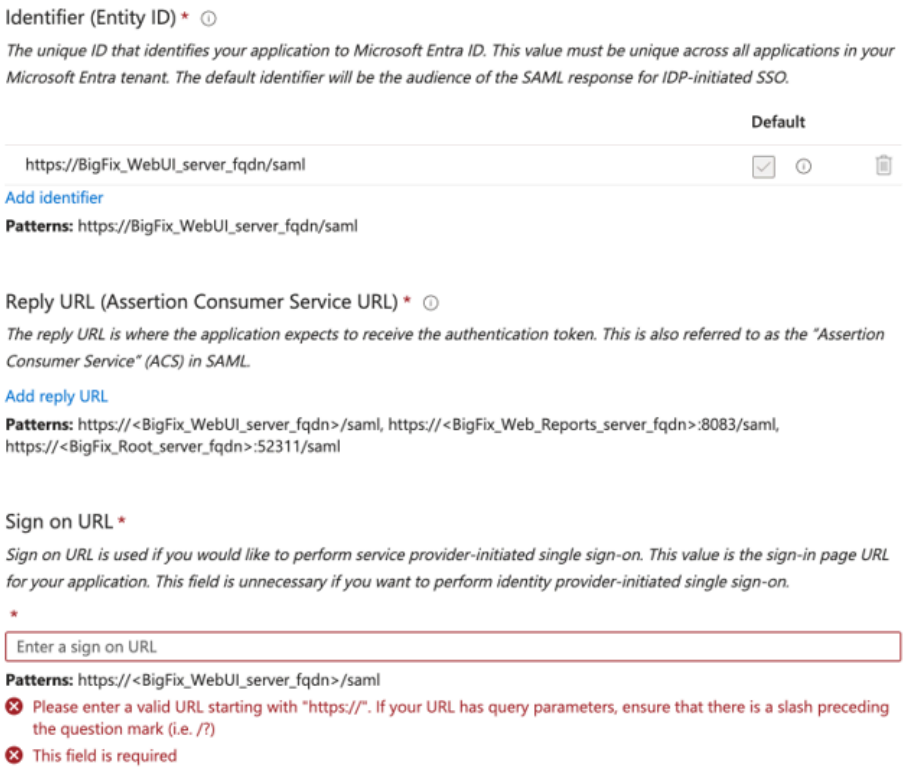
1. In the overview page of the created Enterprise Application, click **Set up single sign on**.



2. Click **SAML**.



3. Edit the **Set up Single Sign-On with SAML** section and fill the required information according to the patterns example provided under each field.

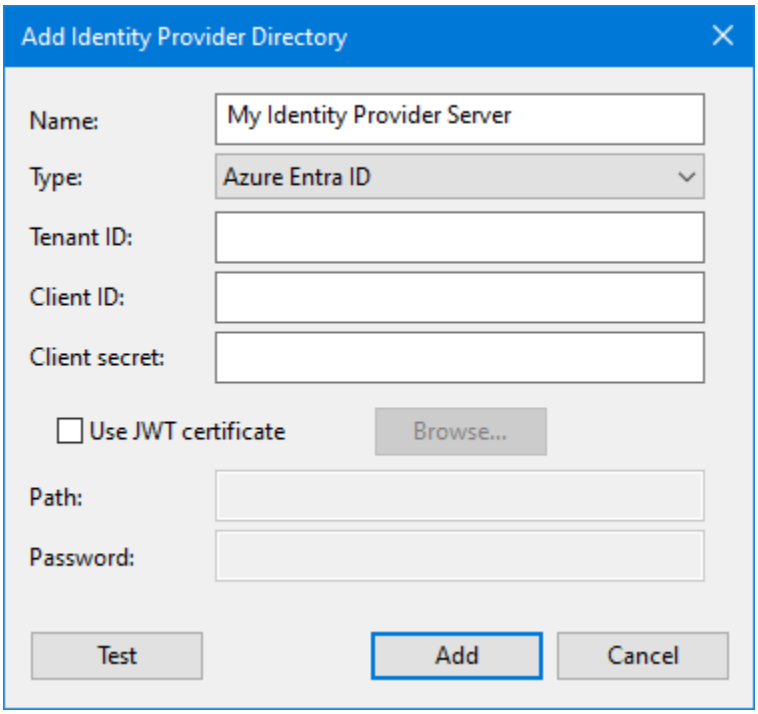


Adding Microsoft Entra ID as Identity Provider

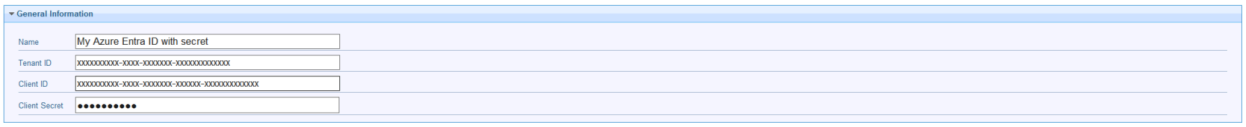
You can add Microsoft Entra ID as Identity Provider as follows.

In the new dialog, the following information are required to set up Microsoft Entra ID:

- **Tenant ID**: A tenant ID is a unique identifier for a Microsoft Entra ID tenant. It is used to identify the tenant when communicating with Microsoft services.
- **Client ID**: A client ID is a unique identifier for an application that has been registered with Microsoft Entra ID. It is used to identify the application when communicating with Microsoft Entra ID.
- **Client secret**: A client secret is a secret key that is associated with a client ID. It is used to prove the identity of the application when communicating with Microsoft Entra ID.

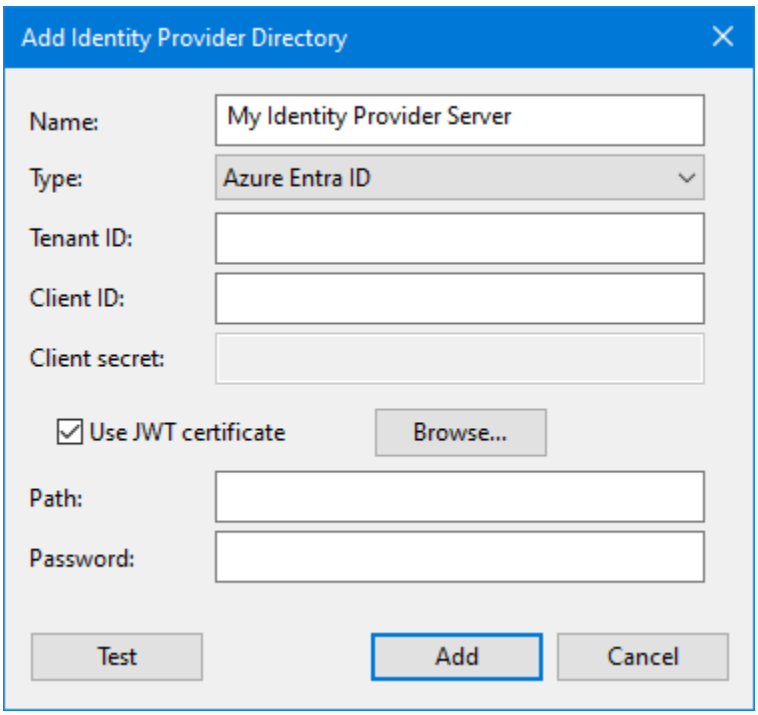


The description page with general information about the newly added identity provider now displays the new details, as shown below:

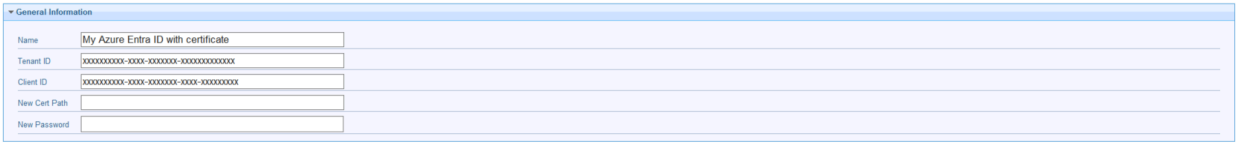


If you want to use a certificate instead of the client secret, select the **Use JWT certificate** check box and insert the following information:

- **Path**: The path to the file containing both the private key and the public certificate. They must be included in a single file with a `.pem` extension.
- **Password**: The password for the private key if it is password-protected. Leave the text field empty if no password is set.



The description page with general information about the newly added identity provider now displays the new details, as shown below:



Note: The certificate path and private key password are not shown in the general description. Instead, the "New Cert Path" and "New Password" text fields are displayed to allow you to update the certificate.

After adding the new identity provider from the Console, you can now add Azure users to create BigFix operators. You can then also associate Azure groups to BigFix roles.

Scenario 1: Creating a specific BigFix Operator from an Azure User

Prerequisites to satisfy before running this scenario:

- The application has been registered in Microsoft Entra ID.
- The identity provider has been successfully added to BigFix.

Perform these steps:

1. Add the user to the application in the Azure portal. You can either add the user individually or add a group that the user belongs to.
2. Create the BigFix operator in the Console as follows:
    a. Go to the "Operators" dialog.
    b. Right-click and select "Add Identity Provider Operator".
    c. Locate and select the user from the previously added tenant.
3. Assign the desired permissions to the new operator. The master operator can assign them.
4. The new operator account is now ready to be used.

Scenario 2: Creating BigFix Operators from an Azure Group

Prerequisites to satisfy before running this scenario:

- The application has been registered in Microsoft Entra ID.
- The identity provider has been successfully added to BigFix.

Perform these steps:

1. Add the group to the application in the Azure portal.
2. Assign the identity provider group to a BigFix role as follows:
    a. Go to the "Role" dialog.
    b. Create or select a role for which you want to assign permissions to the Azure group.
    c. Click the "Identity Provider Groups" section within the role configuration.
    d. Select the Azure group you added in step 1 from the available list.
    e. Save the changes to the role configuration.

Note:

- Initially, no operator account is created.
- When an Azure user from the added group logs in for the first time, a BigFix operator account associated with that user will be automatically created. This account will inherit the permissions assigned to the role in step 2.

Adding Identity Provider Operators

You can create accounts for operators to access the console by using an existing Active Directory or LDAP account or Microsoft Entra ID.

When you select this option, an operator with the same name as the one specified in the Identity Provider directory, is added to the operators node in the Domain Panel on the BigFix console. These operators can then log in as usual, using one of the following notations:

username
username@domain
domain\username

The permissions assigned to that user in the Identity Provider directory are not inherited by the newly created operator. You must either assign the needed permissions to the operator or assign the operator to an existing role.

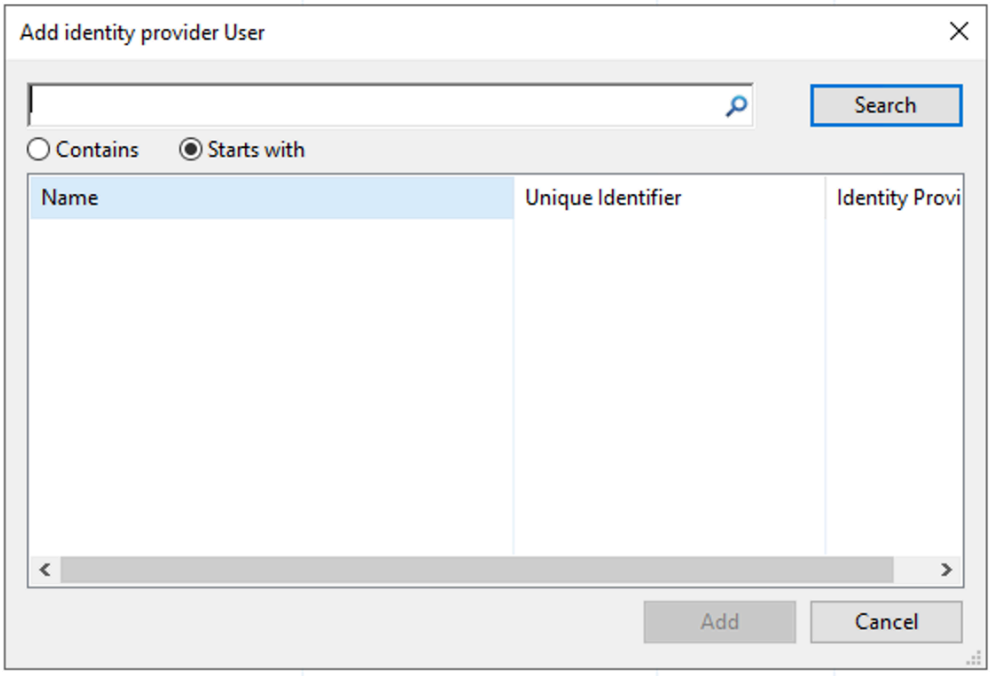
Note: You can integrate BigFix with SAML V2.0 to provide BigFix Identity Provider operators with:

- Two-factor authentication with Common Access Cards (CAC), Personal Identity Verification (PIV) cards, or other factors, if required by the Identity Provider.
- Web-based Single Sign-On authentication method from the identity provider login URL.

For more information, see Enabling SAML V2.0 authentication for identity provider operators *(on page 96)*.

To add an Identity Provider user, complete the following steps:

1. Ensure that the needed Active Directory or LDAP directory or Microsoft Entra directory is added to the BigFix environment.
2. Click the **Tools > Add Identity Provider Operator** menu item or right click in the work area and then select **Add Identity Provider Operator**. The following dialog appears.



3. You can query and filter the users defined on the specified Identity Provider server using the Search field and the two radio buttons.

   Note: If you have configured at least one Microsoft Identity Provider in your environment and have a significant number of users in your Tenant, it is highly recommended to use the "Starts With" option, as it can be significantly faster than the "Contains" option.

   A new client property, named _Enterprise Server_MSEntraID_PageSize , has been introduced to let BigFix users configure the Microsoft Entra page size for REST API responses. This property must be set on the machine where the BigFix Server is running. By default, it is set to 0, indicating that no paging option is applied, and the Microsoft Entra default value of 100 items is returned for each request. Increasing the page size can enhance performance when dealing with a large number of Microsoft Entra users.

Name	**_Enterprise Server_MSEntraID_Page-Size**
Default Value	0
Setting Type	int
Value Range	0 - 999
Component Restart Required	No

4. When you find the user to add as Identity Provider operator, select it and click **Add**. The Console Operator panel opens.



5. From the **Details** tab assign operator permissions.

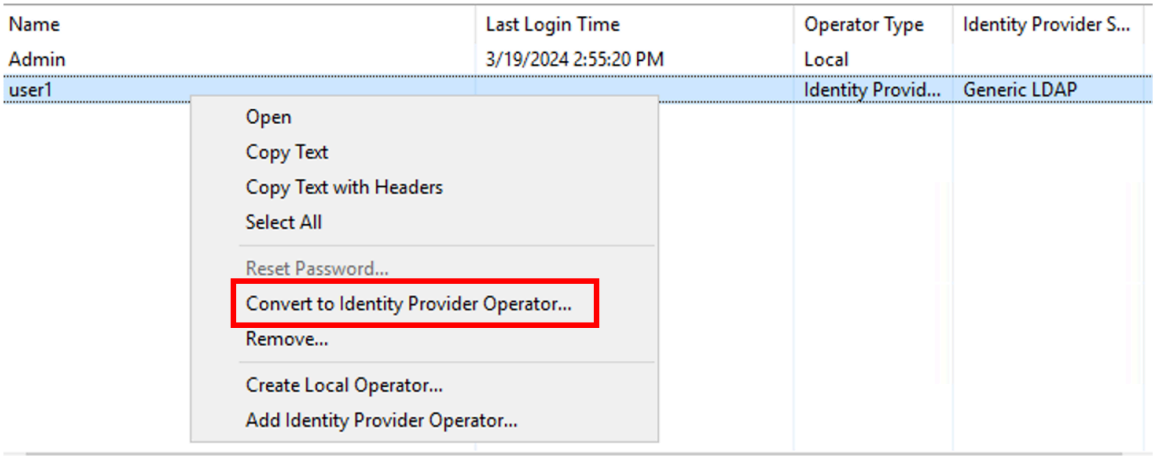
   You can decide to give the operator the ability to trigger restart and shutdown as Post-Action or to include them in BigFix Action Scripts. Depending on the configuration that you set for a specific operator for shutdown and restart, the radio button in the Post Action tab of the Take Action panel might be disabled for that operator. This configuration has no effect on actions with action script type other than BigFix Action Script.

   You can also set permissions to access the BigFix Console and REST API.
6. The **Administered Computers** tab lists the computers managed by this operator.
7. From the **Assigned Role** tab, select the roles that you want to assign or unassign this operator to.
8. From the **Sites** tab, assign the sites that you want this operator to have access to or unassign them.

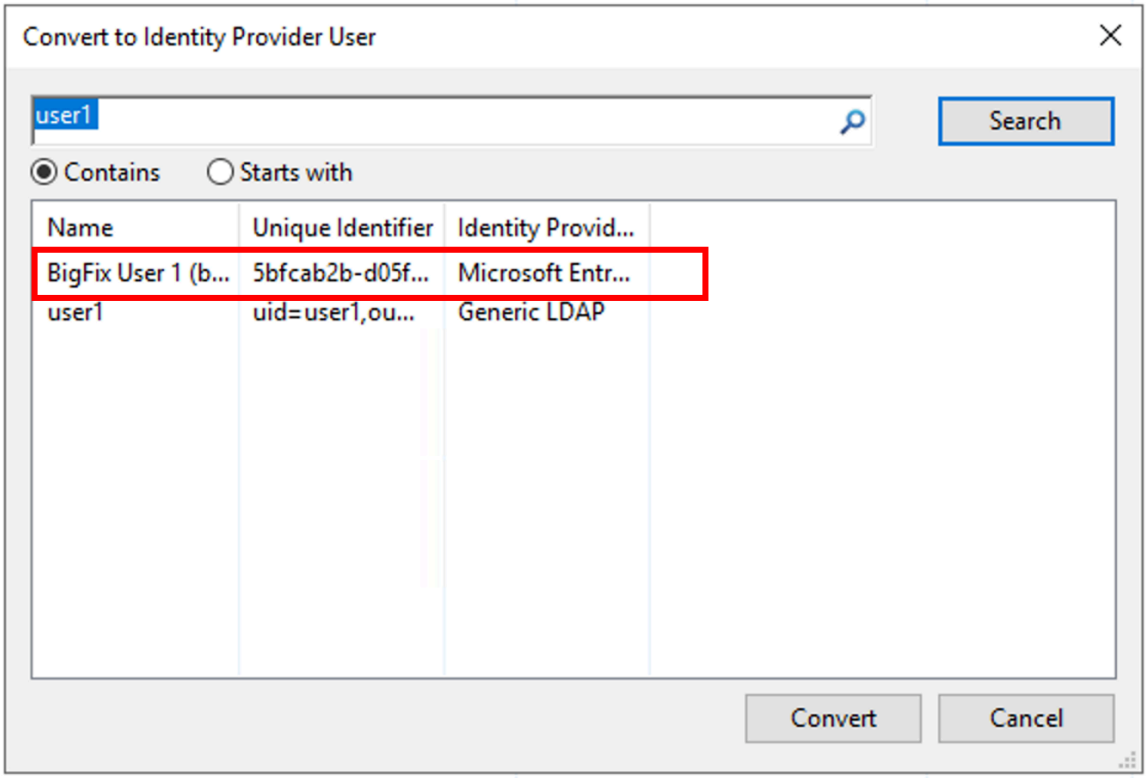
9. From the **Computer Assignments** tab, specify the properties that must be matched by the computers that the operator can manage.
10. To save the changes click **Save Changes**.

To convert an existing local or old LDAP operator to an Identity Provider operator, complete the following steps:

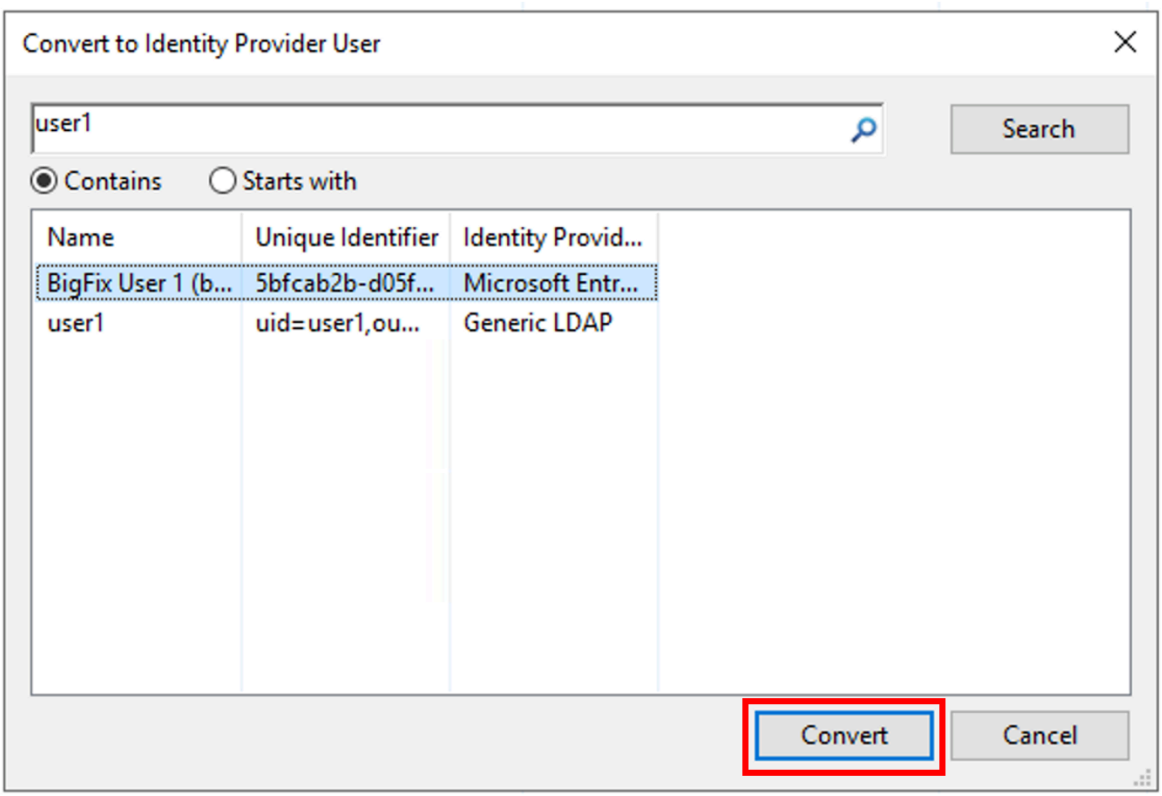
1. From any list of local operators, right-click the operator you want to convert.
2. From the context menu, select **Convert to Identity Provider Operator**.



3. Locate the Identity Provider user you want to convert the operator to.



4. Select the user name and click **Convert**.



Permissions Inheritance: The converted operator inherits all permissions assigned to the chosen Identity Provider user. However, there is an exception: permissions granted to the old operator based on their group membership are not inherited.

Example

This is an example of converting a user displayed in the screen captures above.

A user named "user1" belongs to a group "TestGroup" associated with the "TestRole" role granting permission "permission1".

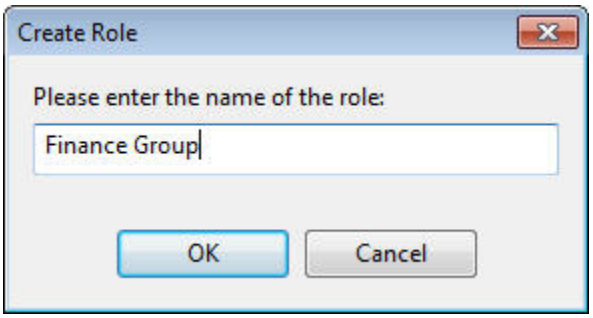
To ensure that "BigFix User 1" (converted from "user1") has "permission1", "BigFix User 1" must belong to a new group "newTestGroup" associated with "TestRole."

Associating an Identity Provider Group

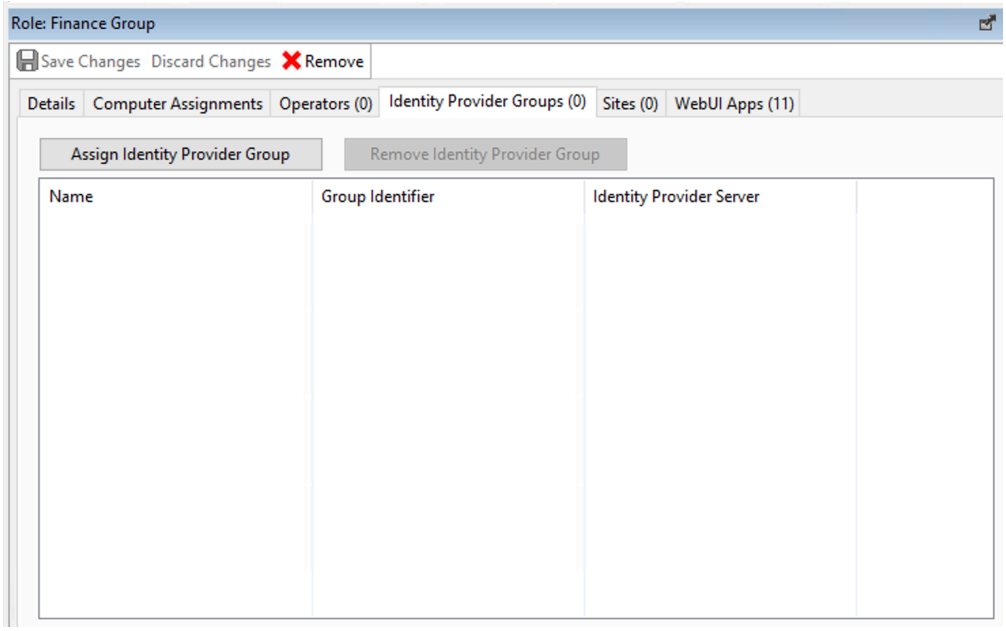
You can associate identity provider users or groups, that have been defined in an existing Active Directory or LDAP directory or Microsoft Entra directory, to console operators or roles.

To add such a group, perform the following steps:

1. Ensure that the needed Active Directory or LDAP directory or Microsoft Entra directory is added to the BigFix environment.
2. Create a role to accept your new group by selecting **Tools > Create Role** or right click in the work area and then select **Create Role**.

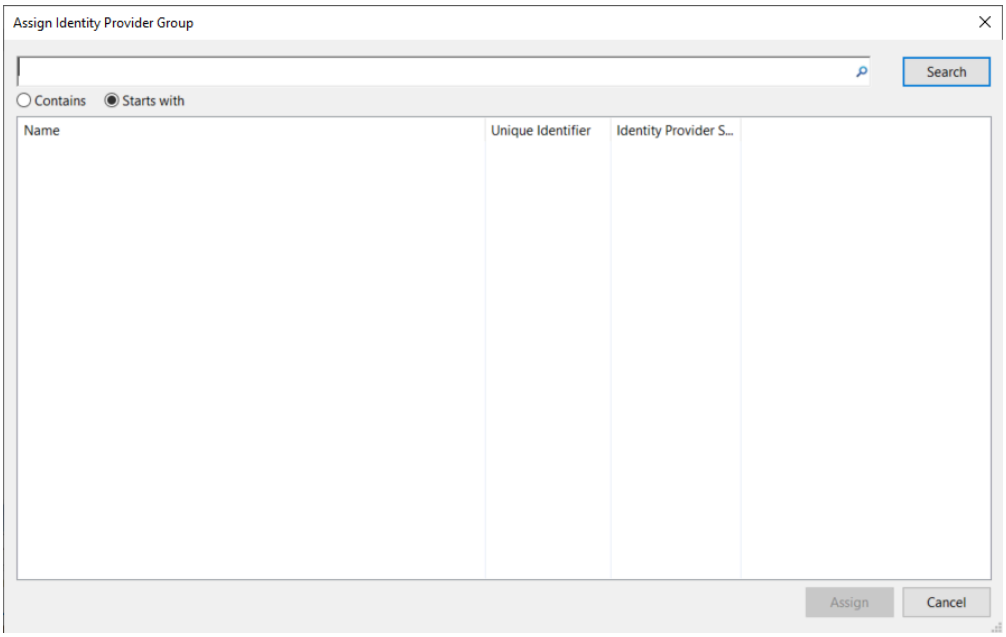


   Enter a name for your group and click **OK**.
3. The **Role** panel appears.



Click the **Identity Provider Groups** tab.

4. Click the **Assign Identity Provider Group** button. The following dialog displays.



5. You can query and filter the groups defined on the specified Identity Provider server using the Search field and the two radio buttons.

Note: If you have configured at least one Microsoft Identity Provider in your environment and have a significant number of groups in your Tenant, it is highly recommended to use the "Starts With" option, as it can be significantly faster than the "Contains" option.

A new client property, named _Enterprise Server_MSEntraID_PageSize , has been introduced to let BigFix users configure the Microsoft Entra page size for REST API responses. This property must be set on the machine where the BigFix Server is running. By default, it is set to 0, indicating that no paging option is applied, and the Microsoft Entra default value of 100 items is returned for each request. Increasing the page size can enhance performance when dealing with a large number of Microsoft Entra groups.

Name	**_Enterprise Server_MSEntraID_Page-Size**
Default Value	0
Setting Type	int
Value Range	0 - 999
Component Restart Required	No

6. Select the identity provider group that you want to assign to this role and click **Assign Identity Provider Group**.
7. To save the changes click **Save Changes**.

When you assign an identity provider group to a role, any user from that group can then log in to the console. Only those users who actually log in will be provisioned with accounts and thus end up in the list of operators. This avoids the creation of unnecessary accounts. Operators are granted the highest privileges resulting from the sum of all their roles and permissions. For instance, if a user has access to computer set A and sites X from role 1, and computer set B and sites Y from role 2, they will have permissions for Sites X and Y across both computer sets A and B.

User-based management

For domain-joined computers, starting with BigFix Platform version 11.0.7, BigFix is able to periodically retrieve computer data from Microsoft Entra ID and Active Directory, such as the User associated with the computer and the User Groups to which that User belongs, as well as the Computer Groups to which the computer itself belongs, and more.

This data is made available as computer properties, allowing BigFix operators to achieve user-centric endpoint management, advanced dynamic targeting, and reporting.

Reserved Properties

To enable out-of-the-box dynamic targeting and global reporting within the BigFix Console and WebUI, the framework implements a set of core Server Reserved Properties. These properties abstract provider-specific details (Active Directory vs. Microsoft Entra ID) by evaluating native relevance expressions that automatically query and map the correct identity attributes.

When an endpoint cannot be correlated with a directory identity, or if specific attributes are unpopulated, these properties gracefully fall back to returning "N/A".

The following table details how each Reserved Property handles identity resolution across hybrid directory types:

Property Name	Source Object	Retrieved identity attribute	Default Fall-back
IDP User Principal Name	Managing User	Returns the user principal name associated with the computer.	"N/A"
IDP User Display Name	Managing User	Returns the user display name associated with the computer.	"N/A"
IDP Computer Groups	Computer Object	Returns the list of groups to which the computer belongs.	"N/A"
IDP User Groups	Managing User	Returns the list of groups to which the user associated with the computer belongs.	"N/A"
IDP User Department	Managing User	Returns the department to which the user associated with the computer belongs.	"N/A"
IDP User Location	Managing User	Returns the office location of the user associated with the computer.	"N/A"

IDP User Site	Managing User	Returns the aggregated data about the worksite address of the user associated with the computer.	"N/A"

Architectural Overview

The identity integration relies on an automated, asynchronous dual-component flow (Server-Side and Client-Side) to map users and groups to computers.

Identity Aggregation Service

A scheduled task on the BigFix Server maintains a cache of directory objects from all configured Identity Providers. This ensures fast lookups and high performance without repeatedly querying external directory systems. At administrator-defined intervals, the server performs a data refresh. It iterates through all configured IdPs, querying cloud sources via the **Microsoft Graph API** and on-premises sources using standard **LDAP queries**. After each polling cycle, for each domain-joined computer, relevant updated directory data is packaged and delivered to the endpoint **only if changes are detected** (e.g., changes in group membership or office location), reducing network overhead.

Identity Discovery

The BigFix Client automatically discovers the local system's directory state during initialization. In hybrid scenarios where an endpoint is simultaneously joined to a traditional Active Directory domain and Microsoft Entra ID, the client **always gives precedence to Entra ID (Azure AD) information** to align with modern cloud-first standards. The discovered identity state is transmitted to the BigFix Server during every client registration request.

Chapter 4. Enabling SAML V2.0 authentication for identity provider operators

BigFix supports SAML V2.0 authentication via LDAP-backed SAML identity providers.

After configuration, SAML V2.0 support enables:

- Two-factor authentication for BigFix with Common Access Cards (CAC), Personal Identity Verification (PIV) cards, or other factors, if required by the Identity Provider.
- Web-based Single Sign-On authentication method from the identity provider login URL. Logged in users are automatically redirected, upon request, to the web-based components that support SAML V2.0 authentication without having to log in again.

What Is SAML 2.0

The OASIS Security Assertion Markup Language (SAML) is a standard that uses an XML-based framework to describe and exchange security information between online entities.

For more details about SAML terminology, see SAML Key terms.

SAML 2.0 supports:

Web-Based Single Sign-On

It provides a standard vendor-independent grammar and protocol for transferring information about a user from one web server to another, independent of the server DNS domains.

Identity federation

It allows partner services to agree on and establish a common name identifier for the user to share information about themselves across organizational boundaries.

This type of sharing helps to reduce identity management costs.

Federated identity implements FIPS 201 to define a US Government-wide interoperable identification credential, known as the Personal Identity

Verification (PIV), for controlling physical access to federal facilities and logical access to federal information systems.

The CAC PIV card is a multi-application smart card for PIV Cardholder authentication that contains a linear barcode, two-dimensional barcode, magnetic stripe, color digital photograph, and printed text. It serves as a token for:

- Logical access to computer systems
- Personnel identification
- Physical access to buildings
- Public-Key Infrastructure (PKI) for signing, encryption, and non-repudiation.

Web services and other industry standards

SAML allows its security assertion format to be used outside a "native" SAML-based protocol context. This modularity has proved useful to other industry efforts addressing authorization services (IETF, OASIS), identity frameworks, web services (OASIS, Liberty Alliance), and so on.

How SAML works

The SAML specification defines three parties.

They are the following:

- The principal, which is typically a user.
- The Identity provider (IdP), which is the LDAP-backed SAML identity provider.
- The service provider (SP), which in this case are the BigFix services.

The SAML standard controls how the identity assertions are exchanged among these three parties. SAML does not specify the method of authentication at the identity provider.

In SAML, one identity provider can provide SAML assertions to many service providers.

For more information about SAML V2.0 use case scenarios, see SAML V2.0 Overview.

Which BigFix user interfaces integrate with SAML V2.0

The SAML authentication enhancement, when configured, affects all BigFix LDAP or Microsoft Entra ID managed users accessing the Web UI, Web Reports and, starting from BigFix Version 9.5.5, the BigFix console.

How BigFix integrates with SAML V2.0

The integration with SAML V2.0 uses the passport-saml authentication provider to allow both Identity provider (IdP) initiated and Service provider (SP) initiated authentication.

The SAML use and requests are managed, for all the BigFix user interfaces that support it, by a WebUI component.

The way you configure the integration with SAML depends on the use that you plan to do:

- If you want to use the SAML authentication for Web Reports and for theBigFix console only, and you do not need to use it with any WebUI application, you can start the WebUI in SAML-only mode. This SAML configuration allows you to minimize resource consumption. For more information about how to set up this configuration, see SAML 2.0.
- If you want to use the SAML authentication for all the BigFix user interfaces, including the full set of WebUI components, or the WebUI ETL process, follow the instructions provided in WebUI Installation.

If the BigFix environment uses one LDAP server as a user repository, user provisioning is not affected by this integration, and administrators continue to define operators and roles to authorize them to use BigFix services. If your BigFix environment operators are defined on more than one LDAP server, read carefully the information provided in Assumptions and requirements (on page 99).

Integration with SAML 2.0 maintains existing audit scenarios and includes SAML-authenticated user entries in the `server_audit.log` file.

See the following sample use case:

1. The user requests a service from BigFix, for example, accesses a page or attempts to log in, through the Web UI, the Web Reports or the BigFix console.
2. BigFix requests an identity assertion from the LDAP-backed SAML identity provider.
3. Before delivering the identity assertion, the LDAP-backed SAML identity provider might request some user authentication information, such as user name and password, or another form of authentication, including multi-factor authentication. A directory service such as LDAP or Active Directory is a typical source of authentication token at an identity provider.
4. On the basis of the identity assertion provided by the identity provider, BigFix decides whether to perform the service requested by that user.
5. The authentication information is retained and used to allow automatic access for the user, according to the assigned permissions, to the services provided by BigFix.

Assumptions and requirements

Before configuring BigFix to use SAML V2.0, carefully read the following list of assumptions and requirements.

- BigFix supports SAML V2.0 authentication with an SAML V2.0-compliant identity provider such as Active Directory Federation Services (ADFS).
- The SAML V2.0 authentication is restricted to:
  - Only one SAML IdP backed by one or more LDAP directories. If you already defined multiple LDAP servers as user repositories in your BigFix environment, be aware that, after enabling SAML authentication, only the users and the groups managed by the selected IdP will still be known to the BigFix environment. In this case, ensure that your IdP environment is correctly configured so that the SAML IdP (ADFS or ISAM) can authenticate users from the different LDAP environments that you want to use as the user repository.
  - Identity providers using SHA-256 as secure hash algorithm.
  - Web Reports servers connecting to only one data source (Root server) and configured with SSL.
- To configure and use SAML authentication, you must have the WebUI installed. If you are using the WebUI solely for providing SAML authentication for Web Reports and

the BigFix console, you can start the WebUI in SAML-only mode to reduce resource consumption. For information about how to start the WebUI in SAML-only mode, see SAML 2.0.

- In DSA architecture, the configuration is replicated to replica DSA servers. However, the replica does not enable WebUI for SAML on non-primary DSA, because multiple WebUI configuration is not supported.
- Starting from BigFix Version 10 Patch 1, the X.509 certificate used as server signing key is generated with a *subjectAltName* field containing DNS name and IPs of the Root server. This prevents the `The name on the security certificate is invalid or does not match the name of the site` security warning from appearing during the authentication process.
    - For fresh installations, a new certificate is created during the process.
    - For upgrades, the old certificate is left in place. To prevent the security warning, do the following steps:
        - Rotate the server signing key for the server to which you are connecting. In DSA architecture, you do not need to rotate keys for all the servers.

          Important: This operation resigns all the existing content. In very large deployments, it can take up to some hours. To minimize the impact on the day to day deployment operations, plan a maintenance window.

        - Apply, in alternative, the workaround described in What changes from the BigFix user's perspective *(on page 101)*.
- When running Web Reports, if SAML is enabled, the check on the referrer is not performed. You can use the setting `_HTTPServer_Referrer_CheckEnabled` to enable or disable the referrer check. The referrer is an optional header of the HTTP protocol. It identifies the address of the web page (that is the URI or IRI) that linked to the resource being requested. For information about how BigFix manages the referrer check, see List of settings and detailed descriptions *(on page 404)*.

- Starting from BigFix Version 11, if TLS 1.3 is required, the Console SAML authentication works only on a Windows version that supports the TLS 1.3. See Windows TLS protocol version support matrix to verify that the Windows version you are using supports TLS 1.3.

What changes from the BigFix user's perspective

From the BigFix user interfaces operator's perspective, this enhancement affects only authentication.

After enabling SAML authentication for LDAP users:

LDAP or Microsoft Entra ID operators:

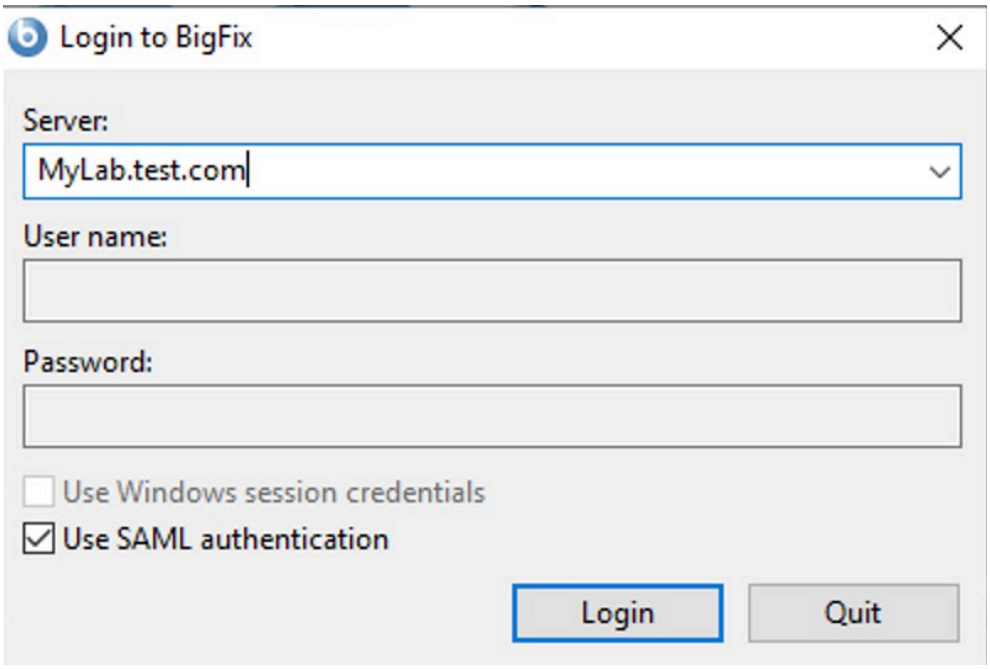
- Must authenticate to the Web UI and to the Web Reports from the SAML identity provider only by accessing the following URLs:

  `https://<WebUI_server>` (for the Web UI server, assuming that it uses port 443)

  `https://<Web_Reports_server>:8083` (for each Web Reports server, assuming that port 8083 is used)

  Note: The buttons and links to log out from the Web UI and the Web Reports redirect these users to a page where they can click a Re-authenticate button to get back to Web UI and Web Reports pages without having to log back on, unless the IdP login timeout has expired; in this case they are brought back to the IdP login page.

- Must enable the **Use SAML authentication** check box in the Console login panel, if the BigFix server was configured to integrate with SAML V2.0.



The selection is automatically validated and retained by BigFix for future login requests.

Note: To override the `The security certificate was issued by a company you have not chosen to trust` Windows Security Alert warning:

Install the BigFix certificate (known as ServerSigningCertificate_0 by default) in the Trusted People store of Windows.

Local operators:

- Log in to the Web UI or to the Web Reports by accessing the usual login URLs:

  `https://<WebUI_server>/login` (assuming that the Web UI is set on port 443)

`https://<Web_Reports_server>:8083/login` (for each Web Reports server, assuming that Web Reports is set on port 8083)

- Log in to the BigFix Console from the usual login panel ensuring that the **Use SAML authentication** check box is not selected.

> **Note:** If SAML is not enabled in the environment, the **Use SAML authentication** check box is greyed out.

After SAML is configured and enabled only local operators will be able to log in using API; the 4-eyes authentication approvers must be local accounts.

How to configure BigFix to integrate with SAML 2.0

How to configure the SAML identity provider and the BigFix server.

Before configuring the integration, ensure that:

- The BigFix server can resolve the hostname used in the URL for the identity provider login page.
- The identity provider (ADFS server or another type of supported SAML authentication providers) can resolve the BigFix root server hostname specified in the redirect URLs used to communicate with the Web UI, Web Reports, and BigFix console.
- The Web UI is enabled and active.

The overall configuration comprises two parts:

- The configuration of the SAML identity provider for explicit two-factor authentication, which is under the responsibility of the identity provider administrator. For what concerns this part, ensure that:
  - The redirect URLs are added to the relying party trust indexed, with binding HTTPS_POST, and in this format:

    `https://<WebUI_server>/saml` (for the Web UI server, assuming that it listens on port 443)

`https://<Web_Reports_server>:8083/saml` (for each Web Reports server, assuming that they listen on port 8083)

`https://<Bigfix_server>:52311/saml` (for the BigFix Console)

Note: If the identity provider is ADFS, the redirect URLs must be added, as SAML Assertion Consumer Endpoints, in the Endpoints tab inside the ADFS Relying Party Trust properties.

- In the Identity Provider configuration, the login setting must be set for FORMS login.
- If you plan to use the smart card authentication, ensure that the Identity Provider is correctly configured to use multi factor authentication. For example, if you use ADFS, ensure that at least one between Certificate Authentication and Windows Authentication, if you want to use the Windows Integrated Authentication, is enabled in the Global Authentication Policy configuration.
- For Active Directory or Microsoft Entra ID user authentication, set the identity provider Claim Rules as follows:

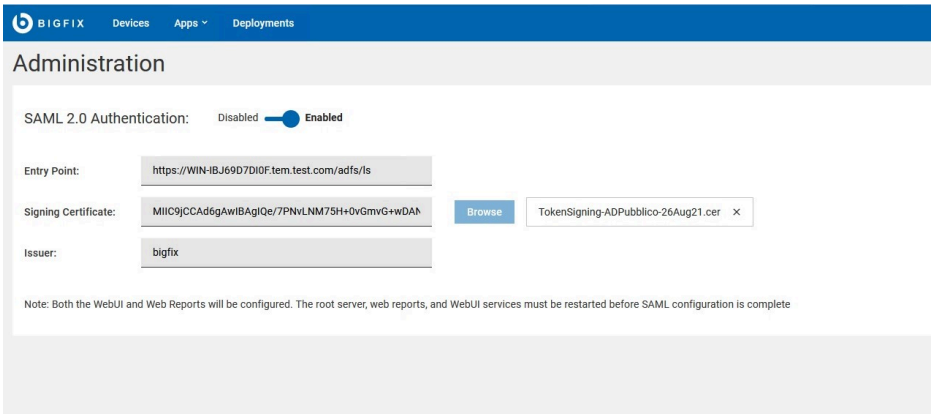
  Mapping of attributes to outgoing claim types:

  - Active Directory or Microsoft Entra ID Attribute: User-Principal-Name
  - Outgoing Claim: Name ID

- The configuration to allow the BigFix server to use SAML authentication, which is a Master Operator (MO) and Web Reports administrator responsibility. Complete these steps to accomplish this task:
  1. Configure LDAP with Active Directory or Microsoft Entra ID in the BigFix Console. For more details, see Integrating the BigFix Windows server with Active Directory *(on page 61)* and Integrating with Microsoft Entra ID *(on page 75)*.
  2. Define LDAP operators. For more details, see Adding Identity Provider Operators *(on page 84)*.
  3. Define Web Reports LDAP operators in the Web Reports user management pages.
  4. Access the Administration page to configure the integration with SAML 2.0:

a. Log in to the Web UI server:
   - If the Web UI listens on port 443: `https://<WebUI_server>`
   - If the Web UI listens on a port that is different than 443: `https://<WebUI_server>:<webui_port_number>`

b. Open the Administrator page:
   - If the Web UI listens on port 443: `https://<WebUI_server>/administrator`
   - If the Web UI listens on a port that is different than 443: `https://<WebUI_server>:<webui_port_number>/administrator`



5. In the Administration page, specify:

Entry Point:

The Identity Provider login URL. It is the URL from where the operator can log in and be redirected back to Web UI or to the Web Reports, for example `https://<idp_fqdn>/adfs/ls`.

Signing Certificate:

Browse for the certificate file or paste in this field the key from the Identity Provider certificate in Base-64 encoded X.509 (.CER) format.

Issuer:

Enter the Identity Provider Identifier in a textual format, for example "BigFix". If you are configuring ADFS configuration, this value must match the *ADFS Relying Party* Identifier setting. If you are using Microsoft Entra ID, the value must match the *Client ID* used when the Identity Provider was added (see Adding Microsoft Entra ID as Identity Provider *(on page 81)*).

6. After filling in all the fields, click **Enable**.
7. If WebUI is installed on a separate remote server, set the `_WebUI_AppServer_Hostname` key of the BigFix server computer to the hostname, fully qualified domain name (FQDN) or IP address of the computer where the WebUI is installed (the WebUI Server computer), ensuring that it matches the WebUI certificate subject name, as specified in **BES WebUI\cert \auth_cert.crt** on Windows and in **BESWebUI/cert/auth_cert.crt** on Linux. If the default WebUI port was changed (`_WebUIAppEnv_APP_PORT`), you must set the `_WebUI_Monitor_Port` key of the BigFix server computer to use the new WebUI port.

   Note: Ensure that the port of the WebUI server (default HTTPS 5000) is reachable by the BigFix root server.

8. If you want to enable the Web-based Single Sign-On (SSO) authentication method, on the WebUI machine set the _WebUIAppEnv_SAML_SSO_ENABLE key to 1.
9. Depending on your SAML authentication method (FORM, smart card, IWA, etc), you need to set the correct SAML Authentication Context using the WebUI setting _WebUIAppEnv_SAML_AUTHNCONTEXT. For details, see Server Settings Definitions.
10. Restart the BigFix root server.
11. Restart the BigFix Web Reports services.
12. Restart the WebUI Service.

It might take a while for the Web UI to restart after you set up this configuration and restarted the BES root server.

After these steps are successfully run, all LDAP operators from these services must authenticate through the configured identity provider.

An administrator can use the Administration page also to update the existing configuration.

Note: After completing these steps, to prevent errors when logging on to the BigFix console, ensure that you set for the `_BESDataServer_AuthenticationTimeoutMinutes` configuration setting a value, specified in minutes, bigger than 5 minutes.

The following links contain some configuration examples for well known Identity Providers:

- How to configure BigFix to authenticate using SAML SSO via ADFS
- How to configure BigFix to authenticate using SAML SSO via Okta
- How to configure BigFix to authenticate using SAML SSO via Azure AD

Configure SAML 2.0 authentication on other BigFix products

How to make BigFix products work with SAML 2.0.

Other BigFix products can take advantage of SAML integration. Links to specific product documentation follow:

- BigFix Inventory
- BigFix Remote Control
- BigFix Compliance
- BigFix MCM and BigFix Mobile
- BigFix WebUI

Chapter 5. Disabling local operators

Starting from BigFix Version 10.0.8, this feature provides a mechanism where the creation and use of any local operator is prohibited in favor of LDAP-based operators.

When the local operators are disabled:

- The login into the BigFix Console, Web Reports, Rest API and WebUI using a local operator is not allowed. The login using LDAP users is allowed.
- It is not possible to modify local or LDAP operators (for example, create a new operator, set explicit permissions, or directly associate BigFix roles with LDAP users).
- The roles are inherited according to LDAP groups - BigFix roles association.

Prerequisites:

Before disabling local operators and working only with LDAP users, it is necessary to perform the following steps:

1. Web Reports has been launched at least once in order to create a local administrative user.
2. An LDAP server has been configured on the BigFix Console and Web Reports.
3. LDAP groups have been associated with BigFix roles on the BigFix Console and Web Reports.
4. At least an LDAP group has been associated with a BigFix role having Master Operator permissions on the BigFix Console.
5. At least an LDAP group has been associated with the Administrator role on Web Reports.
6. After you disabled all local operators using the "Disabling local operators" feature, it is nevertheless required that:
    - At least one Local Master Operator must exist in the BigFix deployment.

   The still existing Local Master Operator does not represent an issue as it cannot be used until it is enabled.

Affected Components:

The BigFix components affected by this feature are the following:

- BigFix Administration Tool. For details about how the component is affected by the feature, see:

  BESAdmin Windows Command Line

  BESAdmin Linux Command Line
- BigFix Console. For details about how the component is affected by the feature, see:

  Disabling local operators on BigFix Console
- Rest API. For details about how the component is affected by the feature, see:

  Authentication
- Web Reports. For details about how the component is affected by the feature, see:

  Disabling local operators on Web Reports



Note:

If, for any reason, the information needed to disable the local operators is not present in the BigFix Database or is corrupted, it will not be possible to access the BigFix Console, Web Reports, Rest API and WebUI both with local operators and LDAP users.

Workaround:

To solve this issue, it is necessary to launch the BigFix Administration tool to enable or disable the local operators. For details, see the **securitysettings** described in BESAdmin Windows Command Line and BESAdmin Linux Command Line.

Limitations:

- When the local operators are disabled it is not possible to login into the REST API and SOAP API using a local operator. If you configure REST API or SOAP API credentials for the BES Server Plugin Service with a local operator and the local operators are disabled, the service will not work properly.
- When the SAML authentication is configured and the local operators are disabled, it is not possible to use REST APIs and 4-eyes authentication.

Chapter 6. Disaster Server Architecture

BigFix Distributed Server Architecture (DSA) has a highly sophisticated built-in ability to install multiple servers that will replicate information from each other for the purpose of disaster recovery. When configured and in the event of a failure of one server, other servers will automatically takeover as fully-functional servers (will receive data from the relays and clients and accept console connections). When the failed server is restored, it will automatically receive updated information.

DSA Expectations

The DSA architecture health is dependent upon the health and efficiency of the database replication process facilitated by the FillDB service. If actions are successfully propagated (in the Console) and the database has successfully replicated (see the Replication tab in the BigFix Administration tool) actions will run appropriately on all child endpoints.

However if the Primary DSA server is disabled after action propagation but before successful database replication then the secondary will NOT have the newly propagated actions. In this case, children of the Secondary server will not be provided the version of the actionsite that contains changes prior to replication and as such the Secondary will not receive the new actions or associated downloads for those actions. In this case the desired actions would need to be taken again from the Secondary.

In all cases, clients will continue to be provided an actionsite (containing open actions) for gathering and you can continue to manage the deployment and take new actions from the Secondary server.

- **Note**: Action completion time might look slower. For more information, see the section below "Completion time of action taken from Replication Server".

The term 'high availability' refers to disaster or event recovery that is immediate and indeed in any case a secondary DSA server is immediately available for deployment management. However, the actionsite and content data that is being replicated between the Primary and Secondary servers should not be considered 'high availability' as it depends on a database replication process rather than a real time load balanced and concurrent data process.

DSA Requirements

- The BigFix services on the DSA servers must all use the same database authentication method (i.e. all, NT Authenticated Domain Users/Groups or all SQL Authentication)
- The DSA servers must all run on the same OS server platform, type and version, (i.e. all Windows 2012 or All RHEL 7.2)
- The DSA servers should have similar hardware configurations and performance characteristics in terms of CPU, memory, disk, and overall system performance (otherwise your performance will suffer in the event of a failure) and replication may not work optimally or as expected.
- The DSA servers must have the same type and version of the SQL or DB2 (i.e. SQL Server 2014 or all DB2 10.5) database installed.

DSA Instructions

Click the following links to go to the topic you are interested in:

- Authenticating Additional Servers
- Installing Additional Windows Servers (DSA)
- Installing Additional Linux Servers (DSA)
- Uninstalling a Windows replication server
- Uninstalling a Linux replication server
- Managing Replication (DSA) on Windows systems *(on page 119)*
- Managing Replication (DSA) on Linux systems *(on page 121)*
- Configuring relay failover *(on page 116)*
- Message Level Encryption and DSA *(on page 118)*

Verify and Manage DSA Replication

The Replication tab of the BigFix Admin tool is the only user interface in BigFix to properly verify that replication is proceeding successfully between the Primary and Secondary servers. The tool reports important information such as **Server**, **Distance**, **Expected Latency**, **Last Replication Time**, and **Last Error Message** each of which can be used to troubleshoot any potential issues. Additional messages can be found in the FillDB.log files located in

the FillDBData folder (Windows: `\Program Files (x86)\BigFix Enterprise\BES Server`; RHEL: `/var/opt/BESServer`)

- **Note**: Be patient as the initial replication can and will take time depending upon the size of the database size and network latency between the DSA Servers.

The DSA Replication Interval

If you are using Distributed Server Architecture (DSA) and replication is failing with the error message 'Replication was interrupted to process server database insertions.' in the BigFix Administration tool, you will need to raise the maximum amount of time spent doing replication on the BigFix Server that is failing.

To increase the maximum replication time, set the following server setting on the BigFix DSA servers.

[HKEY_LOCAL_MACHINE\SOFTWARE\BigFix\Enterprise Server\FillDB]

UnInterruptableReplicationSeconds (DWORD): Seconds

Default: 30

Note: You must restart the FillDB service for changes to take effect.

By raising the value, the BigFix Server will spend more time performing replication each time it attempts to do so based on the replication interval. The error is caused because the BigFix Server is unable to complete replication using the default value.

For larger deployments of BigFix, try a value of 60-120 seconds. If you are installing a new BigFix Server, you might raise the value to 300-600 seconds during the initial replication period to reduce the amount of time spent initializing the new BigFix Server.

- **Caution:** Increasing the UnInterruptableReplicationSeconds extends the time given to the FillDB process for the purpose of DSA replication. However, it takes time away from the FillDB for inserting client reports in the database. If the UnInterruptableReplicationSeconds setting is set too high for too long it can backup the client reports on the relays and servers and cause a cascading reporting failure in the deployment. If this happens, clients will appear offline and reports from the endpoints will be be delayed.

Completion time of action taken from Replication Server

It might seem that the completion time of an iOS action is slower if the device is connected to a replication server instead of the master server.

In a Distributed Server Architecture, take actions to Clients only from the primary server. If you take actions to an iOS Client, the completion time might look slower because of the presence of a Proxy Agent and because of time when iOS/MDM server actually takes the action.

Chapter 7. Using multiple servers (DSA)

Some important elements of multiple server installations.

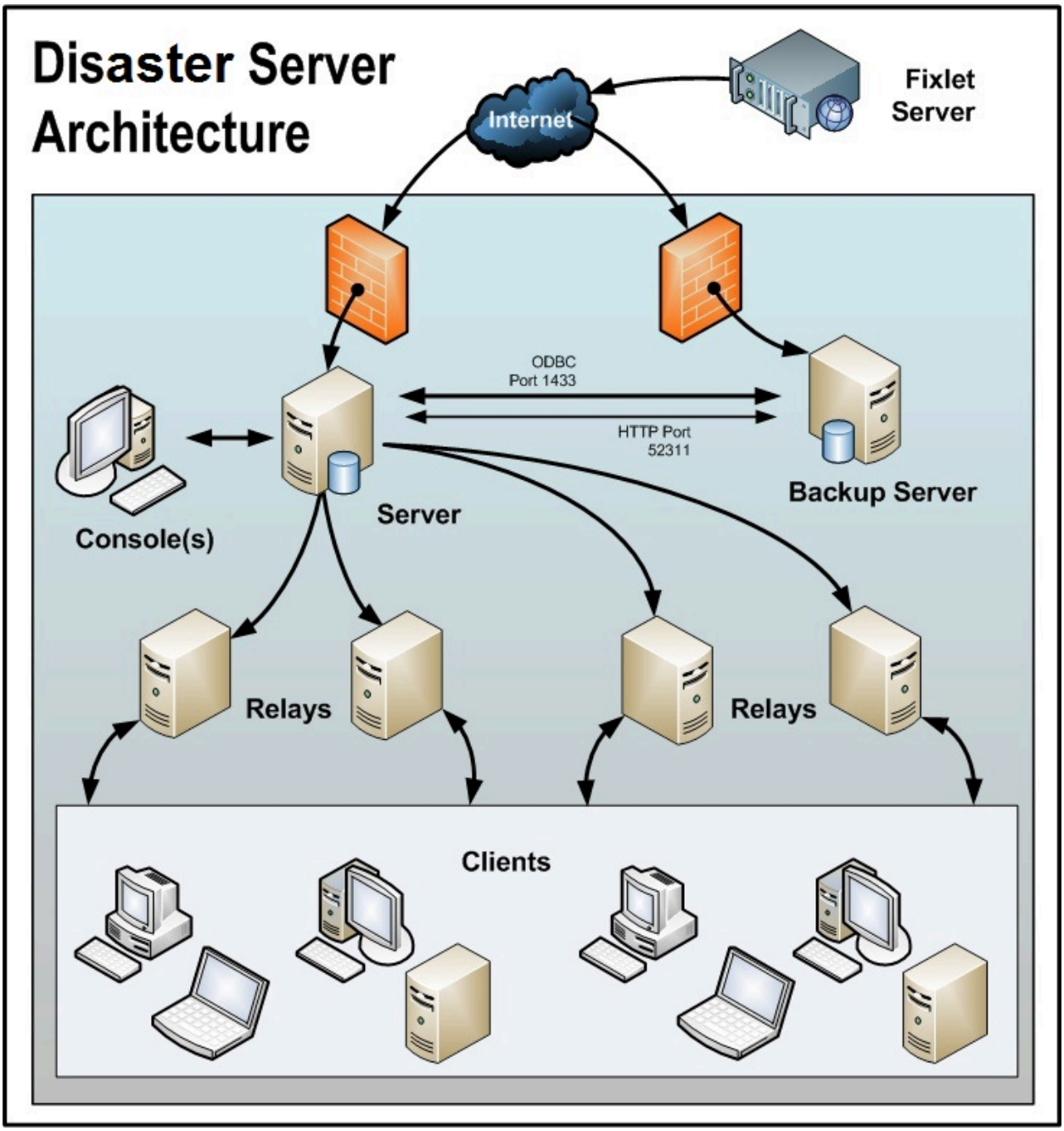
- Depending on the platform where you plan to install the additional server, you can follow the procedures described in Installing Additional Windows Servers (DSA) or Installing Additional Linux Servers (DSA).
- Servers communicate on a regular schedule to replicate their data. To review the current status and adjust the replication interval, see Managing Replication (DSA) on Windows systems *(on page 119)* or Managing Replication (DSA) on Linux systems *(on page 121)*.
- When each server is ready to replicate from the other servers in the deployment, it calculates the shortest path to every other server in the deployment. Primary links are assigned a length of 1, secondary links 100, and tertiary links 10,000. Links that resulted in a connection failure the last time they were used are considered to be non-connected.
- When an outage or other problem causes a network split, it is possible for a custom Fixlet or a retrieved property to be modified independently on both sides of the split. When the network is reconnected, precedence goes to the version on the server with the lowest Server ID.
- If multiple copies of **Web Reports** are installed, they operate independently. Each Web Report server can connect to the server that is most convenient, because they all contain equivalent views of the database.
- By default, server 0 (zero) is the master server. The **BigFix Administration Tool** on Windows and the **BESAdmin** command on Linux only allow you to perform certain administrative tasks (such as creating and deleting users) when connected to the master server.

Disaster Server Architecture (DSA)

The following diagram shows a typical DSA setup with two servers. Each Server is behind a firewall, possibly in a separate office, although it is easy to set up multiple servers in a single office as well.

The servers must have high-speed connections to replicate the BigFix data (generally LAN speeds from 10 to 100Mbps are required). The BigFix servers communicate over ODBC and HTTP protocols.

In case of a failover, the specific configured relays automatically find the backup server and reconnect the network. For more information about the relay configuration see Configuring relay failover (on page 116).



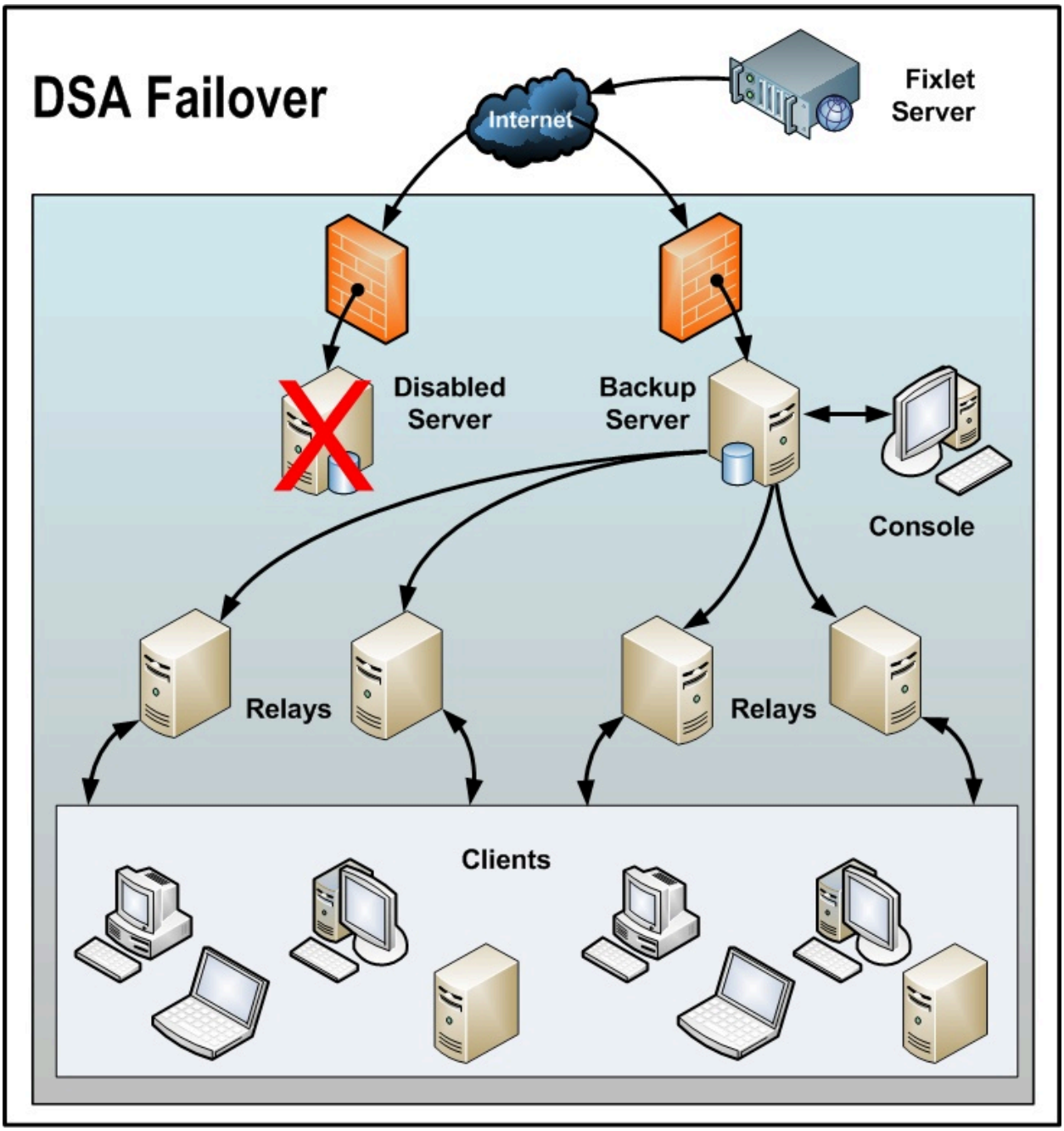
Configuring relay failover

If an BigFix server goes down, whether due to disaster or planned maintenance, the DSA server might be used to find a new server connection. When the disabled server comes back online, its data will automatically be merged with the data on the healthy server.

In order for the failover process to successfully occur set the DSA server as the secondary relay in client settings using `__RelayServer2` for the top-level relays (or via the console Computer right-click settings user interface). When a failure on the primary BigFix server occurs and lower level BigFix relays are unable to report, they use the secondary BigFix relay value during normal relay selection process to find and report to the secondary BigFix server.



Note: The setting `_BESClient_RelaySelect_ResistFailureIntervalSeconds` specified on the client system can have an impact on failover timing. Its value can range from 0 seconds to 6 hours and it defines how many seconds the client ignores reporting failures before attempting to find another parent relay. The default value is 10 minutes. In case of a failover configuration, ensure that, if defined, `_BESClient_RelaySelect_ResistFailureIntervalSeconds` is set to a low value.



Message Level Encryption and DSA

If Message Level Encryption is enabled and clients are set using **Task: BES Client Setting: Encrypted Reports**, move the BigFix server encryption key to the secondary BigFix DSA server.

This enables the BigFix DSA server to process reports from encrypted BigFix clients during normal operations or in the event of an outage on the primary BigFix server.

Copy the encryption key (`.pvk`) from the BigFix server directory:

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

to the DSA secondary server.

Managing Replication (DSA) on Windows systems

To install additional Windows servers, follow the procedure described in Installing Additional Windows Servers (DSA).

You might want to change the interval or allocate your servers differently. Most of these changes are done through the BigFix Administration Tool. Here you can see the current settings for your servers and make the appropriate changes.

Changing the replication interval on Windows systems

On Windows systems if you have multiple servers in your deployment, you can schedule when each one replicates.

The default is five minutes, but you can shorten the time for greater recoverability or increase it to limit network activity:

1. Start up the **BigFix Administration Tool**.
2. Select the **Replication** tab.
3. Click the Refresh button to see the latest **Replication Graph**.
4. Select the server you want from the drop-down menu. Using longer replication intervals means that the servers replicate data less often, but have more data to transfer each time. Note that replication intervals can be different for replicating from and replicating to a server.

5. Select the replication interval from the menu on the right.
6. Click **OK**.

Switching the master server on Windows systems

By default, server 0 (zero) is the master server. The Administration Tool allows you to perform certain administrative tasks (such as creating and deleting users) only when you are connected to the master server.

If you want to switch the master to another server, you must set the deployment option **masterdatabaseServerID** to the other server ID. Here is how:

1. Start up the **BigFix Administration Tool.**
2. Select the **Advanced Options** tab and click **Add**.
3. Type masterDatabaseServerID as the name, and then enter the other server ID as the value.
4. Click **OK**.

After the value has successfully replicated to the new server, it becomes the master server.

When a server suffers a failure while it is the master, if you cannot use the **BigFix Administration Tool**, instead you can use the following alternative procedure:

1. From the `BES Server\IEM CLI` folder, using a command prompt, run the command:

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

   followed by the command:

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

2. In the switchmaster.xml file, created previously during step 1, add or edit the following keyword and its value:

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

3. To switch the master server to another one, in this example with ID 3:

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

4. Run the following command to modify the value:

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

Managing Replication (DSA) on Linux systems

To install additional Linux servers, follow the procedure described in Installing Additional Linux Servers (DSA).

You might want to change the interval or allocate your servers differently. Most of these changes are done through the `iem` command line. Here you can see the current settings for your servers and make the appropriate changes.

Changing the replication interval on Linux systems

On Linux systems if you have multiple servers in your deployment, you can schedule when each one replicates.

The default is five minutes, but you can shorten the time for greater recoverability or increase it to limit network activity:

To change the replication interval, perform the following steps:

1. From the `/opt/BESServer/bin` command prompt, start the command line:

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

2. From the `/opt/BESServer/bin` command prompt, run the following command:

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

3. In the `/appo/replicationServer0.xml` file, edit the following keyword:

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

to change the value in seconds of the replication interval. Using longer replication intervals means that the servers replicate data less often, but have more data to transfer each time.

```
<?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. Upload the modified file by running the following command:

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

Switching the master server on Linux systems

By default, server 0 (zero) is the master server.

If you want to modify the master server ID using the BigFix Administration Tool, perform these steps:

1. Modify the actual value for the master server ID to, for example, 3 by running the command:

```
/opt/BESServer/bin/BESAdmin.sh -setadvancedoptions
 -sitePvkLocation=<path+license.pvk> -update masterDatabaseServerID=3
```

2. You can verify the value by running the command:

```
/opt/BESServer/bin/BESAdmin.sh -setadvancedoptions
 -sitePvkLocation=<path+license.pvk> -display
```

After the value has successfully replicated to the new server, it becomes the master server. If a server suffers a failure while it is the master, you cannot use the **BigFix Administration Tool**, instead you can use the following alternative procedure:

To switch the master to another server, set the deployment option `masterDatabaseServerID` to the other server ID as follows:

1. From the `/opt/BESServer/bin` command prompt, start the command line:

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

2. From the `/opt/BESServer/bin` command prompt, run the following command:

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

3. In the `/tmp/switchmaster.xml` file, created previously during step 2, add or edit the following keyword and its value:

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

   to switch the master server to another master server, in this example with ID 3:

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

4. Run the following command to modify the value:

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

Chapter 8. Server object IDs

The BigFix server generates unique ids for the objects that it creates: Fixlets, tasks, baselines, properties, analysis, actions, roles, custom sites, computer groups, management rights, subscriptions.

These ids are stored as 32-bit fields into the Platform database, such as:

- ActionID
- FixletID
- ID
- ContentID
- RoleID

The id is displayed in the Console, Web Reports and WebUI interfaces and is used by the REST APIs and tools like AdminTool and PropertyIDMapper.

Before the current implementation, the maximum number of available object ids per server was 16.777.215, and consequently the maximum number of available DSA servers was 256.

To avoid reaching the object id limit when creating Fixlets, tasks, baselines and so on, the bits used for the object id have been rearranged as follows:

|_x__|___y__|_____________z_____________|

where:

- x= 3 bits, 2 of which to be used for the counter (the first bit is used for agent internal processing)
- y= server id (5 bits instead of the previous 8)
- z= initial 24 bits for the counter (unchanged)

In this way, the number of available object ids was increased to 1.627.389.951 and consequently the number of available DSA servers was decreased to 32.

This solution has the advantage of keeping the 32-bit object id so to avoid any backward compatibility issue.

Chapter 9. Using the DHE/ECDHE key exchange method

By default, BigFixVersion 11 components use the DHE/ECDHE key exchange method if the version of the BigFix component on the other side of the SSL communication allows it.

To use DHE/ECDHE in all SSL communications, all BigFix components must be at any of the following versions:

- Version 10.0 Patch 1 or higher.
- Version 9.5 Patch 16 or higher.

Other considerations

- BigFix HTTPS servers use RSA for both authentication and key exchange.
- BigFix enables ephemeral Diffie-Hellman (DHE) and ephemeral elliptic curve Diffie-Hellman (ECDHE) for key exchange (RSA for authentication).
- Ephemeral means new, random asymmetric keys are chosen for each TLS connection that are never written to persistent storage.
- When the TLS connection terminates, keys are securely erased.
- This means if an RSA private key is ever divulged, that key cannot be used to decrypt any recorded TLS sessions.
- Any secrets exchanged in those TLS sessions, such as BigFixConsole passwords, REST API passwords, Web Reports passwords and session tokens will not be divulged in the event of an RSA private key disclosure.
- To determine the protocols in place, you can use the Nmap utility. A sample invocation is:

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

Chapter 10. Configuring secure communication

Customizing HTTPS for Gathering

This page describes how to customize HTTPS for Gathering purpose. For more details about customizing HTTPS for downloads, refer to Customizing HTTPS for downloads (on page 129).

You can gather license updates and external sites by using the HTTP or HTTPS protocol on a BigFix server or in an airgapped environment.

HTTPS is the default protocol.

Enabling HTTPS, you can create or download (from the curl website) a package of certificates that you want to trust. The curl website offers a prebuilt package that contains the same certificates that are included with Mozilla.

The BigFix server starts the certificate verification during gathering, trusting the provided certificates.

Managing HTTPS

To gather the external sites by using the HTTPS protocol, complete the following steps

On the BigFix Server:

Set the client property `_BESGather_Use_Https` to `0`, `1` or `2`.

When setting the property to 0, the server uses the protocol defined in the URL.

When setting the property to 1, the server tries to gather all sites using the HTTPS protocol only.

When setting the property to 2, the server first tries to gather all sites using the HTTPS protocol. If the server fails to gather a site using HTTPS, it will try to gather again using the HTTP protocol. The fallback from HTTPS to HTTP only applies to sites having URLs starting with `http://`

The default value for this setting is 2.

In the airgapped environment:

Launch the `Airgap` command as follows:

```
Airgap
```

The server tries first to gather all sites using the HTTPS protocol. In case of failure, the server will gather the sites using the HTTP protocol. This redirection applies only if the URL is hard-coded with HTTP. This is the default behavior.

```
Airgap -usehttps
```

The server tries to gather all sites using the HTTPS protocol only.

```
Airgap -no-usehttps
```

The server uses the protocol defined in the URL.

Validating HTTPS certificates

By default the HTTPS certificates used for enabling the HTTPS connection are validated by using the certificate bundle included in the BigFix server installation.

The Windows default path is:

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

The Linux default path is:

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

To validate the HTTPS certificates with a custom bundle of trusted certificates before the HTTPS gathering, complete the following steps:

1. Create or download a set of trusted certificates (for example, http://curl.haxx.se/ca/cacert.pem). The certificates that you can use are:
    - "VeriSign Universal Root Certification Authority" (to gather sites)
    - "thawte Primary Root CA - G3" (to check license updates)

2. **On the Server:**

   Set the client property `_BESGather_Use_Https` to `1` or `2` for using the HTTPS protocol and `_BESGather_CACert` keyword to the path of the downloaded set of trusted certificates (for example `c:\TEM\certificates\custom-ca-bundle.crt` on Windows systems and `/TEM/certificates/custom-ca-bundle.crt` on Linux systems).

   In the airgapped environment:

   Launch the Airgap tool with the option `-cacert <path>`:

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

   where `<path>` is the path of the saved set of trusted certificates.

Customizing HTTPS for downloads

This page describes the usage of CA bundles in HTTPS downloads and how to customize them.

On this topic, the word “downloads” refers exclusively to the effect triggered by the action commands prefetch and download. Such commands might also be invoked by BigFix applications (such as BigFix Inventory) to download catalogs or other required data.

Site gathering and/or communication between BigFix components (for example, server – client communication) are not affected by this topic.

For more details about customizing HTTPS for gathering, refer to Customizing HTTPS for Gathering (on page 127).

Overview

The main purposes of HTTPS protocol are authentication of the accessed website and protection of the privacy, as well as integrity of the exchanged data. This is obtained through the verification of the remote server certificate, using a trusted third party (Certification Authority), and the channel encryption. The check is performed during the handshake, and, if it fails, the connection is dropped. It protects the communication against malicious attacks.

Starting from Version 10 Patch 8, to support the download from an HTTPS URL, it is necessary to provide appropriate trusted certificates to verify the remote server identity.

Public CA, Private CA and self-signed certificates

Majority of BigFix Content works without additional changes in Platform configuration. Customization is needed only in specific cases, depending on the type of certificate used:

- Certificates signed by external / public CA: no action needed.
- Certificates signed by internal / private CA: needs the certificate to be added. See Custom certificates *(on page 131)*.
- Self-signed certificates: needs the certificate to be added. See Custom certificates *(on page 131)*.

Certificates shipped with BigFix

The curl website offers a prebuilt package that contains the same certificates that are included with Mozilla.

This package is copied in a certificate bundle named `ca-bundle.crt` and distributed in the BES Support site. The certificate bundle is searched in different locations, depending on the BigFix version:

- In Version 10 Patch 8, both the server/relay and the client load the CA bundle from the BES Support site directory under the BigFix client installation path.

  Note: If the BigFix server/relay is running without administration privileges, for example a Domain user on Windows AD, it will not be able to read the folder located under the client path: in that case, a custom CA bundle must be specified. See the section below named Custom certificates.

- Starting from Version 10 Patch 9, for the BigFix server/relay, the bundle is loaded from the last gathered BES Support folder under the BigFix server/relay directory. This folder is always accessible, regardless of the privileges accorded to the process. For example, on a UNIX platform the bundle path could be:

```
/
var/opt/BESServer/wwwrootbes/bfmirror/bfsites/enterprisemirror_<SiteID
>_<SiteVersion>
```

In case the ca-bundle.crt file is not present or cannot be read, the server/relay will fall back on the default client path, as for Version 10 Patch 8. For what concerns the BigFix client, the bundle location is the same as Version 10 Patch 8.

Notice that BigFix does not perform any active monitoring on certificates validity and/or their expiration dates. No warning is provided prior to certificate expiration. Using expired or invalid certificates will result in download failure.

Enabling HTTPS downloads from untrusted sites

Starting from Version 10 Patch 8, the following settings can be used to allow or disallow the download from untrusted sites:

- **_BESRelay_Download_UntrustedSites** (server/relay). It is a boolean setting. When it is set to 0 (default value), the downloads from untrusted sites are not allowed. When it is set to 1, the downloads from untrusted sites are allowed.
- **_BESClient_Download_UntrustedSites** (client). As described above, but the setting works for the client.

Custom certificates

Starting from Version 10 Patch 8, custom CA bundles (included self-signed certificates) can be set for the components: server, relay and client. The setting changes according to the patch version.

In Version 10 Patch 8 and later, the following parameters can be set:

- **_BESRelay_Download_CACertPath** (server/relay). It is a string setting where you can declare the full path of a custom CA bundle. It overrides the default one.
- **_BESClient_Download_CACertPath** (client). As described above, but the setting works for the client.

Note: These parameters are still recognized in Version 10 Patch 9 and later, but they are considered as deprecated and should not be used. This is done for retro-compatibility purpose. Their behavior is slightly different, though: unlike Version 10 Patch 8, the custom bundle will be appended to the other certificates, and will not override them. Versions from 10.0.9 and later are also provided with a specific parameter to ignore the default certificates (see below).

Default directory for custom certificates

In Version 10 Patch 9, a predefined folder is introduced for all the BigFix components (server, relay and client), `<StorageFolder>/TrustedDownloadCerts`, that can be used to store custom certificates and/or bundles. All the `.crt` and `.pem` files contained in this folder are added to the default certificates when an HTTPS download is performed. For example, on a default Linux/UNIX installation, the folder full path for the client custom CA bundle is: `/var/opt/BESClient/TrustedDownloadCerts`, while the full path for the relay custom CA bundle is `/var/opt/BESRelay/TrustedDownloadCerts`.

Only files with extensions `.pem` and `.crt` are loaded, and they can contain both certificates bundles or single certificates. No check is performed on such files: if a file contains invalid data, it might cause a download failure.

The following is a sample task that creates a `.crt` file inside the `<StorageFolder>/TrustedDownloadCerts`. This sample can also be used as a template for understanding how to retrieve the `<StorageFolder>/TrustedDownloadCerts` directory for server, relays and clients.

```
<?xml version="1.0" encoding="UTF-8"?>
 <BES xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:noNamespaceSchemaLocation="BES.xsd">
  <Task>
   <Title>Add Download Certificate</Title>
   <Description><![CDATA[Sample Task for creating and downloading a custom
 certificate.]]></Description>
```

```
  <Relevance>if exists property "in proxy agent context" then ( not in
proxy agent context ) else true</Relevance>
  <Category></Category>
  <Source>Internal</Source>
  <SourceID></SourceID>
  <SourceReleaseDate>2023-02-14</SourceReleaseDate>
  <SourceSeverity></SourceSeverity>
  <CVENames></CVENames>
  <SANSID></SANSID>
  <MIMEField>
   <Name>x-fixlet-modification-time</Name>
   <Value>Tue, 14 Feb 2023 15:04:24 +0000</Value>
  </MIMEField>
  <Domain>BESC</Domain>
  <DefaultAction ID="Action1">
   <Description>
    <PreLink>Click </PreLink>
    <Link>here</Link>
    <PostLink> to deploy this action.</PostLink>
   </Description>
   <ActionScript MIMEType="application/x-Fixlet-Windows-Shell"><![CDATA[
// This is a sample Task for reading the TrustedDownloadCerts folder and
copying a certificate
// This Task will add a certificate in the bundle.
// The certificate will be added in the Server folder if the Server is
installed, on the Relay folder if the relay is installed, in the Client
folder otherwise

if {exists main gather service}
    if {windows of operating system}
```

```
      parameter "storageDir" = "{value "StoragePath" of key
 "HKEY_LOCAL_MACHINE\SOFTWARE\BigFix\Enterprise Server\GlobalOptions" of
 registry as string}"
  else
      parameter "storageDir" = "{key ("StoragePath") of section
 ("Software\BigFix\Enterprise Server\GlobalOptions") of file
 "/var/opt/BESServer/besserver.config"}"
  endif
 elseif {exists relay service}
     if {windows of operating system}
      parameter "storageDir" = "{value "StoragePath" of key
 "HKEY_LOCAL_MACHINE\SOFTWARE\BigFix\Enterprise Server\GlobalOptions" of
 registry as string}"
  else
      parameter "storageDir" = "{key ("StoragePath") of section
 ("Software\BigFix\Enterprise Server\GlobalOptions") of file
 "/var/opt/BESRelay/besrelay.config"}"
  endif
 else
     parameter "storageDir" = "{storage folder of client}"
 endif

 delete __appendfile

 appendfile -----BEGIN CERTIFICATE-----
 appendfile
 MIIDhTCCAm2gAwIBAgIEFtnq9zANBgkqhkiG9w0BAQsFADBbMScwJQYDVQQDDB5SZWdlcnkgU2
V
 appendfile
 Zi1TaWduZWQgQ2VydGlmaWNhdGUxIzAhBgNVBAoMGlJlZ2VyeSwgaHR0cHM6Ly9yZWdlcnkuY2
9
```

```
 appendfile
 MQswCQYDVQQGEwJVQTAgFw0yMzAyMjcwMDAwMDBaGA8yMTIzMDIyNzEwMDYyMFowRzETMBEGA1
U
 appendfile
 AwwKbXlzaXRlLmNvbTEjMCEGA1UECgwaUmVnZXJ5LCBodHRwczovL3JlZ2VyeS5jb20xCzAJBg
N
 appendfile
 BAYTAlVBMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAh8F7GgL4IQrc8E4AHJBmdY
u
 appendfile
 X7DEJ0X5XnXeD+tj5XVe7AE3lsoFqLkrNo/mWzZrp9y6ZHoew9AaLrJgUmFdU5c/1aUhWlVtB
U+
 appendfile
 tOBqnZNRKEy+AsObZ2lYryNTVwAbTieTLVPM4pDNr6dwZen1tWfVIKg/2NiApTjwCdamalljP2
J
 appendfile
 V
+NxLsjFhGg8q8lD0JJiqoNUm1bi22JrqWJ8Z9cXXmuiNHzUxzVW7XaxzXyWXE4WNWZMIO/gUl5
 appendfile
 mP9VjXvyXymzMoeDxjMg9cyyrni5ZJsv8Uqkq6ni/Tml6QOYahjNOYmHF8j9v+vTLI/YwfFF4l
U
 appendfile
 QXcOXsHxWyepkwIDAQABo2MwYTAPBgNVHRMBAf8EBTADAQH/MA4GA1UdDwEB/wQEAwIBhjAdBg
N
 appendfile
 HQ4EFgQUeokEssftpr4W/DjX8EDSYkr9V3AwHwYDVR0jBBgwFoAUeokEssftpr4W/DjX8EDSYk
r
 appendfile
 V3AwDQYJKoZIhvcNAQELBQADggEBAE64aGOnSfA0ePiu6CguxH5GKojTM6N7z2ouzPJGymrVPd
M
```

```
 appendfile
 jNfb81Xm+VJ0Eervw8uz/D3NYVCJy4/Q
+gvAyYGHFmRxmIue/4RjsOK482WkeICGTsZWggoz3wX
 appendfile
 12ZX13noMCULcU1lI83eNBn3LFvA/bYH7cwSBGxmLidg1jkbATktZMQYJXIQ6cNtst4NgfMZPz
j
 appendfile
 w6b9+1tf6ohk+zjYNWPf3N58mziQ3JayWaZFqCHkPVvJYITUPySf4vkhcBNvkWF2W+w4SujpqA
E
 appendfile
 xAdZcMwfwCgUGpyAPeNfPxmQcEwVpDyhK+jfMWLB74swoq/1/k2EwKd4rq/PcOoC394=
 appendfile -----END CERTIFICATE-----

 move "__appendfile" "{parameter "storageDir" &
 "/TrustedDownloadCerts/testCert.crt"}
 ]]></ActionScript>
   </DefaultAction>
  </Task>
 </BES>
```

Additional certificates

Starting from Version 10 Patch 9, in addition to `<StorageFolder>/TrustedDownloadCerts`, a custom directory can be specified using the following parameter:

- **_BESRelay_Download_CaCertDirectory** (server/relay). It is a string setting where you can declare the full path of a folder where the user stores its custom CA bundles. All the `.crt` and `.pem` files contained in this folder are added to the default certificates.
- **_BESClient_Download_CaCertDirectory** (client). As described above, but the setting works for the client.

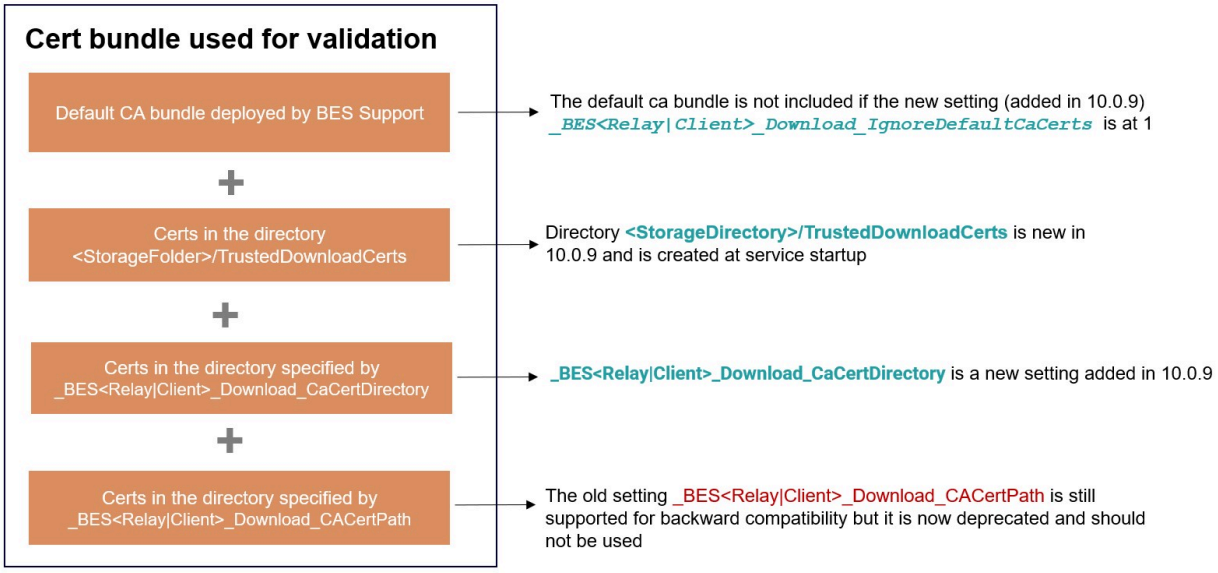
Disabling default certificates

Also, starting from Version 10 Patch 9, a specific parameter can be set in order to ignore the default certificates and exclusively use the custom ones:

- **_BESRelay_Download_IgnoreDefaultCaCerts** (server/relay). It is a boolean setting with two possible values: 0 and 1. When set to 1, it tells the component to ignore the default `ca-bundle.crt` certificate bundle located under BES Support. The default value is 0.
- **_BESClient_Download_IgnoreDefaultCaCerts** (client). As described above, but the setting works for the client.

Summary

The following figure shows a summary of the locations used starting from Version 10 Patch 9 to build the list of certificates. All contributions are appended to the certificate list, unless expressly ignored or empty.



For more details about download settings, see Download (on page 766).

Troubleshooting hints

On Linux/UNIX platforms, the curl utility can be used to verify that a custom certificate bundle is valid and working for a specific download through HTTPS.

For example, on a Linux/UNIX server installation, the default certificate bundle can be tested with the following command, provided that the download URL is `https://download.server.url/file-to-download`:

```
curl --cacert
/
var/opt/BESServer/wwwrootbes/bfmirror/bfsites/enterprisemirror_<SiteID>_<SiteVersion>
/ca-bundle.crt -o file-to-download
 https://download.server.url/file-to-download
```

Note that you need to substitute the parameters <SiteID> and <SiteVersion> to get the actual directory name for the last version of the BES Support site.

To test a custom certificate bundle, the following command can be run (make sure to substitute the parameters with the appropriate values):

```
curl --cacert </path-to-certificate/custom-certificate.pem>
-o file-to-download https://download.server.url/file-to-download
```

Note: In some Windows versions, PowerShell maps the cmdlet Invoke-WebRequest to the curl executable; however, this does not provide a reliable method for certificate verification. Use openSSL instead, as described below.

On a Windows platform, the same test can be performed using openSSL. In the following example, `download.server.url` is the address of the HTTPS server and `C:\path\to\certificate\certificate-bundle.pem` is the local path to the certificate bundle.

```
openssl.exe s_client -connect download.server.url:443 -CAfile
C:\path\to\certificate\certificate-bundle.pem
```

In this case, it is necessary to check the return code of the command in its output, which may be composed by many lines: the ones starting with “Verify return code” are those to look for. Possible examples are:

- Verify return code: 0 (ok)
- Verify return code: 20 (unable to get local issuer certificate)
- Verify return code: 18 (self signed certificate)

Configuring custom certificates

Things to consider when configuring custom certificates.

Private key and certificate format

Ensure that the private key and the certificate files have the following format and structure.

Private key format

PEM-encoded and without a password protection. The pvk format is not supported. Ensure that the private key (*private.key*) is enclosed between the following statements:

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

X509 certificate format

PEM-encoded. If you have also received the intermediate and root certificates as separate files, you should combine all of them into a single one. For example, if you have the primary certificate file (*certificate.crt*) and the intermediate certificate file (*ca_intermediate.crt*), ensure that you combine them in the following order, primary certificate first followed by the intermediate certificate:

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

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

If you received the root certificate (*ca_root.crt*) in addition to the intermediate certificate, combine them as follows:

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

Single file (private key with certificates) format

PEM-encoded. This file can contain both the private key and the primary certificate, or the private key and the chain of certificates, combined in the following order, and with the beginning and end tags on each certificate:

- Private key and primary certificate:

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

- Private key, primary certificate and intermediate certificate:

```
-----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-----
```

- Private key, primary certificate, intermediate certificate and root certificate:

```
-----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-----
```

If your file has DER-encoded or other formats, you can convert it to the PEM format, for example by using OpenSSL.

Creating a Certificate Signing Request (csr)

Procedure to register a certificate.

1. You need a valid configuration file such as the following one:

```
[ 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. Replace `Common` with the fully qualified domain name of the Web Reports server.
3. Create the certificate request `cert.csr` with the following command.

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

   This also generates the private key called `keyfile.pem`.
4. Remove the password from the private key file `keyfile.pem` and generate a new private key (`nopwdkey.pem`) using the following command:

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

Generating a Self-Signed Certificate

Procedure to generate a self-signed certificate (`cert.pem`) from a certificate request file (`cert.csr`).

Perform the following steps:

1. Create a Certificate Signing Request (`cert.csr`).
2. Create a certificate file (`cert.pem`) from your private key (`nopwdkey.pem`) and certificate request file (`cert.csr`) using the following command (valid for 365 days):

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



Important: The following steps explain how to combine the private key file with the signed certificate file for convenience in later configuration steps. If you prefer, you can use them separately and skip the following steps.



Note: You can use a key pair generated for BigFix Inventory and License Metric Tool also for Web Reports only if the private key is not password protected.

3. Open up your private key file `nopwdkey.pem` in Notepad++, or another text editor.
4. Copy the contents and paste them below the certificate in `cert.pem`, as in the following example:

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

where `...` represents any text.

5. Refer to `cert.pem` on your Web Reports server in the certificate path registry setting as described in Customizing HTTPS on Web Reports.

Requesting a Certificate from a Certificate Authority

To encrypt HTTPS Web Reports with a certificate that browsers implicitly trust, request a signed certificate from a trusted Certificate Authority (or CA) such as Verisign as follows.

1. Create a Certificate Signing Request (csr) *(on page 141)*
2. Forward the `.csr` file to a Certificate Authority (CA). They will issue you a signed (browser-trusted) certificate for your server. Request the certificate as a `.pem` file that includes the entire trust chain.

   Important: The following steps explain how to combine the private key file with the signed certificate file for convenience in later configuration steps. If you prefer, you can use them separately and skip the following steps.

   Note: You can use a key pair generated for BigFix Inventory and License Metric Tool also for Web Reports only if the private key is not password protected.
3. After you have received the signed certificate file, DO NOT import it to any Microsoft default certificate handling facilities.
4. Open the private key file from which you removed the password (`nopwdkey.pem`), and copy its content to the clipboard.
5. Open the signed certificate file with Notepad++, or another text editor.
6. Append the content copied in step 4 to the signed certificate file. This is an example of the resulting content:

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

```
...
-----END RSA PRIVATE KEY-----
```

where `...` represents any text.

7. Save the modified `.pem` file containing the public certificate and private key.
8. Store this file on your server and refer to it when setting up your Web Reports.

Customizing HTTPS on Web Reports

For details about how to customize HTTPS on Web Reports, see Customizing HTTPS on Web Reports.

Customizing HTTPS on BigFix Explorer

The BigFix Explorer is configured to use HTTPS by default when it gets installed and creates its own certificate during the installation. If you want to replace it, you need to configure HTTPS manually.

First steps

If you have a trusted SSL security certificate and key from a certificate authority, you can configure the BigFix Explorer to use this certificate and key to enable trusted connections. You can also use a self-signed certificate.

When you have a trusted SSL certificate, copy the .pvk (if you have one) and the .pem files on the computer running the BigFix Explorer.

In the following sections, we show ways to implement these macro-steps:

- Specify that you are using a secure communication.
- Specify where the SSL certificate and private key files are located.
- Restart the relevant services.

After you have completed the configurations described in the following sections, the connections from the Rest API and the BigFix Console use this trusted certificate.

Customizing HTTPS using the BigFix Console

1. From the BigFix Console select the **Computers** tab.
2. Select the computer running Rest API (usually the server) and **Edit Computer Settings** from the **Edit** menu.
3. Look for **_BESExplorer_HTTPServer_SSLPrivateKeyFilePath** setting. If it exists, do not create a second one, but edit its value to the full path name of the private key (.pvk file which contains the private key for the server). The private key must not have a password. If this setting does not exist, add it.
4. Look for **_BESExplorer_HTTPServer_SSLCertificateFilePath** setting. If it exists, do not create a second one, but edit its value to the full path name of the .pem file which might contain both the certificate and private key for the server, or only the certificate. If this setting does not exist, add it. Ensure that the `.pem` file is in standard OpenSSL PKCS7 .pem file format.

   The certificate is supplied by the server to connecting clients and they present a dialog to the user containing information from the certificate. If the certificate meets all of the trust requirements of the connecting client, then the client connects without any interventions by the user. If the certificate does not meet the trust requirements of the client, then the user will be prompted with a dialog asking them if it is OK to proceed with the connection, and giving them access to information about the certificate. A trusted certificate is signed by a trusted authority (such as Verisign), contains the correct host name, and is not expired.
5. Restart the **BES Explorer** service:
   - On Windows, open **Services**, select **BES Explorer** and on the **Action** menu, click **Restart**.
   - On Linux run from the prompt: `service besserver restart` or `/etc/init.d/besserver restart`.



Note: These settings are stored in the registry under the key **HKLM/Software/WoW6432Node/BigFix/EnterpriseClient/Settings/Client**

Customizing HTTPS manually

If you have a trusted SSL security and a key from a certificate authority (`.pem` file), you can configure the computer running REST API (usually the server) to customize trusted connections.

On Windows systems

To customize HTTPS manually on Windows systems, complete the following steps:

1. Run **regedit** and locate `HKEY_LOCAL_MACHINE\Software\Wow6432Node\BigFix\EnterpriseClient\Settings\Client`

   You need to add or modify subkeys for the HTTPS flag, and for the location of the SSL certificate.
2. **Important: If you combined the private key file with the certificate file, move to step 3.**

   Create a subkey of **Client** called `_BESExplorer_HTTPServer_SSLPrivateKeyFilePath` (if it does not exist yet). Add a string value (reg_sz) called "value" to the key and set it to the full path name of the private key (.pvk file which contains the private key for the server).
3. Create a subkey of Client called `_BESExplorer_HTTPServer_SSLCertificateFilePath` (if it does not exist yet). Add a string value (reg_sz) called "value" to the key and set it to the full path name of the SSL certificate (cert.pem).
4. Restart the `BES Explorer` service.

On Linux systems

To customize HTTPS manually on Linux systems, complete the following steps:

Save the files `cert.pem` and `pvtkey.pvk` (if you have it) in a protected area of the file system, where it can be accessed by the BigFix besserver process, for example, `/etc/opt/BESServer/`.

Edit the `/var/opt/BESServer/besserver.config` file, by adding the following entries.

Important: If you combined the private key file with the certificate file, skip these settings.

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

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

Stop and restart the BigFix Explorer.

Customizing HTTPS on REST API

The BigFix root server is configured to use HTTPS by default when it gets installed and creates its own certificate during the installation. If you want to replace it, you need to configure HTTPS manually.

First steps

If you have a trusted SSL security certificate and key from a certificate authority, you can configure the BigFix root server to use this certificate and key to enable trusted connections. You can also use a self-signed certificate.

When you have a trusted SSL certificate, copy the .pvk (if you have one) and the .pem files on the computer running the BigFix root server.

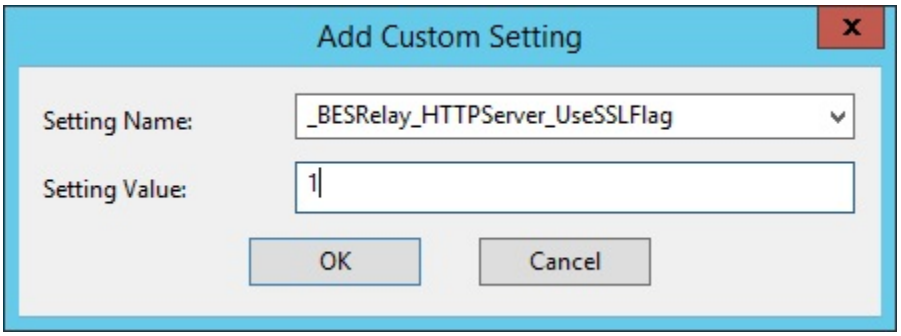
In the following sections, we show ways to implement these macro-steps:

- Specify that you are using a secure communication.
- Specify where the SSL certificate and private key files are located.
- Restart the relevant services.

After you have completed the configurations described in the following sections, the connections from the Rest API and the BigFix Console use this trusted certificate.

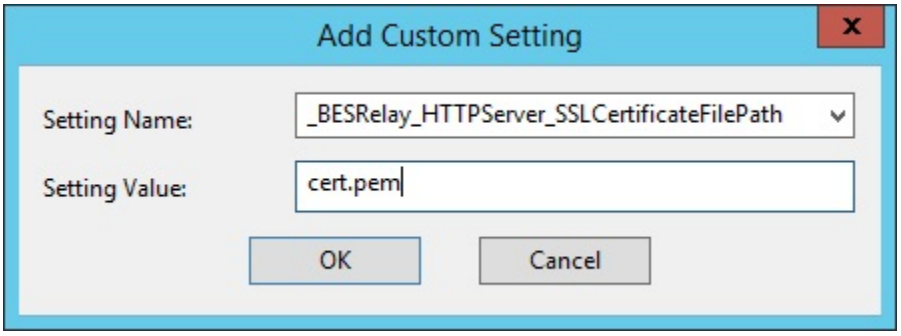
Customizing HTTPS using the BigFix Console

1. From the BigFix console select the **Computers** tab.
2. Select the computer running Rest API (usually the server) and **Edit Computer Settings** from the **Edit** menu.
3. Look for **_BESRelay_HTTPServer_UseSSLFlag** setting. If it exists, do not create a second one, but edit its value to `1` to enable HTTPS. If it does not exist, add it:



4. If you combined the private key file with the certificate file, skip this step and set only the **_BESRelay_HTTPServer_SSLCertificateFilePath**.

   Look for **_BESRelay_HTTPServer_SSLPrivateKeyFilePath** setting. If it exists, do not create a second one, but edit its value to the full path name of the private key (.pvk file which contains the private key for the server). The private key must not have a password. If this setting does not exist, add it.
5. Look for **_BESRelay_HTTPServer_SSLCertificateFilePath** setting. If it exists, do not create a second one, but edit its value to the full path name of the .pem file which might contain both the certificate and private key for the server, or only the certificate. If this setting does not exist, add it:



Ensure that the `.pem` file is in standard OpenSSL PKCS7 .pem file format.

The certificate is supplied by the server to connecting clients and they present a dialog to the user containing information from the certificate. If the certificate meets all of the trust requirements of the connecting client, then the client connects without any interventions by the user. If the certificate does not meet the trust requirements of the client, then the user will be prompted with a dialog asking them if it is OK to proceed with the connection, and giving them access to information about the certificate. A trusted certificate is signed by a trusted authority (such as Verisign), contains the correct host name, and is not expired.

6. To require TLS 1.3, look for **_BESRelay_HTTPServer_RequireTLS13** If it exists, do not create a second one, but edit its value to `1` .

   Note: The REST API component always uses the higher and most secure version available of TLS when communicating with the BigFix server, (regardless of local settings or settings of the masthead).

7. Restart the **BES Root Server** service:
   - On Windows, open **Services**, select **BES Root Server** and on the **Action** menu, click **Restart**.
   - On Linux run from the prompt: `systemctl restart besserver`

     Up to BigFix Version 11.0.1, for backward compatibility reasons, you can also run the command:

     ```
     /etc/init.d/besserver restart
     ```

8. To restore the connection between the BES Root Server and Web Reports, from Web Reports edit the datasource settings for the datasource whose certificate was modified as follows:
   a. Select **Administration > Datasource Settings > Edit**.
   b. Enter the password in the appropriate field and submit the form to exchange the certificate and accept the request warning.

Note: These settings are stored in the registry under the key **HKLM/Software/WoW6432Node/BigFix/EnterpriseClient/Settings/Client**

Customizing HTTPS manually

If you have a trusted SSL security and a key from a certificate authority (`.pem` file), you can configure the computer running REST API (usually the server) to customize trusted connections.

On Windows systems

To customize HTTPS manually on Windows systems, complete the following steps:

1. Run **regedit** and locate `HKEY_LOCAL_MACHINE\Software\Wow6432Node\BigFix\EnterpriseClient\Settings\Client`

   You need to add or modify subkeys for the HTTPS flag, and for the location of the SSL certificate.
2. Create a subkey of **Client** called `_BESRelay_HTTPServer_UseSSLFlag` (if it does not exist yet). Add a string value (reg_sz) called "value" to the key and set it to 1 to enable HTTPS.
3. **Important: If you combined the private key file with the certificate file, move to step 4.**

   Create a subkey of **Client** called `_BESRelay_HTTPServer_SSLPrivateKeyFilePath` (if it does not exist yet). Add a string value (reg_sz) called "value" to the key and set it to the full path name of the private key (.pvk file which contains the private key for the server).
4. Create a subkey of Client called `_BESRelay_HTTPServer_SSLCertificateFilePath` (if it does not exist yet). Add a string value (reg_sz) called "value" to the key and set it to the full path name of the SSL certificate (cert.pem).
5. **To require TLS 1.3:** Create a subkey of Client called `_BESRelay_HTTPServer_RequireTLS13` (if it does not exist yet). Add a string value (reg_sz) called "value" to the key and set it to 1 to enable TLS 1.3.
6. Restart the `BES Root Server` service.

7. To restore the connection between the BES Root Server and Web Reports, from Web Reports edit the datasource settings for the datasource whose certificate was modified as follows:
    a. Select **Administration > Datasource Settings > Edit**.
    b. Enter the password in the appropriate field and submit the form to exchange the certificate and accept the request warning.

On Linux systems

To customize HTTPS manually on Linux systems, complete the following steps:

Save the files `cert.pem` and `pvtkey.pvk` (if you have it) in a protected area of the file system, where it can be accessed by the BigFix besserver process, for example, `/etc/opt/BESServer/`.

Edit the `/var/opt/BESServer/besserver.config` file, by adding the following entries.

Important: If you combined the private key file with the certificate file, skip these settings.

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

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

To enable HTTPS:

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

To require TLS 1.3:

```
[Software\BigFix\EnterpriseClient\Settings\Client\_BESRelay_HTTPServer_RequireTLS13]
value = 1
```

Stop and restart the BigFix root server.

To restore the connection between the BES Root Server and Web Reports:

From Web Reports, edit the datasource settings for the datasource whose certificate was modified as follows:

1. Select **Administration > Datasource Settings > Edit**.
2. Enter the password in the appropriate field and submit the form to exchange the certificate and accept the request warning.

Chapter 11. Configuring bearer token authentication

Starting with **BigFix Platform v11.0.6**, operators can authenticate to BigFix services (BigFix Server REST APIs, BigFix Explorer REST APIs and the IEM CLI) using personal bearer tokens.

Overview

Bearer tokens offer a secure alternative to basic authentication (username/password) because they allow for automated tasks without exposing primary credentials. These follow the industry-standard bearer token scheme (RFC 6750).

Prerequisites and Permissions

Below is a list of relevant prerequisites and permissions required to use this feature:

- **Administrative Oversight**: A token for an operator can only be created by that operator; Master Operators can edit or delete existing tokens of any operator via REST API; Non-Master Operators can only manage their own.
- **Privileges**: To authenticate using a token, you must be a Master Operator or a Non-Master Operator with "Can use REST API" login privilege.
- **Inheritance**: Authentication via token benefits from the same permissions that the operator has when using basic authentication.

Enabling or Disabling the Feature

To issue and validate authentication tokens, BigFix needs to use a dedicated secret key, which is absent by default. To enable the bearer token authentication feature, you must first create this key.

You can create, rotate, or revoke the token secret key using the BigFix Administration Tool.

- Creating the token key: Use the `createtokenkey` service of the BigFix Administration Tool to create the token key and enable the feature.
- Rotating the token key: Use the `rotatetokenkey` service of the BigFix Administration Tool to rotate the existing token key. Please note that this operation permanently

invalidates all existing tokens for all operators, so you will need to create new ones to continue using the feature.

- Revoking the token key: Use the `revoketokenkey` service of the BigFix Administration Tool to revoke the existing token key. Please note that in addition to disabling the feature, this operation permanently invalidates all existing tokens for all operators, so if the feature is later re-enabled, you will need to create new ones to use it.

For more information about using these commands, see BESAdmin Windows Command Line and BESAdmin Linux Command Line.

Creating a Token

Important Security Note: BigFix does not store tokens in clear text anywhere.

Using the BigFix Console:

1. From the **Tools** menu, navigate to the **Manage Tokens...** window.
2. Click **Add New...**.
3. Enter a **Name** (this will be known and visible only to the operator who creates the token).
4. Set the **Duration** in days. Check "Non-expiring token" for long-term tokens.
5. Click **OK**. A pop-up will display your token. Copy and store it securely. **Important Note:** The token can only be viewed and copied at this time. If you lose it, you will need to create a new one.

For more details, see Manage Tokens.

Using the BigFix Server REST APIs:

- **Endpoint**: POST api/token
- **Parameters**: name (string), duration (integer indicating the token validity in days; 0 for non-expiring tokens; default is 30 days if parameter is not specified).

For more details, see Token REST API.

Modifying and Deleting Tokens

You can view token metadata (Id, Name, Expiration, User) but never the token string itself after creation.

Action	Console Method	REST API Method
List	Open the "Manage Tokens..." window from the "Tools" menu.	GET api/tokens; GET api/token/name/{token_name}; GET api/token/id/{token_id}; GET api/tokens/user/{operatorID}
Edit	On the "Manage Tokens..." window, click "Edit..." on a specific token row.	PUT api/token/{token_id}
Delete	On the "Manage Tokens..." window, click "Delete" on a specific token row.	DELETE api/token/{token_id}

Using Tokens

Once you have created a token, use it across the following interfaces:

BigFix Server REST APIs and BigFix Explorer REST APIs

Include the token in your HTTP request header using the **Bearer** authentication scheme:

```
Authorization: Bearer <your_token_here>
```

For more details, see Token REST API and Administering the BigFix Explorer.

IEM Command Line (IEM CLI)

Log in using the `--token` parameter:

```
./iem login --server={ServerName}:{ServerPort} --token={your_token_here}
```

For more details, see Token Authentication.

Chapter 12. Real Time AV Exclusions

BigFix Console, Server and Relay components of the architecture perform high volume file operations. This activity is a substantial part of the functionality that these BigFix architecture components provide.

If file operations are interrupted or "shimmed" by anti-virus or heuristic type applications (like HIPS), the performance of these components will be significantly impacted. Sometimes, this can result in errors and instability. The BigFix Client also is continuously evaluating the machine and this also creates a large volume of API, registry and file operations. The client is also negatively impacted by the same concerns and as a result can experience significantly slower content evaluation times.

To address this issue, configure Anti-virus and heuristic applications (such as HIPS) to exclude the following directories and processes. It is important to note the specifications below are related to the exclusion of folders paths and processes for real-time scans and heuristics, we do still recommend scheduled scans be configured and enabled from a security perspective.

Important Caveats

The following applies to BigFix platform core components only and excludes solutions such as BigFix Inventory, ILMT or OSD (which may have their own guidance around AV exceptions). This also assumes that you are using the default installation paths, otherwise you might need to adjust appropriately to the configurations of your environment.

For more details on the AV Exclusions, see AV Exclusions on Windows (on page 158), AV Exclusions on Linux (on page 160) and AV Exclusions on MacOS (on page 163).

Refer to instructions from your virus scanner for more information on how to set this exclusion rule.

For more details, see the technote Configuring your virus scanner to exclude the BigFix client and the BigFix Inventory Scanners.

In case the BigFix Client leverages Self Service Application, refer to AV Exclusions on Windows & MacOS.

AV Exclusions on Windows

How to apply the AV exclusion on Windows OS for the BigFix Platform core components.

Note: The default value for <*installation path*> is `C:\Program Files (x86)\BigFix Enterprise`.

- **On the BigFix Server**

  The following folder and sub folder paths should be excluded:

  <installation path>\BES Server*

  C:\Windows\Temp\big*.tmp

  Additionally the following processes should be excluded as well:

  <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

- **On the BigFix Relay**

  The following folder and sub folder paths should be excluded:

  <installation path>\BES Relay*

  Additionally the following processes should be excluded as well:

  <installation path>\BES Relay\BESRelay.exe

- **On the BigFix Client**

  The following folder and sub folder paths should be excluded:

  <installation path>\BES Client*

  Additionally the following processes should be excluded as well:

  <installation path>\BES Client\BESClient.exe

<installation path>\BES Client\BESClientUI.exe

Optionally the following process should also be excluded if the following component is installed within the BES Client directory:

<installation path>\BES Client\BESClientHelper.exe

Optionally the following process should also be excluded if leveraging the QNA component within the BES Client directory:

<installation path>\BES Client\qna.exe

- **On the BigFix Console**

  The following folder and sub folder paths should be excluded: this primary AV exception for the console relates to the console cache directory. This directory by default is located within the users profile path. For example:

  %LOCALAPPDATA%\BigFix*

  The user BigFix Console cache location is configurable as well via a registry setting (this may make it easier to apply AV exclusions in some AV and heuristics products). More information on this configuration can be found in Altering BigFix Console cache location.

  Additionally the following processes and files should be excluded as well:

  <installation path>\BES Console\BESConsole.exe

  %LOCALAPPDATA%\Temp*\bigfix*.tmp

  %LOCALAPPDATA%\Temp\bigfix*.tmp

  Optionally the following directory should also be excluded if leveraging the QNA component within the BigFix Console directory:

  <installation path>\BES Console\QNA*

  Additionally, the following processes:

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

- **On the BigFix WebUI Server**

  The following folder and sub folder paths should be excluded:

<installation path>\BES WebUI*

Additionally the following processes should be excluded:

<installation path>\BES WebUI\WebUIService.exe

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

- **On the BigFix Plugin Portal**

  The following folder and sub folder paths should be excluded:

  <installation path>\BES Plugin Portal*

  Additionally the following processes should be excluded:

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

- **On the BigFix Explorer**

  The following folder and sub folder paths should be excluded:

  <installation path>\BES Explorer*

  Additionally the following processes should be excluded:

  <installation path>\BES Explorer\BESExplorer.exe

AV Exclusions on Linux

How to apply the AV exclusion on Linux OS for the BigFix Platform core components.

- **On the BigFix Server**

  The following folder and sub folder paths should be excluded:

  /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

Additionally the following processes should be excluded as well:

/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

- **On the BigFix Relay**

  The following folder and sub folder paths should be excluded:

  /opt/BESRelay/

  /var/opt/BESRelay/

  /var/log/

  /etc/init.d/

  /usr/lib/systemd/system

  Additionally the following processes should be excluded as well:

  /opt/BESRelay/bin/BESRelay

- **On the BigFix Client**

The following folder and sub folder paths should be excluded:

/opt/BESClient/

/var/opt/BESClient/

/var/opt/BESCommon/

/etc/opt/BESClient/

/etc/init.d/

/usr/lib/systemd/system

Additionally the following processes should be excluded as well:

/opt/BESClient/bin/BESClient

/opt/BESClient/bin/qna

/opt/BESClient/bin/XBESClientUI

/opt/BESClient/bin/XOpenUI

/opt/BESClient/bin/xqna

- **On the BigFix WebUI Server**

  The following folder and sub folder paths should be excluded:

  /opt/BESWebUI/

  /var/opt/BESWebUI/

  /etc/init.d/

  /usr/lib/systemd/system

  Additionally the following processes should be excluded as well:

  /opt/BESWebUI/bin/BESWebUI

  /var/opt/BESWebUI/node

- **On the BigFix Portal**

  The following folder and sub folder paths should be excluded:

  /opt/BESPluginPortal/

/var/opt/BESPluginPortal/

/var/log/

/etc/init.d/

/usr/lib/systemd/system

Additionally the following processes should be excluded as well:

/opt/BESPluginPortal/bin/BESPluginPortal

- **On the BigFix Explorer**

  The following folder and sub folder paths should be excluded:

  /opt/BESExplorer/

  /var/opt/BESExplorer/

  /var/log/

  /usr/lib/systemd/system

  Additionally the following processes should be excluded:

  /opt/BESExplorer/bin/BESExplorer

AV Exclusions on MacOS

How to apply the AV exclusion on MacOS for the BigFix Platform core components.

On the BigFix Client

The following folder and sub folder paths should be excluded:

/Library/Application Support/Bigfix/Bes Agent

/Library/BESAgent

Additionally the following processes should be excluded as well:

/Library/BESAgent/BESAgent.app/Contents/MacOS/BESAgent

/Library/BESAgent/BESAgent.app/Contents/MacOS/QnA

/Library/BESAgent/BESAgent.app/Contents/MacOS/BESClientUI.app/Contents/MacOS/BESClientUI

Chapter 13. Downloading files in air-gapped environments

In air-gapped environments, to download and transfer files to the main BigFix server, use the Airgap utility and the BES Download Cacher utility.

Overview

In an air-gapped environment where a secure network is physically isolated from insecure networks, such as the public Internet or an insecure local area network, and the computers on opposite sides of the air gap cannot communicate, to download and transfer files to the main BigFix server, you can use the Airgap utility and the BES Download Cacher utility.

Note: The Airgap utility does not support a configuration where the clients are air-gapped separately from the main BigFix server. The clients must be air-gapped together with the main BigFix server to be able to gather across the network from the main BigFix server.

Starting from BigFix Version 9.5.5, you have two different modes to work in an air-gapped environment. The "Extraction usage" mode, that was already available before Version 9.5.5, and the new "Non-extraction usage" mode.

Note: To avoid license issues, ensure that the Airgap tool major version you are using matches the BigFix server major version.

Non-extraction usage overview

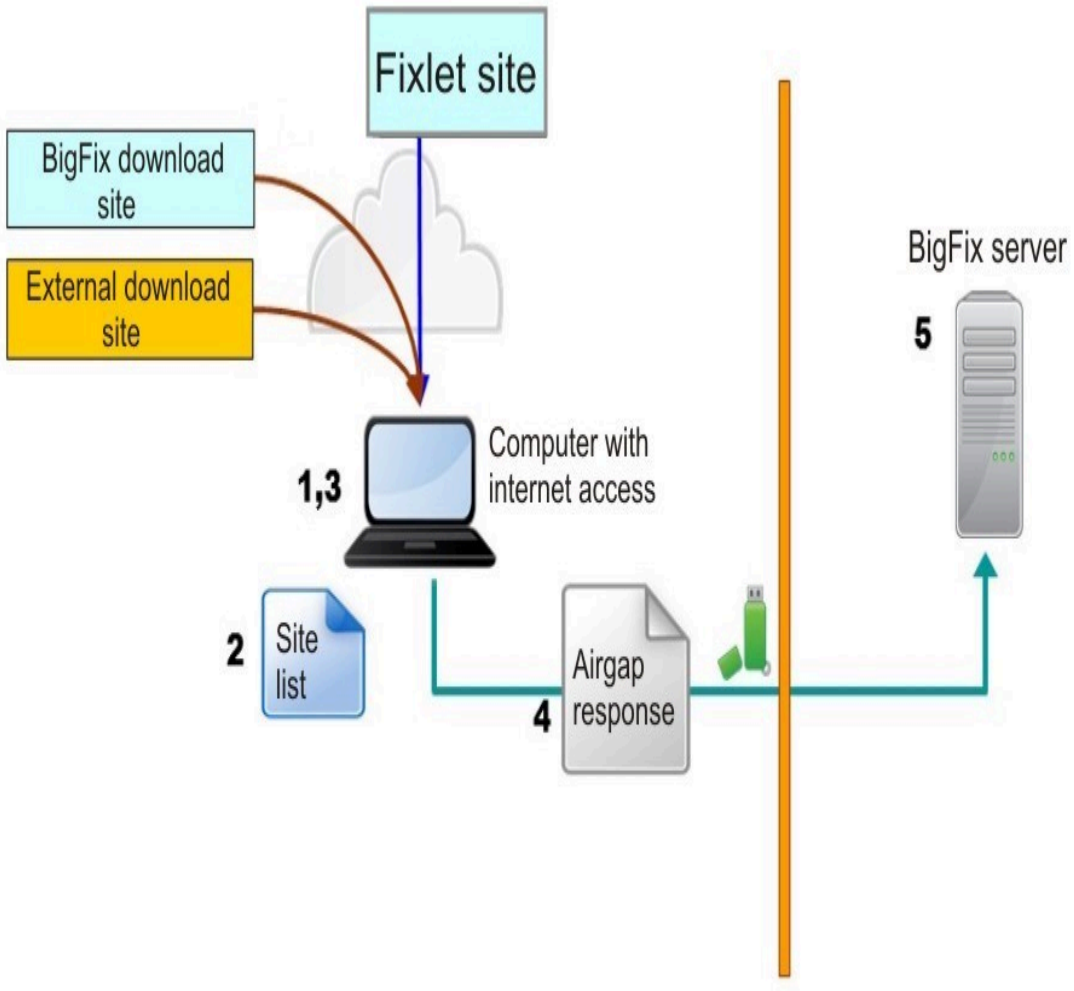
The "Non-extraction usage" mode is available only starting from BigFix Version 9.5.5.

Airgap might need to work without extracting any information from the BigFix server because in some places a rule forbids to extract any information in a secure network and move to an external network, such as Internet. To satisfy these requirements, the Airgap tool can now work without creating any Airgap request.

You can use the Airgap tool in three different ways:

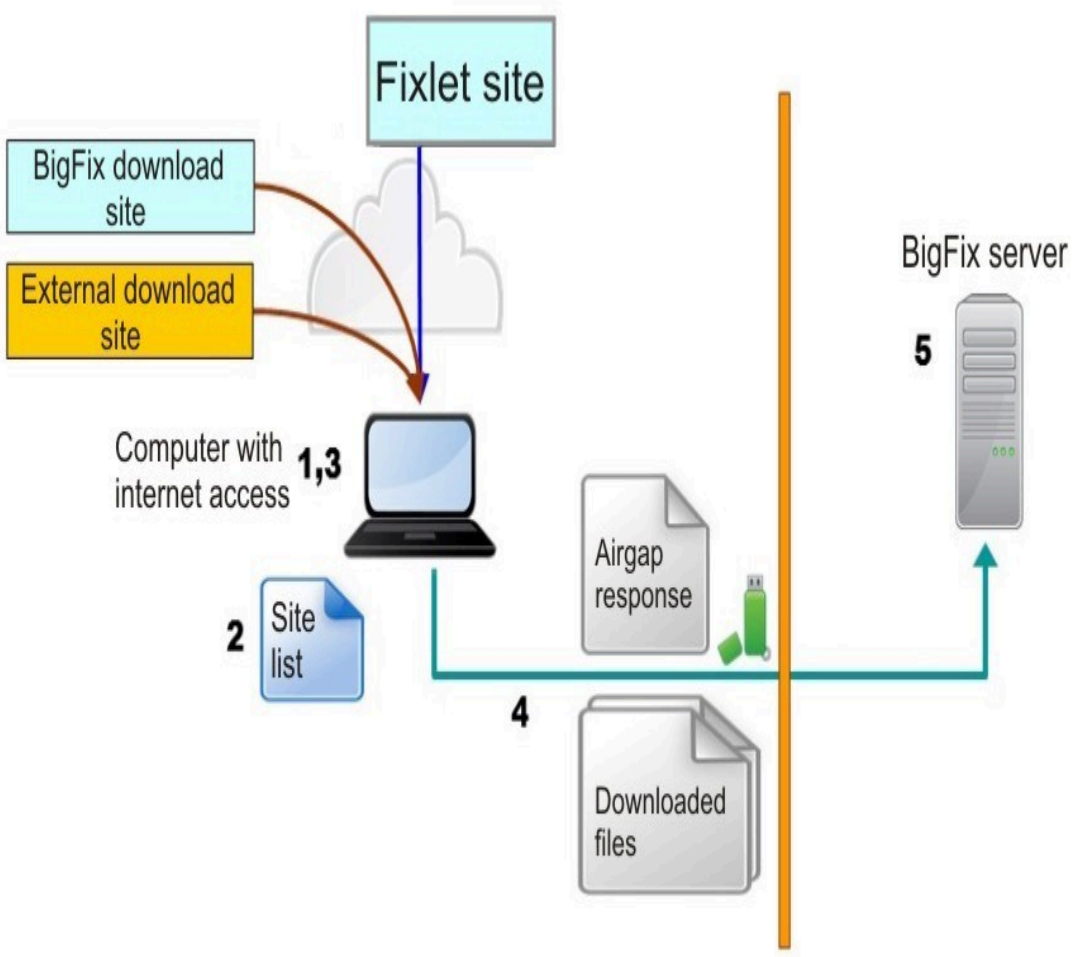
Gather site contents

1. Run the Airgap tool on the internet facing computer to gather license information and create a site list file, which contains information related to the sites that you have licensed.
2. Edit the site list file and change the flags to specify the sites that you want to gather contents from.
3. Run the Airgap tool on the internet facing computer to gather license information and site contents as specified by the site list file into the Airgap response.
4. Move the Airgap response to the BigFix server.
5. Run the Airgap tool on the BigFix server to load the Airgap response into the BigFix server.



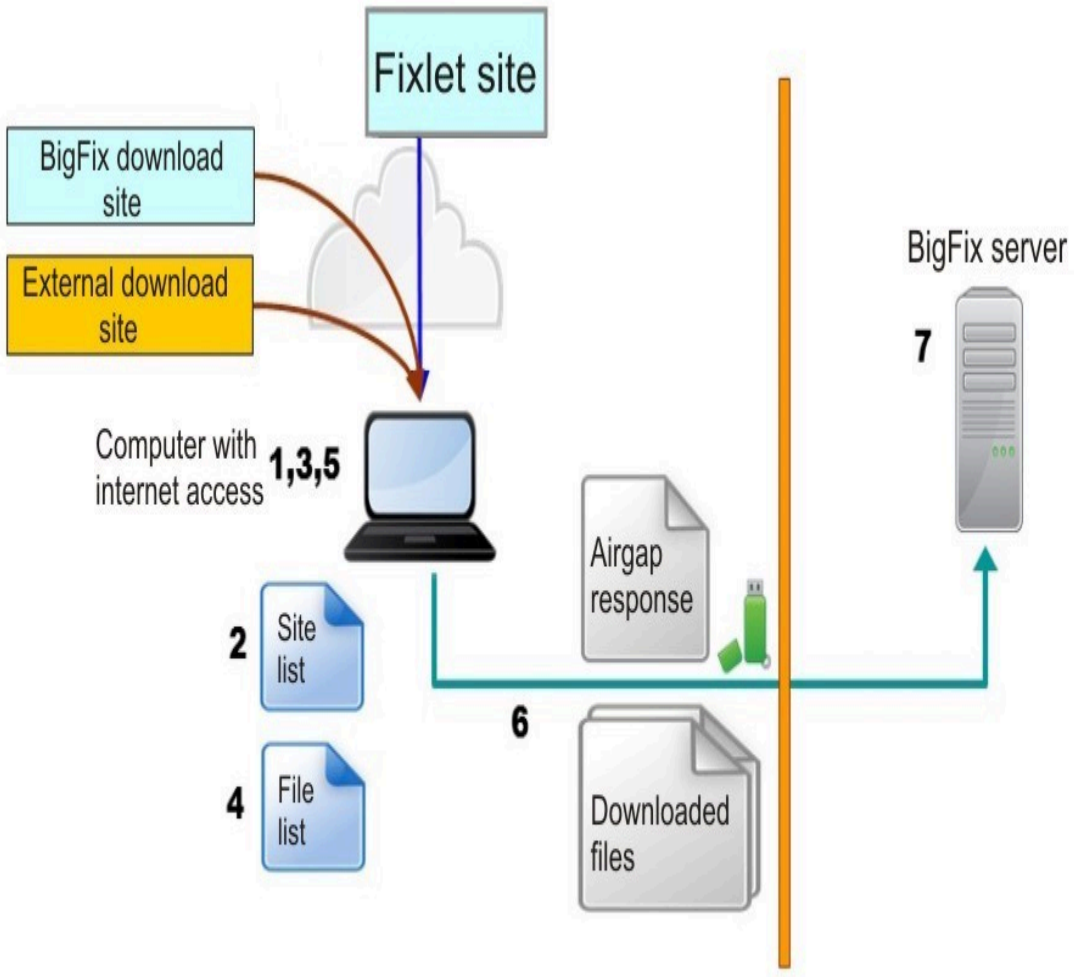
Gather site contents and download files

1. Run the Airgap tool on the internet facing computer to gather license information and create a site list file, which contains information related to the sites that you have licensed.
2. Edit the site list file and change the flags to specify the sites that you want to gather contents from, and the sites from which you want to download referenced files.
3. Run the Airgap tool on the internet facing computer to gather license information and site contents as specified by the site list file into the Airgap response, and then download the files referenced by the Fixlets.
4. Move the Airgap response and the downloaded files to the BigFix server.
5. Run the Airgap tool on the BigFix server to load the Airgap response into the BigFix server, and copy the downloaded files to the cache folder of the BigFix server.



Gather site contents and download files selectively

1. Run the Airgap tool on the internet facing computer to gather license information and create a site list file, which contains information related to the sites that you have licensed.
2. Edit the site list file and change the flags to specify the sites that you want to gather contents from, and sites from which you want to download referenced files.
3. Run the Airgap tool on the internet facing computer to gather license information and site contents as specified by the site list file into the Airgap response, and then create a file list file, which contains information about the referenced files.
4. Edit the file list file to specify the files that you want to download.
5. Run the Airgap tool on the internet facing computer to download the files as specified by the file list file.
6. Move the Airgap response and the downloaded files to the BigFix server.
7. Run the Airgap tool on the BigFix server to load the Airgap response into the BigFix server, and copy the downloaded files to the cache folder of the BigFix server.

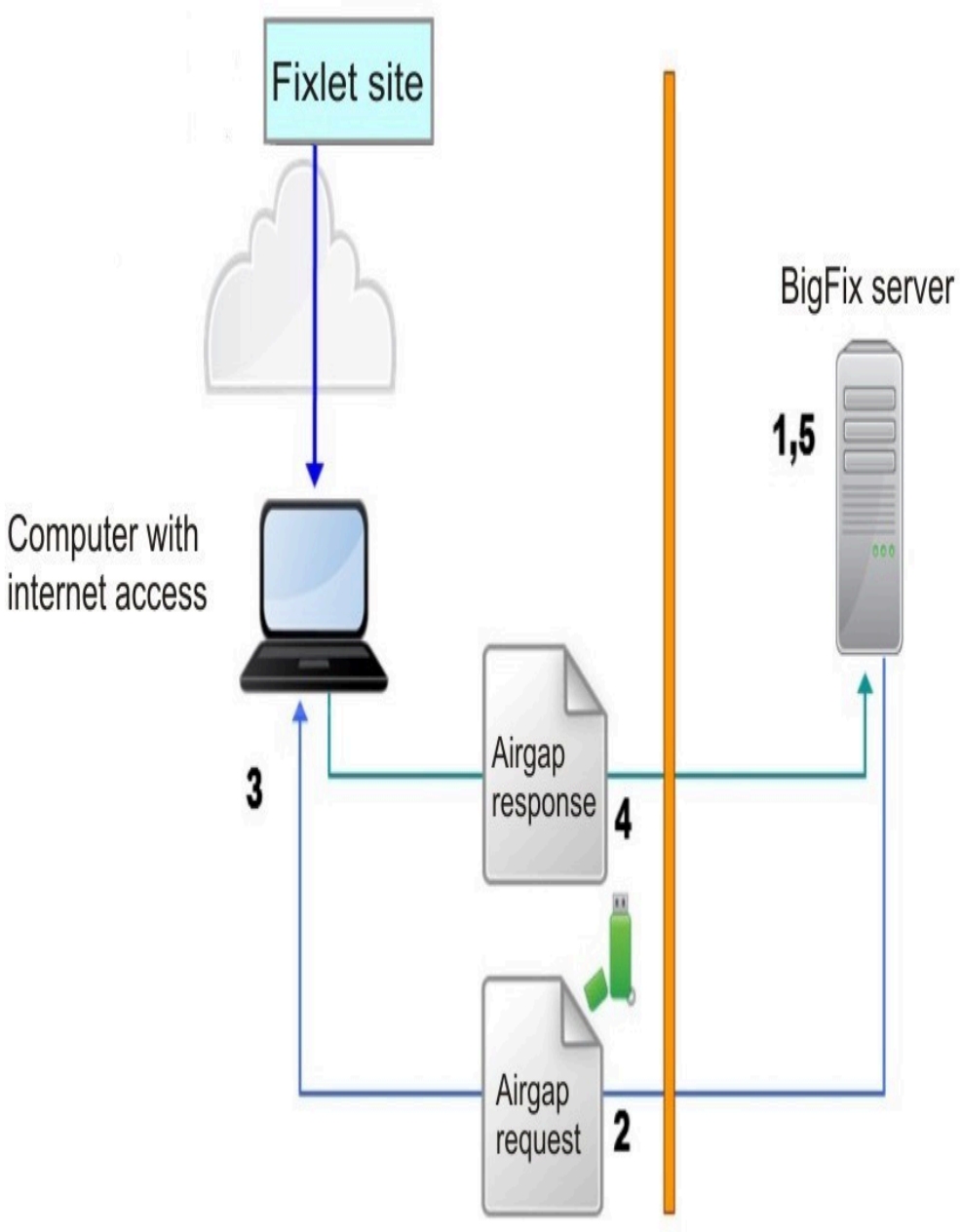

Fixlet site
BigFix download site
External download site
BigFix server
7
Computer with internet access
1,3,5
Airgap response
2
Site list
6
4
File list
Downloaded files

Extraction usage overview

In this mode, the Airgap tool extracts information from the BigFix server.

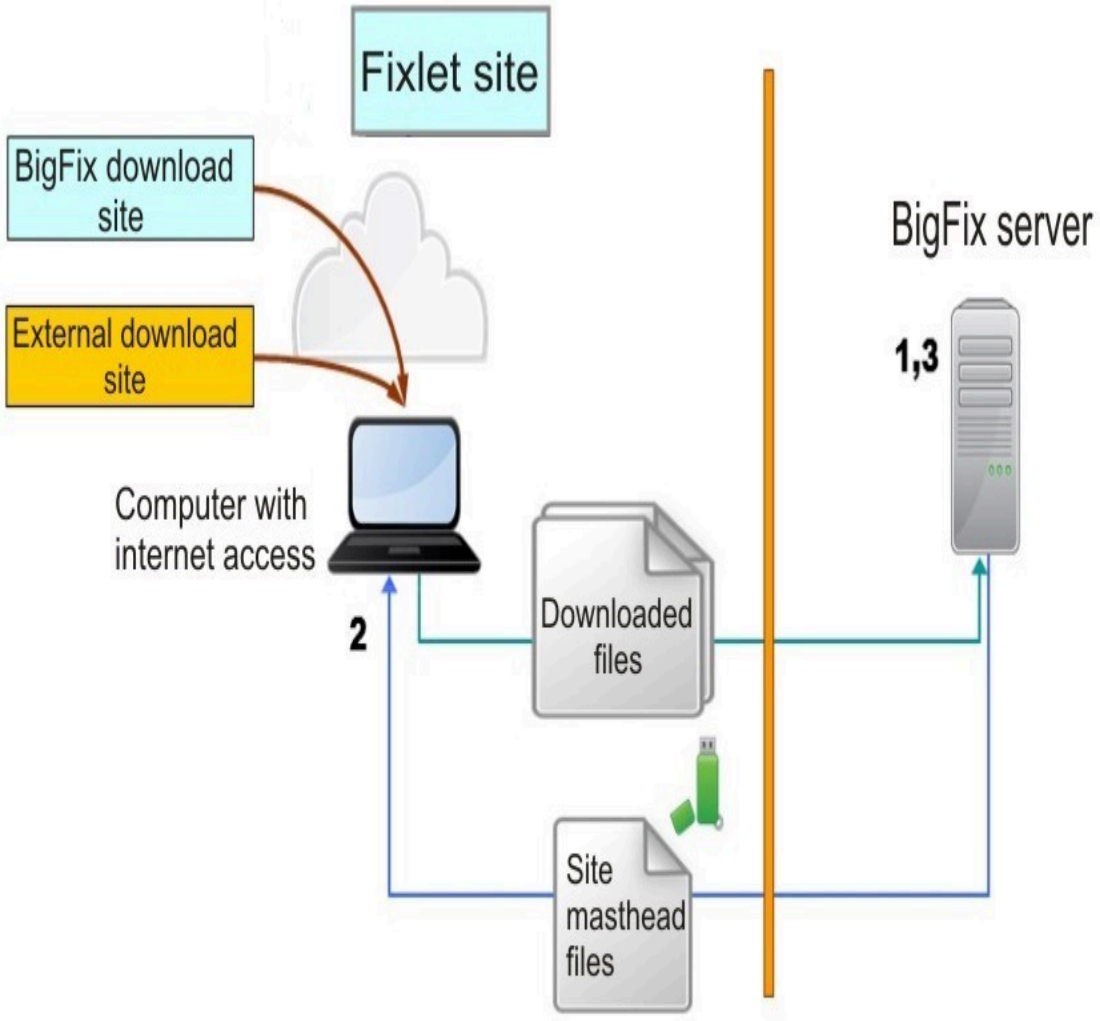
You run the Airgap tool starting from the BigFix server by performing the following steps:

1. Run the Airgap tool on the BigFix server to create the Airgap request.
2. Move the Airgap request to the internet facing computer.
3. Run the Airgap tool on the internet facing computer to gather the license information and the site contents into the Airgap response.
4. Move the Airgap response to the BigFix server.
5. Run the Airgap tool on the BigFix server to load the Airgap response into the BigFix server.


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

In this mode, Airgap gathers the contents of the site, but not the files. To download the files referenced by the Fixlets, such as the patch modules, run the BES Download Cacher utility by performing the following steps:

1. Locate the site masthead files for the sites you want to download files for, and copy the site masthead files to the computer with internet access.
2. On the internet facing computer, run the BES Download Cacher utility for each site masthead file to download files referenced from the site that the site masthead file represents.
3. Move the downloaded files to the cache folder of the BigFix server.


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

Requirements

When your BigFix server is installed in an air-gapped environment where a secure network is physically isolated from insecure networks, such as the public Internet or an insecure local area network, and the computers on opposite sides of the air gap cannot communicate, you need a workstation that has access to the public Internet to download Fixlet site contents using the Airgap tool, and to download files referenced in the Fixlet action scripts.

This workstation cannot be a BigFix server or a BigFix relay.

The Airgap tool is platform dependent, but the `AirgapRequest.xml` (for extraction usage only) and `AirgapResponse` files are not. For the workstation that has access to the public Internet, you can use different operating systems available for the BigFix server.

Depending on sites gathered, the `AirgapResponse` file can be larger than 4GB. Your workstation must have enough free disk space to save the Airgap tool, the `AirgapResponse` file, and the files to download.



Note: To avoid license issues, ensure that the Airgap tool major version you are using matches the BigFix server major version.

To run the Airgap tool on Windows computers, you must have the following libraries and files installed:

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

You can get all the above files by downloading a compressed file (Airgap Tool) from the Utilities page.

To run the Airgap tool on Linux computers, you must have the following files installed:

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

If DB2 is not installed on the Linux computer that has access to the public Internet, to run the Airgap tool you must have installed the HCL Data Server Client or HCL Data Server Runtime Client using the `db2setup` command. The DB2 instance must be created with user `db2inst1`.

Using the Airgap tool

Non-extraction usage

The Airgap command line interface can gather site information without having to access the BigFix server and can optionally download files without passing through a download cacher.

With the non-extraction usage, the Airgap tool can download the files specified in Fixlets from download sites like Windows that do not require to authenticate. When you need to download files from sites that require to authenticate with an userid and password, or to download files not specified by prefetch or download commands in Fixlets, as in the case of patch modules for AIX, CentOS, HP-UX, RedHat, Solaris or SUSE, you must use a download cacher.

As a prerequisite for the following procedure, ensure that you have the files required for the Airgap tool to run.

On Windows

You can download the appropriate Airgap tool version from the Support page.

On Linux

Starting from BigFix Version 10.0.2, you must install the package named `unixODBC.x86_64`. The same package version installed on the BigFix Server

must also be installed in the workstation connected to the Internet, where you are running the NON-Extraction procedure for Airgap environments.

Access the BigFix server computer, open the `/opt/BESServer/bin` folder and run this command:

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

Where *directory* is a folder of your choice.

Move to the directory containing the output generated by the above command, locate the file named `airgap.tar` and decompress it. Delete the `AirgapRequest.xml` file from the directory, copy all the other files portable drive.

To gather site information without accessing the BigFix server, complete the following steps:

1. **Create a site list**

   Run the tool on a workstation that has access to the public Internet specifying the license serial number, the email address used to register your license, and the name of the file in which the tool lists the sites for your license. You must have writing access for the folder where the Airgap tool is located. Enter the following command:

   On Windows operating systems:

   ```
   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]
   ```

   On Linux operating systems:

```
./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]
```

Where:

mail_address

Is the mail address that you specified in your license; if it does not match, the Airgap tool fails. Option `-email` can be used only together with option `-createSiteList`.

-proxy

Option used when the workstation that has access to the public Internet can connect only by a proxy server. In this case, after the `-proxy` option, specify the hostname and port of the proxy server in the form *hostname:port*. If the proxy is an authenticating proxy, add also the userid and password in the form *userid:password@hostname:port*.

-usehttps

When this option is specified, "https" is used to contact the license server. Use option `-cacert` to specify a path in which to put the file `ca-bundle.crt` if you want to use a different folder from that in which the Airgap tool runs. The file `ca-bundle.crt` is used to validate the server certificate when you use the `-usehttps` option, or when the URL in the Fixlet begins with "https".

-cacert

This option can only be used together with option `-usehttps`.

-othersites

Use this option if your license is entitled to AllowOtherSites, to include sites of your choice to your site list. Create a folder, copy in it all the masthead files (*.efxm files) related to your mastheads not included in your license, and specify the name of this folder with option `-othersites` when you create a site list.

-timeout

This option is available starting from Version 9.5.7. It specifies a http timeout interval in seconds. Values range from 30 to 3600. The default value is 30. In the event you get the error "HTTP Error 28: Timeout was reached" while using a proxy, try also to use option `-usehttps` as it makes proxy to work in tunneling mode and that might help avoiding timeouts.

After running the tool, a file is created with the name that you specified as *`site_list_filename`*.

Note: The site list file, once created, can be used until you change the license, or HCL adds a new site to the existing license. If you delete the site list file for any reason, you can create it again with the same command, as the history of downloaded files is maintained as long as the license serial number does not change.

2. Edit the site list file

Each line of the file created in step 1 contains three pieces of information separated by a double colon:

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

You can edit only the *`flag`* parameter, that can have one the following values:

A

Site contents are gathered when a newer site version is available and stored in the `AirgapResponse` file, and used for downloading files or creating a file list.

R

Site contents are always gathered and stored in the `AirgapResponse` file regardless of the version of the site, and used for downloading files.

G

Site contents are gathered when a newer site version is available and stored in the `AirgapResponse` file, but not used for downloading files or creating a file list.

Q

Site contents are always gathered and stored in the `AirgapResponse` file regardless of the version of the site, but not used for downloading files or creating a file list.

D

Site contents are not gathered, but are used for downloading files or creating a file list. This flag is useful when you want to keep the current contents of a site without updating it and download files to run Fixlets at your current site. This option is valid only when the site contents have already been gathered.

N

Site is ignored, but site information is kept in the file for future reference.

Note: When you create a site list file, the default values for the BES Support and Web UI Common components are set to G. If you are not interested in the Web UI component, modify the default Web UI Common value from G to N. The default values for the

other components are set to N. At the first run after installing the BigFix server, the license information, the BES Support and the Web UI Common components must be gathered. Only after moving this first Airgap response generated on the workstation that has access to the public Internet to the BigFix server, you can enable the other components that you can access from the License Overview dashboard of the console and continue with the process. Be sure to enable the required components other than default before gathering.

3. Gather site contents and create the Airgap response file

After you have edited the flags in the site list file, run the Airgap tool again to complete one of the following site operations:

a. Gather site contents

To gather site contents for sites with flag **A** or **R** or **G** or **Q**, run the following command:

On Windows operating systems:

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

On Linux operating systems:

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

On completion, you have created the `Airgapresponse` file.

b. Gather site contents and download files

To gather site contents for sites with flag **A** or **R** or **G** or **Q**, and download files referenced by Fixlets on sites with flag **A** or **R** or **D**, run the following command:

On Windows operating systems:

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

On Linux operating systems:

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

where *cache_name* is the folder path where to store the downloaded files. On completion, you have created the `Airgapresponse` file and downloaded the files to the *cache_name* folder.

c. Gather site contents and download files selectively

To gather site contents for sites with flag **A** or **R** or **G** or **Q**, and create a list of files referenced by Fixlets on sites with flag **A** or **R** or **D**, run the following command:

On Windows operating systems:

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

On Linux operating systems:

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

On completion, you have created the `Airgapresponse` file and the file list with the name specified in *referenced_list*.

In all cases, site contents gathered for sites with flag **A** or **R** or **G** or **Q** are put in the `AirgapResponse` file. When you run the Airgap tool for the first time, all sites with flag **A** or **R** or **G** or **Q** are gathered. For subsequent times, the

contents of sites with flag **A** or **G** are gathered only if either they have not been previously gathered or a newer site version is available. For sites with flag **R** or **Q**, contents are always gathered.

Optionally, you can also specify the following options:

`-usehttps`

License information and site contents are gathered using "https". For case "b. Gather site contents and download files", all urls beginning with "http" are forced to use "https". Note that some urls in Fixlets begin with "https" and some patch sites might redirect requests to urls beginning with "https".

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

Used when the workstation that has access to the public Internet can connect only through a proxy server. In this case, after the `-proxy` option, specify the host name and port of the proxy server in the format *hostname:port*. If the proxy is an authenticating proxy, add also the user ID and password in the format *userid:password@hostname:port*.

`-cacert crt_filename`

To specify a path in which to put the file `ca-bundle.crt` if you want to use a different folder from that in which the Airgap tool runs. The file `ca-bundle.crt` is used to validate the server certificate when you use the `-usehttps` option, or when the url in the Fixlet begins with "https". The option `-cacert` can only be used together with option `-usehttps`.

`-timeout timeout_seconds`

This option is available starting from V9.5.7. It specifies a http timeout interval in seconds. Values range from 30 to 3600. The default value is 30. In the event you get the error "HTTP Error 28: Timeout was reached" while using a proxy, try also to use option

`-usehttps` as it makes proxy to work in tunneling mode and that might help avoiding timeouts.

For cases **b** and **c**, you can also use other options to reduce the number of files to download or to gather in the file list. These filtering options select Fixlets that refer to files, not the files themselves. For example, when you specify last 5 days, it means files referenced by Fixlets modified in the last 5 days, not files added or changed by vendors in the last 5 days. To create a list of possible values for filtering options, run the following command:

On Windows operating systems:

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

On Linux operating systems:

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

The list of available values is limited to the following options: `-fcategory`, `-fcve`, `-fproduct`, `-fseverity`, `-fsource`, and `-fsourceid`. The following options are available for filtering:

`-fcategory`

Fixlet category property.

`-fcve`

To specify the CVE (Common Vulnerabilities and Exposures) id associated with a security patch.

`-fdays`

To select Fixlets whose last modified date falls within a specified number of days from the date you run the command.

`-fproduct`

To specify the product name to which the Fixlet is applicable, such as `Win2008` or `Win7`. This information is not shown in the Console. This option is available only for sites related to patches for Windows operating systems.

`-fseverity`

To specify the severity that a vendor associates with a security patch.

`-fsource`

Provider of file, such as BigFix, Adobe, or Microsoft.

`-fsourceid`

Identification specified by the provider.

`-includeCorrupt`

To include Fixlets marked as Corrupted, that are excluded by default when this option is not specified.

`-includeSuperseded`

To include Fixlets marked as Superseded, that are excluded by default when this option is not specified.

When multiple filter conditions are specified, only Fixlets that satisfy all conditions are selected. For options `-fsource`, `-fsourceid`, `-fcve`, `-fcategory`, and `-fseverity`, you can specify multiple comma-separated values, for example: `-fseverity "Critical, Important"`. When you use commas to separate values, or values contain spaces, enclose parameters in double quotes, as in the previous example. Note that values are case sensitive.

4. Edit the file list

Applicable only to case **c. Gather site contents and download files selectively** of step 3.

With `-createFileList` option, you create a file that contains a list of files. Each line of the list contains pieces of information separated by a double colon:

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

For example:

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

You can edit only the *`flag`* value, changing it to **Y** to download the file, or to **N** to not download the file.

5. Run the tool on the Internet facing workstation to download files

Applicable only to case **c. Gather site contents and download files selectively** of step 3.
After editing the file list in step 4, to download only the files with flag **Y** in the file list, run the Airgap tool by issuing the following command:

On Windows operating systems:

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

On Linux operating systems:

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

where *cache_foldername* is the folder path where to store the downloaded files. The files already in the cache folder are not downloaded again.

6. Move the Airgap response file to the BigFix server and run the Airgap tool on the BigFix server

Copy in a portable drive the `AirgapResponse` file, and the file list that you have created in step 3 or the downloaded files that you collected in step 5, and transfer them to the BigFix server computer. Make sure that the `AirgapResponse` file is in the same folder as the Airgap tool, and run it by issuing the following command:

On Windows operating systems:

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

On Linux operating systems:

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

This imports the response file with the Fixlet content and license updates into your deployment.

Note: The Airgap tool passes site contents in the response file to the GatherDB component of your BigFix server, and the GatherDB component imports site contents. For sites other than WebUI sites, you can monitor the import progress in the DebugOut of the GatherDB component (default name `GatherDB.log`).

Copy the downloaded files also into the BigFix server cache folder. The cache folder default location is:

On Windows operating systems:

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

On Linux operating systems:

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

Repeat these steps periodically to keep updated the Fixlet content in the main BigFix server. Join the new Fixlet mailing list to receive notifications when Fixlets are updated. Always make sure that the Airgap tool version is compatible with the version of the BigFix server installed.

Usage tips:

1. Unzip the exact same version of the AirgapTool used in Step 1 into a directory on the BigFix root server.
2. Copy the `airgapresponsefile` into this same directory.
3. Run `BESAirgapTool.exe` with no options.
   The contents of the `airgapresponsefile` is imported in to the directory. If you downloaded any files at Step 5, then copy those files in to the SHA1 directory on the root server as well. This might be necessary because the Airgap tool downloads files and names them with their SHA256 values.



Note: You do not need to rename the SHA256 value as its SHA1 value after pasting it to the SHA1 directory.

Optional actions:

Check if all required files have been downloaded

To check if you have downloaded all the files required for the Fixlet you are planning to apply, use option `-checkfixlet` when you run the Airgap tool. For example:

On Windows operating systems:

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

On Linux operating systems:

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

For Fixlets satisfying the specified filtering conditions, the tool checks the downloaded history and contents of destination folder, and if there are still files to download, Fixlet names and urls are displayed.

Files to be downloaded manually

Some files referenced by Fixlets might not be downloaded because they can be obtained only by contacting the vendor support center, or because the download site requires that you explicitly accept the license terms and this action cannot be automated for legal reasons. In these cases, the involved files have the download url containing the string `MANUAL_BES_CACHING_REQUIRED` and must be downloaded manually. To create a list of these files, use option `-createmanuallist` as in the following example:

On Windows operating systems:

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

On Linux operating systems:

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

You can also use the `-checkmanual` option to check if your destination folder contains all the files that must be manually downloaded, as in the following example:

On Windows operating systems:

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

On Linux operating systems:

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

Reset history

The Airgap tool keeps a history of downloaded files. Even if you move all the downloaded files from your public Internet facing workstation to the BigFix server, this history is maintained and files previously downloaded are not downloaded again to save time and disk space. If you deleted part or all of your previously downloaded files and you need them again, you can use the `-resync` option. This option clears the download history and checks the files in the folder specified with `-cache` option. Note that the newly-created download history is based only on the files contained in the folder specified with the `-cache` option.

Changing license

If you want to manage another license, you must erase the history of gathered sites and downloaded files. To complete this action, use the `-force` option as in the following example:

On Windows operating systems:

```
BESAirgapTool.exe -serial serial_number -
email
```

```
mail_addess -createSiteList site_list_fil
ename -force
```

On Linux operating systems:

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

Miscellaneous options

By default, the Airgap tool simultaneously downloads two files. You can change the number of files to download concurrently by specifying a number after the `-download` option . This number can range from 1 to 8. For example, to download 3 files at the same time, specify `-download 3`. Note that you need a larger band width when downloading more than 2 files simultaneously.

When the url specified in a Fixlet begins with "https", or if you specify the `-useHttps` option, the Airgap tool tries to verify that the server specified in the url has an appropriate SSL Server Certificate. If, for any reason, you want to skip this check and avoid a download failure when the Airgap tool cannot verify the server certificate, use the `-noverify` option. With this option, the Airgap tool does not verify the authenticity of the server certificate while it verifies that the server certificate is for the server specified in the URL you operate against. You must check that your workstation translates correctly host names by checking your DNS.

To have the Airgap tool to print more information than usual, use the `-verbose` option.

Working with multiple BigFix servers

If you want to use the same public Internet facing workstation for several BigFix servers, like a test server and a production

server, create a folder for each server, copy the Airgap tool in each folder, and work with each folder separately. You can share the same site list among the different folders, but each server keeps its own history in its folder. When using multiple Airgap tools with different servers, you can also share a cache folder to download only once files that are common to different servers, but you must ensure to run only one instance of the Airgap tool at the same time.
In case you need to gather set of sites, load them to your test server, then perform tests with the gathered sites and load the tested sites, not the latest ones, to your production server, you can load one `AirgapResponse` file to multiple BigFix servers when they are licensed for the same products (like BigFix Lifecycle, BigFix Compliance, etc.). When you intend to load one `AirgapResponse` file to multiple BigFix servers, it is recommended to gather only sites enabled on all of your BigFix servers.



Note: At the first run after installing the BigFix server, the license information, the BES Support, and the Web UI Common components must be gathered for each installation. For this step, an AirgapResponse file must be created for each BigFix server because license information is unique to each serial number.

If you want to update the license information of a particular BigFix server without changing version on any site, you can create an `AirgapResponse` file that contains only license information by running the Airgap tool with a site file containing no lines or with site files where all sites have the flag **N**. Run the following command:

On Windows operating systems:

```
BESAirgapTool.exe -site empty_site_list_f
ilename
-allowemptysite
```

On Linux operating systems:

```
./Airgap.sh -site empty_site_list_filenam
e
-allowemptysite
```

Enabling WebUI in air-gapped environments

To install the WebUI in air-gapped environments, perform the following steps:

1. Gather the latest BES Support and WebUI Common sites, and download the required files to install the WebUI Service. Load them to your BigFix server.
2. Install the WebUI Service by using the task "Install HCL BigFix WebUI Service" in BES Support site.
3. After the installation completes, wait for the activation of a WebUI Service (on Windows operating systems) or process (on Linux operating systems) on the WebUI targeting system. The WebUI initialization has started; wait for its completion. Initialization usually completes in few minutes, but it is suggested to wait 30 minutes or more before proceeding with step 4.
4. Gather all the latest WebUI sites and load them to your BigFix server. You can gather WebUI sites before running the task to install the WebUI service, but you can load them only after the WebUI initialization has completed.

Extraction usage

The "Extraction usage" mode of the Airgap tool.

Important: If you have a BigFix fresh installation, to make the WebUI sites available, you must complete the following steps:

1. Install the WebUI and run the Airgap tool.
2. Wait a few minutes for the WebUI initialization to complete.
3. Rerun the Airgap tool.

To make Fixlet content and product license updates available in the isolated network, the utility must be transferred from a computer with internet connectivity using the following steps:

On Windows operating systems

1. **Run on the BigFix server**

   From the BigFix server installation directory, double-click `BESAirgapTool.exe` or run it from the command line without any parameters, a Graphical User Interface opens.

   Provide a destination folder for the Airgap tool to store its site request and all the files it requires to run. After the Airgap tool finishes copying the files, copy the entire folder to a portable drive.
2. **Move the Airgap request and run on the internet facing computer**

   Bring the portable drive to a computer with Internet connectivity. You must have the rights to write in the folder where the `BESAirgapTool.exe` is located. Enter the folder and run the Airgap tool by double-clicking `BESAirgapTool.exe` or invoking it from the command line.

   Optionally, you can also specify the following command line parameters:

   -usehttps

   All urls beginning with "http" are forced to use "https" to gather license information and site contents. Note that some urls in Fixlets begin with "https" and some patch sites might redirect requests to urls beginning with "https".

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

This option is available only starting from BigFix Version 9.5.5. Used when the workstation that has access to the public Internet can connect only through a proxy server. In this case, after the `-proxy` option, specify the host name and the port of the proxy server in the format *hostname:port*. If the proxy is an authenticating proxy, add also the user ID and the password in the format *userid:password@hostname:port*. In extraction usage, when a proxy server is configured in the client registry settings or in the Internet Explorer settings for the current user and the `-proxy` option is not specified, the proxy settings are used as in earlier versions of the Airgap tool. When you use the `-proxy` option, the specified values are used regardless of other settings.

-cacert *<full_path_to_ca-bundle.crt_file>*

To specify a path in which to store the file `ca-bundle.crt`, if you want to use a different folder from that where the Airgap tool runs. The file `ca-bundle.crt` is used to validate the server certificate when you use the `-usehttps` option, or when the URL in the Fixlet begins with "https". The option `-cacert` can only be used together with the `-usehttps` option.

A Graphical User Interface opens. The Airgap tool will download all files required by the Airgap request in the same folder as `BESAirgapTool.exe`. This exchanges the Airgap request file for an Airgap response file. Copy the Airgap response file to a portable drive.

3. **Move the Airgap response to the BigFix server and run the Airgap tool on the BigFix server**

   Take the portable drive back to the BigFix server computer and run the `BESAirgapTool.exe` again by double-clicking `BESAirgapTool.exe` or invoking it from the command line without any parameters. Ensure that you are running it logged on as a user that:

   - Has Administrator privileges.
   - Has the database permissions necessary to add content to the BFEnterprise database.

A Graphical User Interface opens.

This imports the Airgap response file with the Fixlet content and license updates into your deployment.

The Airgap tool creates temporary files in the folder specified by the `TEMP` environment variable. If you want to use a different folder for temporary files, set the `TEMP` environment variable to that folder before you run the `BESAirgapTool.exe`.

To update the Fixlet content on the main BigFix server, repeat these steps periodically. You can join the new Fixlet mailing list to receive notifications when Fixlets are updated.

Ensure that the Airgap tool version is compatible with the installed BigFix server version.

On Linux operating systems

1. **Run on the BigFix server**
   Ensure that on the Linux computer, the Airgap tool is located in the same path where you installed the BigFix server. The default path is `/opt/BESServer/bin`. Open the Linux Terminal, and enter the following commands to create a tar file named `airgap.tar`, containing the `AirgapRequest.xml` file based on the BigFix database information:

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

   Where:

   -remotedir *directory*

   Runs Airgap to generate the request file in the specified folder.
2. **Move the Airgap request and run on the internet facing computer**
   Copy the `airgap.tar` file to a portable drive, and extract the `airgap.tar` file content by issuing the following command:

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

Ensure that your system has an environment variable named `LD_LIBRARY_PATH` set to the path of the folder containing the DB2 library `libdb2.so.1`. Ensure that the `Airgap.sh` and `AirgapRequest.xml` files are in the same folder and that you have writing rights to that folder. Run the `Airgap.sh` command.
Optionally, you can also specify the following command line parameters:

-usehttps

All urls beginning with "http" are forced to use "https" to gather license information and site contents. Note that some urls in Fixlets begin with "https" and some patch sites might redirect requests to urls beginning with "https".

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

Used when the workstation that has access to the public Internet can connect only through a proxy server. In this case, after the `-proxy` option, specify the host name and the port of the proxy server in the format *hostname:port*. If the proxy is an authenticating proxy, add also the user ID and the password in the format *userid:password@hostname:port*.

-cacert *<full_path_to_ca-bundle.crt_file>*

To specify a path in which to store the file `ca-bundle.crt`, if you want to use a different folder from that where the Airgap tool runs. The file `ca-bundle.crt` is used to validate the server certificate when you use the `-usehttps` option, or when the URL in the Fixlet begins with "https". The option `-cacert` can only be used together with the `-usehttps` option.

This exchanges the Airgap request file for an Airgap response file. Copy the Airgap response file to a portable drive.

If you receive the following error message when running the Airgap tool:

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

Create and export the `LD_LIBRARY_PATH` variable by running the command:

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

Where:

/your/path

Is the path of the folder containing the DB2 library `libdb2.so.1`

3. **Move the Airgap response to the BigFix server and run the Airgap tool on the BigFix server**
   Connect the portable drive back to the BigFix server computer and run the `Airgap.sh` command. This imports the response file with Fixlet content and license updates into your deployment.

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

Optionally, you can also specify the following option:

-temp *directory*

The Airgap tool creates temporary files under the `/tmp` directory, but in the event you do not have enough space left in it, you can use this option to specify a different folder where you have enough space.

Note that the `Airgap.sh` and `AirgapRequest.xml` files must be in the same folder.

To update the Fixlet content on the main BigFix server, repeat these steps periodically. You can join the new Fixlet mailing list to receive notifications when Fixlets are updated.

Ensure that the Airgap tool version is compatible with the installed BigFix version.

Transferring downloaded files

Deploying Fixlets on the main BigFix server requires downloaded patches and other files from the Internet. You can use the Airgap tool in extraction usage for gathering site contents and in non-extraction usage for downloading files (you can ignore the AirgapResponse file generated in non-extraction usage).

As an alternative, you can use the BES Download Cacher utility. This utility helps to:

- Download and transfer files to the main BigFix server.
- Download patch contents in a Fixlet site or single file downloads from an URL.

You can download the current utility from http://software.bigfix.com/download/bes/util/BESDownloadCacher.exe. To see the list of available options run `BESDownloadCacher.exe /?`. If the BigFix server or an BigFix relay is installed on the system where you run the BES Download Cacher utility, the `-x` utility parameter is optional because the utility detects relevant local BES settings and reuse them as defaults

Some sites require additional steps to download content from patch vendors that restrict access. For additional information see the following documentation pages that describe using a tool to manually download patches for Solaris, Red Hat,SuSE, and AIX.

These sites require a three step process:

1. Run the BESAirgapTool.exe to download Fixlets and Tasks for each site.
2. Run the BES Download Cacher utility to download any site tools from BigFix.
3. Run the download tool for each vendor to download patch contents.

Transferring all files from Fixlet sites

To transfer files from Fixlet sites, perform the following steps.

1. Locate the `.efxm` file for the site from which you want to gather downloads, for example, `BES Asset Discovery.efxm`.
2. Run the BES Download Cacher utility with the following command:

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



Note: This might take a very long time because it downloads every file referenced in the Fixlet site and puts the files into the downloads folder. If the files already exist in the downloads folder, they are not re-downloaded. Files are named with their sha1 checksum.

3. When the download finishes, copy the contents of the downloads folder (just the files, not the folder) into the sha1 folder on the main BigFix server. The default location for the sha1 folder is:
   - On Windows systems: `%PROGRAM FILES%\BigFix Enterprise\BES Server\wwwrootbes\bfmirror\downloads\sha1`
   - On Linux systems: `/var/opt/BESServer/wwwrootbes/bfmirror/downloads/sha1`

   The BigFix server uses these files instead of trying to download them from the Internet.

   Note: If you run the BES Download Cacher utility later, you can look at the modification time of the files to see which files are the newest. Using this method, you transfer only the newest files to the Main BigFix server instead of copying every file each time.

You might need to increase the size of the cache on the main BigFix server so that it does not try to delete any files from the cache. Run the BES Download Cacher utility to increase the size of the cache with the following command:

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

The default size of the cache is 1024 MB.

Note: Use the `-c` option only when the BigFix server or a relay is installed on the system where you run the BES Download Cacher utility. If no BigFix component is installed, cache has no limit.

After the files are cached in the BigFix server sha1 folder, they are automatically delivered to the BigFix relays and BigFix clients when you click an action in the Fixlet message that references a downloaded file. If the file is not cached, the BigFix console gives you a status of `Waiting for Mirror Server` after you deploy an action.

Transferring a single file

To transfer a single file from a Fixlet site, perform the following steps.

1. Run the BES Download Cacher utility with the following command:

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

2. When the download finishes, copy the contents of the downloads folder (just the file, not the folder) into the sha1 folder on the main BigFix server. The default location for the sha1 folder is:
   - On Windows systems: `%PROGRAM FILES%\BigFix Enterprise\BES Server \wwwrootbes\bfmirror\downloads\sha1`
   - On Linux systems: `/var/opt/BESServer/wwwrootbes/bfmirror/downloads/ sha1`

You might need to increase the size of the cache on the main BigFix server so that it does not try to delete any files from the cache. Run the BES Download Cacher utility to increase the size of the cache with the following command:

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

The default size of the cache is 1024 MB.



Note: Use the `-c` option only when the BigFix server or a relay is installed on the system where you run the BES Download Cacher utility. If no BigFix component is installed, cache has no limit.

After the files are cached in the BigFix server sha1 folder, they are automatically delivered to the BigFix relays and BigFix clients when you click an action in the Fixlet message that references a downloaded file. If the file is not cached, the BigFix console gives you a status of "Waiting for Mirror Server" after you deploy an action.

Log files

The Airgap tool produces two types of log files: normal log files and debug log files.

Normal log files record the messages you see in the command window so you can check Airgap tasks, such as sites gathered on a specific date. Debug log files are intended for the HCL Support team. The naming convention for normal log files is:

On Windows operating systems:

`BESAirgapTool_yyyy-mm-dd.log`

On Linux operating systems:

`Airgap_yyyy-mm-dd.log`

where *yyyy-mm-dd* is the date when the file has been created. Starting from Version 9.5.7, files older than 30 days are deleted.

The debug log file is `AirgapDebugOut.txt`. Starting from Version 9.5.7, this file contains only information of the last day and older log files are renamed to `AirgapDebugOutyyyymmdd.txt`, where yyyymmdd is the date when the file has been created; files older than 10 days are deleted. The Airgap tool can write more information to the debug log file by using the verbose option `-verbose`.

Chapter 14. Getting client information by using BigFix Query

The BigFix Query feature allows you to retrieve information and run relevance queries on client workstations from the WebUI BigFix Query Application or by using REST APIs.

Use the BigFix Query feature to:

- Quickly collect data from clients without impacting BigFix environment performance.
- Run your query in relevance language on targets identified using an applicability relevance or on a set of target agent IDs.
- Show the collected results in the WebUI Query Application, optionally paging them. The results displayed are updated periodically as new values are received from clients.
- Test relevance expressions on a few selected clients before rolling out to production.

This guide contains the information about how to configure BigFix for using BigFix Query. Additional information is available in:

- Get Started with BigFix Query
- Query *(on page 807)*

BigFix Query requirements

The clients that are targeted by BigFix Query requests must satisfy specific conditions.

The following requirements must be satisfied to run BigFix Query on clients:

- The client can receive UDP notifications. The BigFix Query feature does not support components that are connected to the BigFix server through proxies or firewalls.
- BigFix V9.5 Patch 2 or later must be installed on the client machine and on all the intermediate relays that must be passed through to reach the client.

BigFix Query restrictions

Some restrictions apply when using the BigFix Query feature.

The following limitations affect the use of the BigFix Query feature:

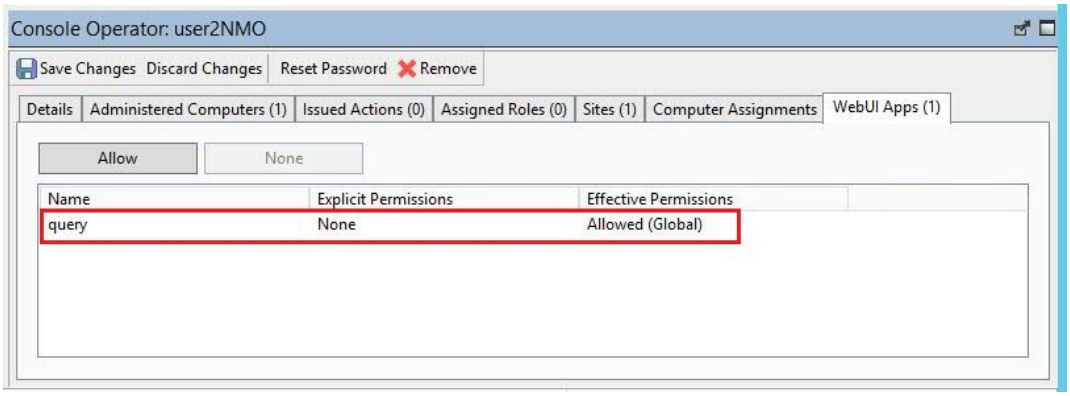
- The feature is available only for BigFix Lifecycle or BigFix Compliance Version 9.5 Patch 2 or later versions.
- Starting from Version 11 Patch 3, the feature is available also for BigFix Workspace, BigFix Workspace+, BigFix Enterprise or BigFix Enterprise+.
- Starting from Version 10 Patch 12, the feature is available also for BigFix Workspace, BigFix Workspace+, BigFix Enterprise or BigFix Enterprise+.
- Starting from Version 9.5.13, the feature supports requests that require the agent context.
- If you configured your environment in a Disaster Server Architecture (DSA), be aware that:
  - The information about BigFix Query is not replicated among the multiple servers.
  - Each server can run BigFix Query requests only on the clients that connect either directly or through relays to the server where the query is submitted.

Who can use BigFix Query

BigFix Query requests can be run by Master Operators and Non-Master Operators. Specific permissions must be set to allow operators to use this feature.

To access the WebUI Query application from the WebUI toolbar:

The user must have, at operator or role level, the effective permission on the **query** WebUI Application set to **Allowed**, for example:

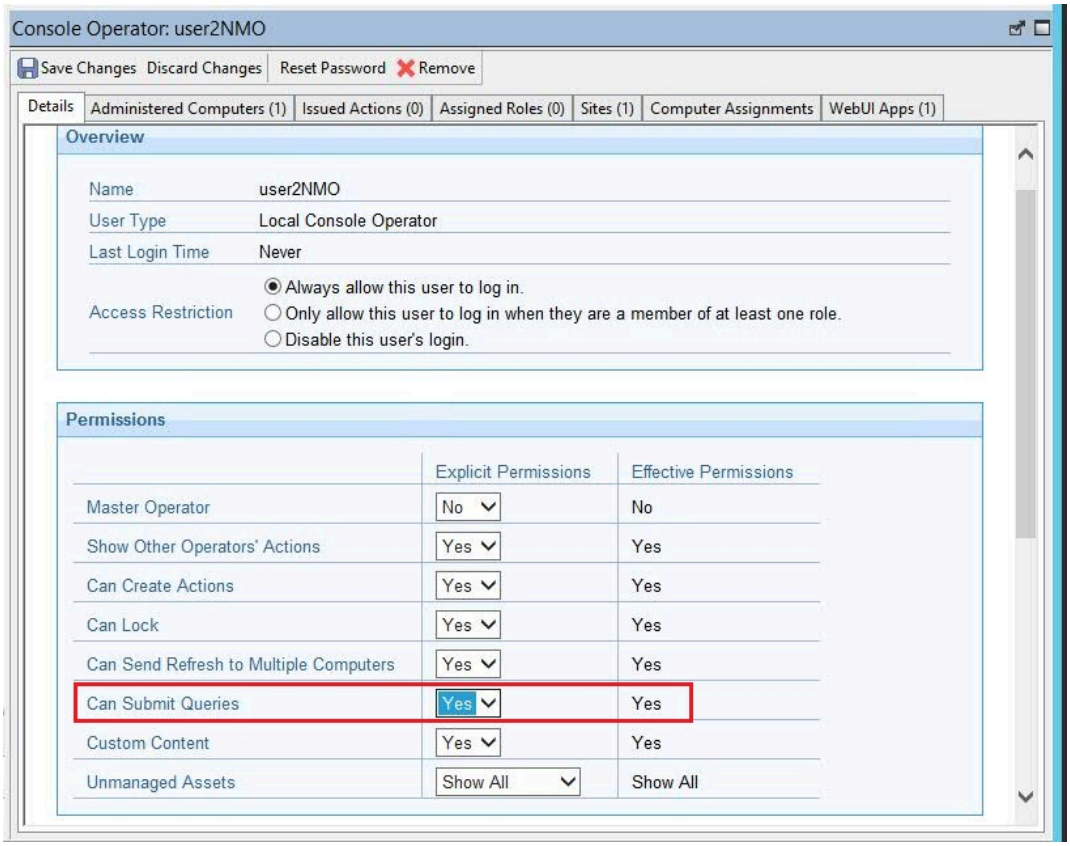


As an alternative, you can see which permissions are assigned to the users on the WebUI Applications in the working area of the **WebUi Apps** Domain. The **WebUi Apps** Domain is available under **All Contents** after you enable the WebUI.

For more information about how to access the WebUI Query Application, see How to run BigFix Query from the WebUI *(on page 206)*.

To run BigFix Query requests and see their results:

Master Operators can run queries by default. A Non-Master Operator must have, at operator or role level, the **Can Submit Queries** permission set to **Yes** in the **Details** tab:



The default value of the **Can Submit Queries** permission for Non-Master Operators is **No**.

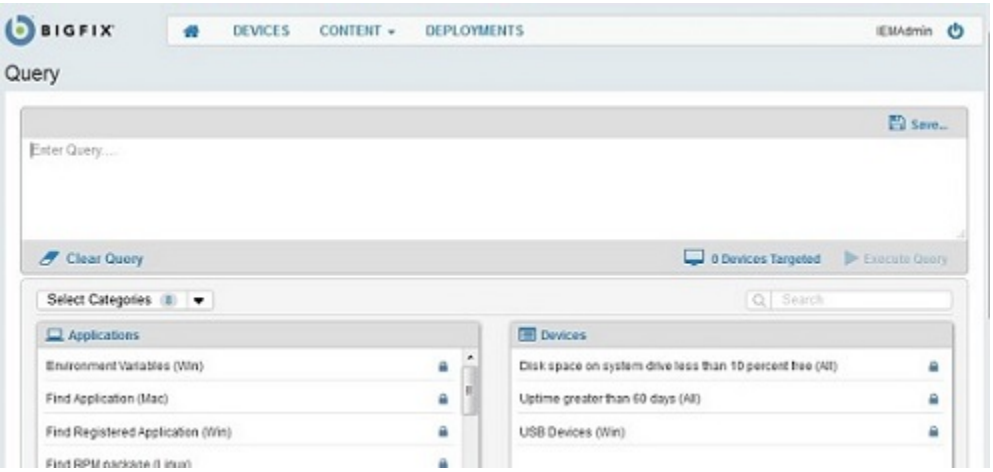
For more information about operator permissions and roles, see Adding Local Operators (on page 45).

How to run BigFix Query from the WebUI

You can access the BigFix Query on the WebUI user interface by selecting **Content -> Query**.



The Query panel opens:

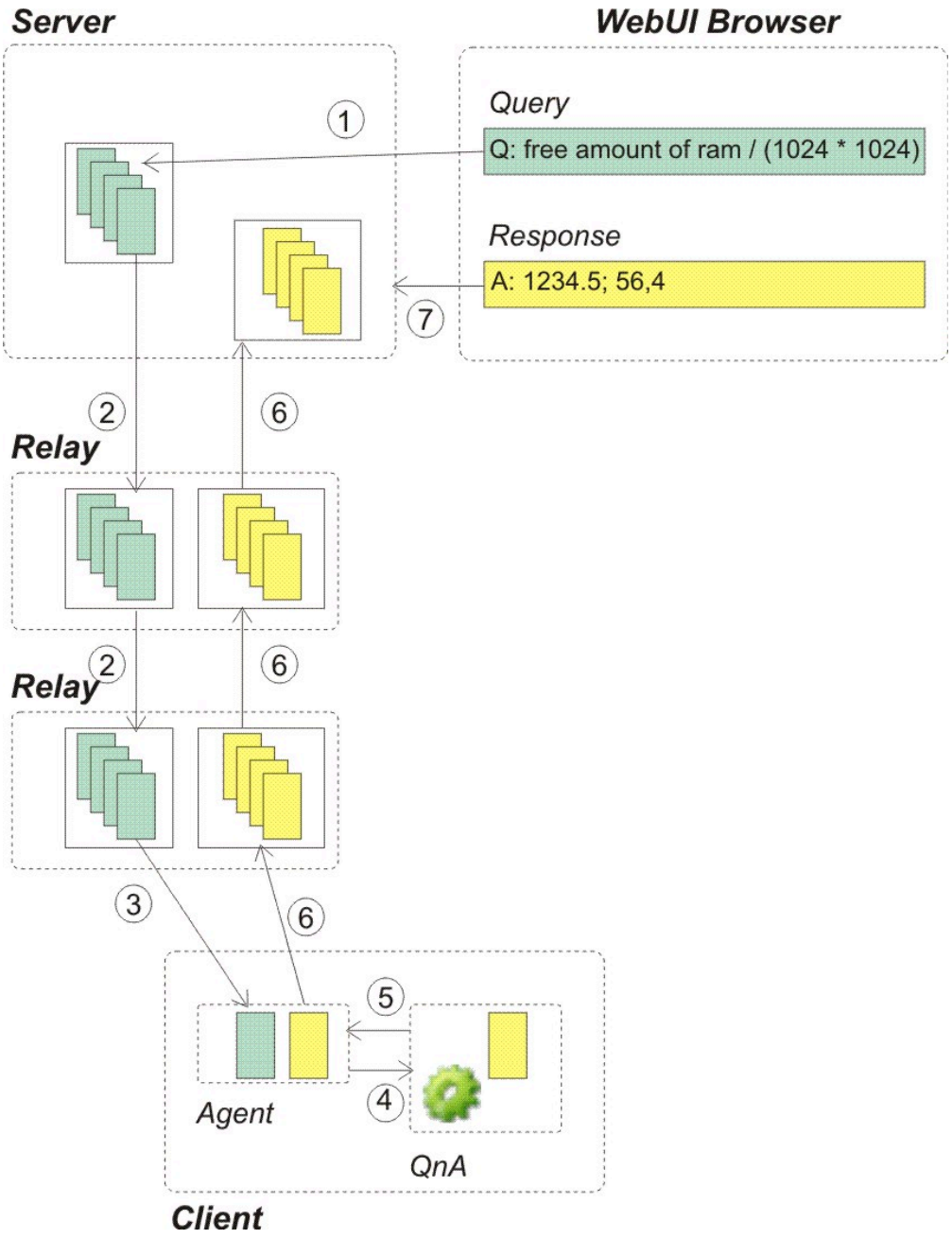


For information about using this feature from the Query panel, see Enable the WebUI (Platform V9.2.6 – V9.5.2).

How BigFix manages BigFix Query requests

A BigFix Query request is processed in a sequence of customizable steps.

The following picture shows the internal flow of a BigFix Query. Each step lists which variables you can configure to tune how BigFix Query requests and responses are managed.



1. The operator that logged on to the WebUI submits a request from the BigFix Query Application.

   What can you customize for this step?

   You can decide to run this step as an operator that is not a master operator. In this case ensure that either the operator permissions or the

permissions specified in the role assigned to the operator contain the **Can Submit Queries** value set to **Yes**.

Note: If you are using the REST API to manage the query, be aware that only the operator issuing the query can see its responses.

2. The submitted request is propagated through the relay hierarchy to the target clients using dedicated memory queues on each relay. This ensures that the request quickly reaches the targets without impacting normal BigFix processing. If a target or a child relay does not answer within a given amount of time, then it is no longer requested to answer.

   What can you customize for this step?

   From the BigFix Console you can customize, for the server and for each relay, how the memory queues are cleaned up:

   How often the cleanup task is run.

   The default value is 10 minutes and the name of the setting is **_BESRelay_Query_RemovalTask**.

   How long a request can stay in the queue before being deleted by the cleanup task.

   The default value is 60 minutes and the name of the setting is **_BESRelay_Query_MinTime**.

   The maximum size of the memory queue dedicated to BigFix Query requests.

   Before running the cleanup task, BigFix checks if the size of this memory queue exceeds the maximum size specified. If it exceeds, when the cleanup task runs, it removes the entries in the queue until the size of the queue returns within the threshold. The default value is 100 MB and the name of the setting is **_BESRelay_Query_MemoryLimit**.

For more information about these settings, see Query *(on page 807)*.

3. When the request reaches the target client parent relay, the relay informs the client, using the UDP protocol, that there is a new request to process, and, in turn, the agent retrieves the request.
4. For each responsive target, the client passes the query to the local QnA to run the query and return the results.

What can you customize for this step?

From the BigFix Console you can customize for the client:

How long can the QnA process a query issued by a Master Operator before the request timeout elapses

The default value is 60 seconds and the name of the setting is **_BESClient_Query_MOMaxQueryTime**.

How long can the QnA process a query issued by a Non Master Operator before the request timeout elapses

The default value is 10 seconds and the name of the setting is **_BESClient_Query_NMOMaxQueryTime**.

How long the QnA waits for new queries to process before stopping.

The default value is 600 seconds and the name of the setting is **_BESClient_Query_IdleTimeout**.

How much CPU is used by the QnA process running the query.

You can limit the CPU used by the QnA process by defining time slots during which the QnA runs. By default the QnA processing the query runs for 10 milliseconds and then sleeps for 480 milliseconds, which corresponds to a CPU usage lower than 1-2 %, and the name of the settings that define this behavior are **_BESClient_Query_WorkTime** and **_BESClient_Query_SleepTime**.

For more information about these settings, see Query *(on page 807)*.

Note: These settings are not considered when running the QnA tool connected as local user to the client system.

5. When the agent receives the response from the QnA, it creates a report containing the response and delivers it to the parent relay in parallel the other reports.
6. The report is delivered back to the server through the relay hierarchy. On each relay the report is stored in a memory queue while waiting to be delivered to the parent relay. If the parent relay is not available, the report waits in the queue and is delivered as soon as the parent relay becomes available again. The same criteria for encryption and signing used for normal reports are applied also to these reports.

   What can you customize for this step?

   From the BigFix Console you can customize for each relay:

   The maximum size of the memory queue dedicated to BigFix Query results.

   Before running the cleanup task, BigFix checks if the size of this memory queue exceeds the maximum size specified. If it exceeds, when the cleanup task runs, it removes the entries in the queue until the size of the queue returns within the threshold. The default value is 100 MB and the name of the setting is **_BESRelay_Query_ResultsMemoryLimit**.

   For more information about this setting, see Query *(on page 807)*.
7. When the server receives the result, it stores it in a dedicated queue from where a dedicated FillDB thread gets the data to store it in the database. In this way, the normal processing on the BigFix server is not impacted.

   The database stores, for a specified time period, both the BigFix Query request and its responses, that can be used, for example, to be filtered, displayed, or to create reports.

On a timely basis the BigFix Query Application checks the database for updates and updates the displayed results accordingly.

What can you customize for this step?

From the BigFix Administrative Tool you can customize on the server:

For how long the BigFix Query requests are stored in the database before being deleted.

The default value is 1440 hours (60 days) and the name of the advanced option is **queryHoursToLive**.

For how long the BigFix Query responses are stored in the database before being deleted.

The default value is 4 hours and the name of the advanced option is **queryResultsHoursToLive**.

How many requests and responses, for which queryHoursToLive or queryResultsHoursToLive elapsed, are deleted at a time from the database.

The default value is 100000 entries and the name of the advanced option is **queryPurgeBatchSize**.

For more information about these advanced options, see Advanced Options.

For information about how to edit computer settings, see Edit Settings for Computer.

Chapter 15. BES Server Plugin Service

For the BigFix Server Plugin Service installation and setup, see BigFix Server Plugin Service installation and configuration.

Chapter 16. The Plugin Portal

The Plugin Portal is a new component introduced in BigFix 10 to help manage cloud devices as well as modern devices such as Windows 10 and MacOS endpoints enrolled to BigFix. For details on modern client management, see Modern Client Management and BigFix Mobile.

The Plugin Portal is a prerequisite for the installation of individual plugins for cloud or modern devices on your network. It is functional only in the presence of cloud plugins.

Scalability and performance considerations

The Plugin Portal should be installed on a dedicated instance. The instance might be physical or virtual. The memory, CPU, and IO requirements scale with the number of endpoints and the amount of content being managed.

In a nutshell, the BigFix 10.0.4 or later Plugin Portal scales higher and consumes fewer resources per managed endpoint.

For details, see BigFix Performance & Capacity Planning Resources.

In BigFix 10.0.4 or later, the Plugin Portal management capabilities have grown from 10,000 to 50,000 endpoints per instance. This was achieved through the optimization of resources, and schedule management improvements.

One important optimization has been content reduction. To be more specific, in BigFix 10.0.4 or later, the Plugin Portal adds two functionalities to help filter not needed content: Custom Site management (on page 219) and Filtering the content of subscribed sites (on page 222).

In general, default installation of the BigFix 10.0.4 or later Plugin Portal has been optimized for excellent performance and resilience. Fine tuning is optional but can further improve performance; it is anyway recommended to carefully read Custom Site management (on page 219) before upgrading from older versions, since certain Custom Sites might get unsubscribed.

Prerequisites for installing the Plugin Portal

The Plugin Portal installation is supported on the following platforms:

- Windows Server 2016 or later.
- Windows 10 or later.
- Linux Red Hat 7, 8 or 9. Linux Red Hat 9 is supported starting from BigFix Version 11.0.1.

On the target computers where you intend to install the Plugin Portal, you must install:

- BigFix Agent, Version 11.0 or higher.

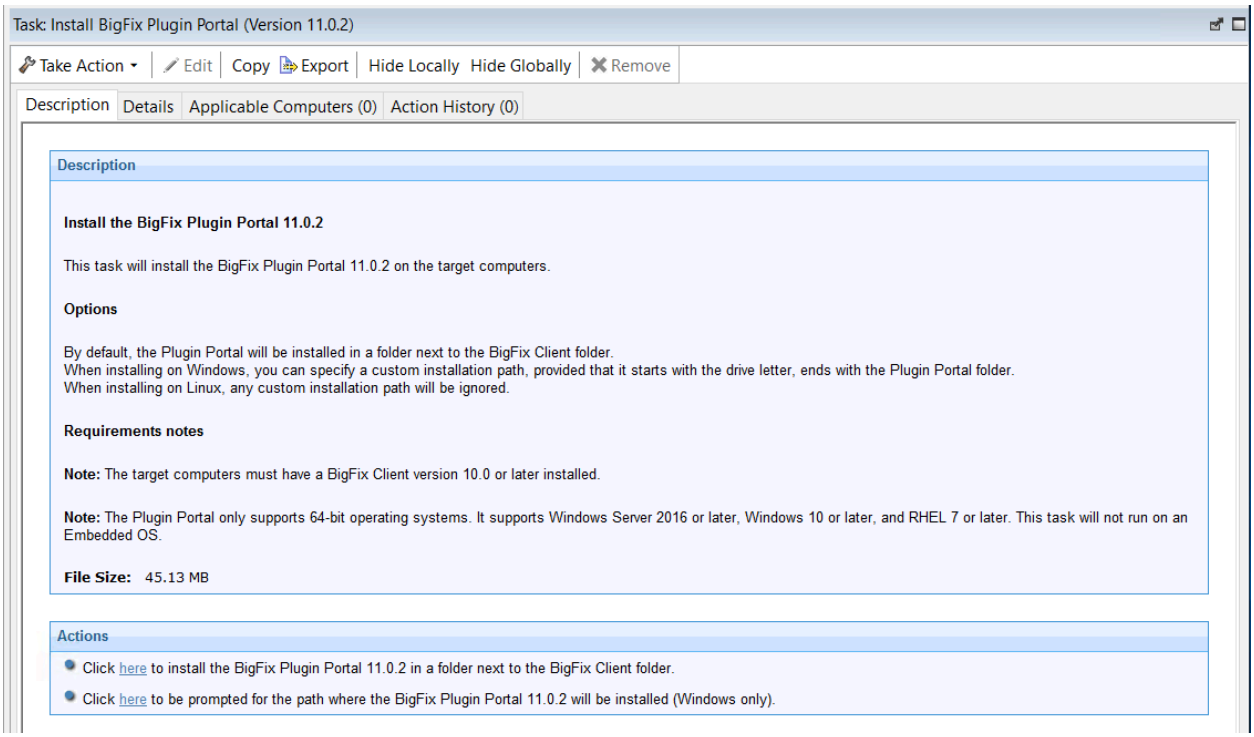
Note: You must enable the HTTPS traffic, incoming into the target computer where you installed the Plugin Portal, on the TCP port 52311. If there is a rule preventing this communication, the Plugin Portal will not receive the updates from its Parent Relay and the actions to the devices managed by the Plugin Portal will not complete.

You must ensure that on these target computers you do not have the BigFix Relay or the BigFix Server installed as the Plugin Portal is not compatible with those components. Moreover, the Plugin Portal cannot be installed on computers on which the PeerNest (on page 378) feature is enabled.

Installing the Plugin Portal

To install the Plugin Portal on a BigFix Agent, run the Task named **Install BigFix Plugin Portal (Version x)** from the BES Support site. This Task installs the BigFix Plugin Portal version x.x on the selected targets.

Note: To successfully install the Plugin Portal at 11.0.4 level or later versions, you must have the BigFix Server at least at BigFix Platform 11.0.4 level.



By default, the Plugin Portal is installed in the following directories:

- Windows:
  - C:\Program Files (x86)\BigFix Enterprise\BES Plugin Portal
- Linux:
  - /var/opt/BESPluginPortal
  - /opt/BESPluginPortal

Note: You might have several Plugin Portals in your environment, but there can be just one Plugin Portal installed on any given target computer.

When installing the Plugin Portal on Windows, you can specify a custom installation path that starts with a drive letter and ends with the Plugin Portal folder. When installing the Plugin Portal on Linux, any custom installation path is ignored.

You can configure the Plugin Portal by using a few client settings. For details, see Plugin Portal (on page 408).

You can install the Plugin Portal through WebUI as well. For details, see Installing cloud plugins.

Upgrading the Plugin Portal

Prerequisite: Upgrade the BigFix Server, Relay, and Console.

To upgrade the Plugin Portal, run the Fixlet named **Updated Plugin Portal - BigFix version X** from the BES Support site. The Fixlet becomes relevant on your endpoints only after the BigFix Server is upgraded. Clients acknowledge that the BigFix Server is upgraded and then begin to report any other upgrade fixlets as relevant (by default, the clients check for the version of the server on registration every once in six hours).



Note: To successfully upgrade the Plugin Portal to 11.0.4 or later versions, you must have the BigFix Server at least at BigFix Platform 11.0.4 level.



Note: Upgrading the Plugin Portal to 11.0.4 or later versions is only possible from versions greater or equal to 10.0.9.

Upgrading the Plugin Portal might prevent your proxied endpoints from reporting on action status and retrieved properties for a specific duration. It is recommended that you run the upgrade when the disruption to your IT operations due to this is minimal. For assistance, contact HCL Technical Support.

Uninstalling the Plugin Portal

To uninstall the Plugin Portal from an endpoint, run the Task named **TROUBLESHOOTING: Uninstall BigFix Plugin Portal** from the BES Support site. The Task removes the following:

- The Plugin Portal
- Client settings as relevant
- Files and directories within the Plugin Portal installation and storage directories

Discovery on Demand

Starting from BigFix 11.0.6, the Plugin Portal adds a method to manually trigger the discovery process for specific cloud plugins on demand.

This new capability, introduced in Version 11.0.6, forces an immediate update of the discovered assets without waiting for the scheduled discovery interval, which is particularly useful during troubleshooting or when immediate visibility of new assets is required.

The new "Discovery on Demand" feature allows administrators to trigger the discovery immediately on the specified Plugin Portal computers.

Mechanism

There are now two methods to trigger this discovery:

1. **Push Notification (via REST API)**
2. **Notify Client Command (via ActionScript)**

Push Notification (REST API)

The discovery can be triggered using the `pushnotification` REST API with the notification type PluginPortalDiscovery.

```
https://<BigFix_Server_IP>:<Port>/api/pushnotification
```

The XML payload structure must contain a `pushnotification` element with the following child elements:

- **Type**: Must be set to "PluginPortalDiscovery"
- **Payload**: A list of cloud plugin IDs separated by | (pipe). Use "ALL" to trigger a discovery for all installed cloud plugins
- **ComputerID**: The unique ComputerID of the computer where the Plugin Portal is installed.

Below is an example of how to trigger the discovery using the curl command and a defined XML file.

1. Create the XML payload (named, for example, `pushStartDiscovery.xml`):

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

 <Payload>VMwareAssetDiscoveryPlugin|AzureAssetDiscoveryPlugin</Payloa
d>
        <ComputerID>1617618421</ComputerID>
    </PushNotification>
</BESAPI>
```

2. Send the request using the cURL command:

```
curl -X POST --data-binary
@/file_path/pushStartDiscovery.xml -u <BigFix Master
 Operator>:<password>
https://192.14.77.34:52311/api/pushnotification
```

ActionScript Command

The discovery can also be triggered directly through BigFix ActionScript using the `notify client` command and targeting the computer where the Plugin Portal is installed. This allows the discovery to be embedded into the Fixlets.

The syntax is as follows:

```
notify client PluginPortalDiscovery <list of plugins>
```

where `<list of plugins>` is a string containing cloud plugin IDs separated by | (pipe), or the keyword ALL to trigger every installed cloud plugin.

For example:

```
notify client PluginPortalDiscovery
 "VMwareAssetDiscoveryPlugin|GCPAssetDiscoveryPlugin
|AWSAssetDiscoveryPlugin|AzureAssetDiscoveryPlugin"
```

Custom Site management

Usually, Custom Sites are designed to distribute content to endpoints having the BigFix Agent installed.

Subscribing the devices discovered by the Plugin Portal to a Custom Site with unusable content requires unnecessary effort to evaluate and manage the content, producing unnecessary resource usage and poor performance.

For this reason, starting from BigFix 10.0.4, the Plugin Portal adds an additional filter to subscribe the devices it discovers to Custom Sites.

Before BigFix 10.0.4, the Plugin Portal subscribed the devices to a Custom Site after evaluating the applicability relevance resulting from the subscription conditions of the site. Now the applicability relevance will be evaluated by the Plugin Portal **only** if it uses one of the following inspectors:

- **in agent context**
- **in proxy agent context**
- **in plugin portal context**

If the applicability relevance of the Custom Site **does not** use any of the inspectors mentioned above, the Custom Site is not evaluated by the Plugin Portal: the devices it discovers will not subscribe the Custom Site.

If the applicability uses at least one of those inspectors, the subscription to the site **depends on the evaluation of the applicability relevance**, as happened with previous releases.

To clarify with an example, before BigFix 10.0.4, to subscribe all agents to a Custom Site you can use this relevance expression:

- **true**

corresponding to the ***All Computers*** choice in the **Computer Subscription** Console tab. Starting from BigFix 10.0.4, if you need to include also the Plugin Portal agents, you have to explicitly use one of the above described inspectors, for example:

- **true or in plugin portal context**

otherwise the agents discovered by the Plugin Portal **do not subscribe** to the site.

This change only applies to the Plugin Portal represented endpoints, not to endpoints represented by the Proxy Agent.

Nothing changes for agents not discovered by the Plugin Portal, they subscribe the Custom Site depending on the evaluation of the site applicability relevance.

Nothing **changes** for the content of the Custom Site: if a device subscribes the site, the applicability of each content element (Fixlet, task, analysis...) depends only on the relevance specified for the element.

The feature can be disabled by setting

_BESPluginPortal_Performance_ExcludeCustomSitesSubscription = 0

to the client where the Plugin Portal is installed. With this setting, the Plugin Portal behaves as in versions prior to 10.0.4.

This feature is enabled by default. As a consequence, when upgrading the Plugin Portal from versions older than 10.0.4, the **Plugin Portal agents might get unsubscribed from certain Custom Sites**. If you want to prevent this behavior, you must disable the feature before upgrading. When upgrading using the BigFix Console, the Fixlet **Updated Plugin Portal - BigFix version X** also gives you an option to disable it. This option is not available using the BigFix WebUI.

Here are some examples of Custom Site applicability relevance:

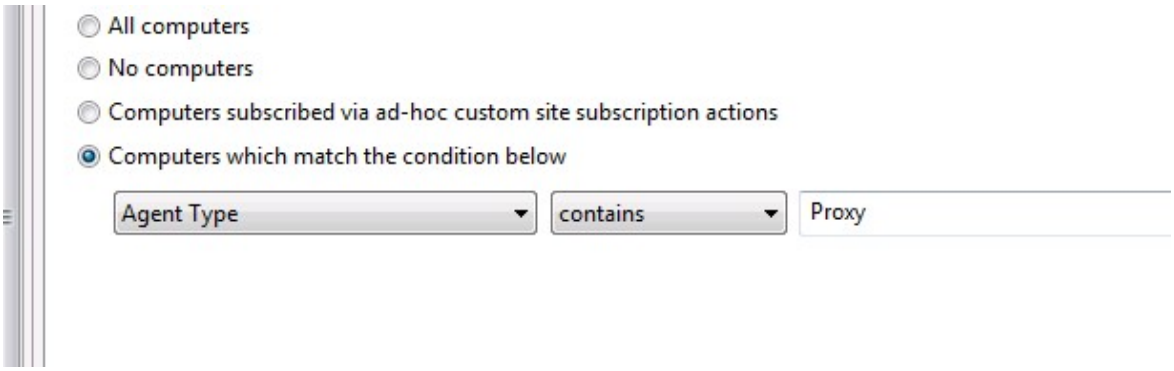
- to subscribe only the Plugin Portal agents:
  - **in plugin portal context**
- to subscribe only the native agents:
  - **in agent context**
- to subscribe all agents but not the Plugin Portal ones:

    - **true**
- to subscribe all agents, use one of these:
    - **true or in plugin portal context**
    - **true or in proxy agent context**
- to subscribe only the ***proxied*** agents, which are the ones discovered by the Plugin Portal or by the Proxy Agent, use one of these:
    - **in proxy agent context**
    - **not in agent context**
- to subscribe all the Plugin Portal agents whose names contain “dept1”, case insensitive:
    - **in plugin portal context and exists (computer name) whose (it as string as lowercase contains "dept1")**
- to subscribe all the proxied agents installed on a Red Hat OS:
    - **in proxy agent context and exists (operating system) whose (name of it as lowercase contains "red hat")**
- to subscribe all agents installed on a Red Hat OS except the Plugin Portal ones:
    - **exists (operating system) whose (name of it as lowercase contains "red hat")**

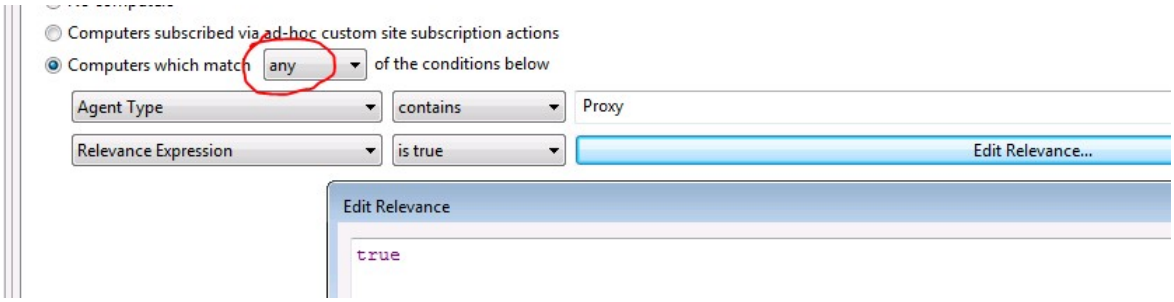
Here are some examples using the **Computer Subscription** Console tab.

Since the Console resolves the **Agent Type** condition into an expression that uses the **in proxy agent** inspector, setting this condition will be enough to ensure that it is not ignored by the Plugin Portal.

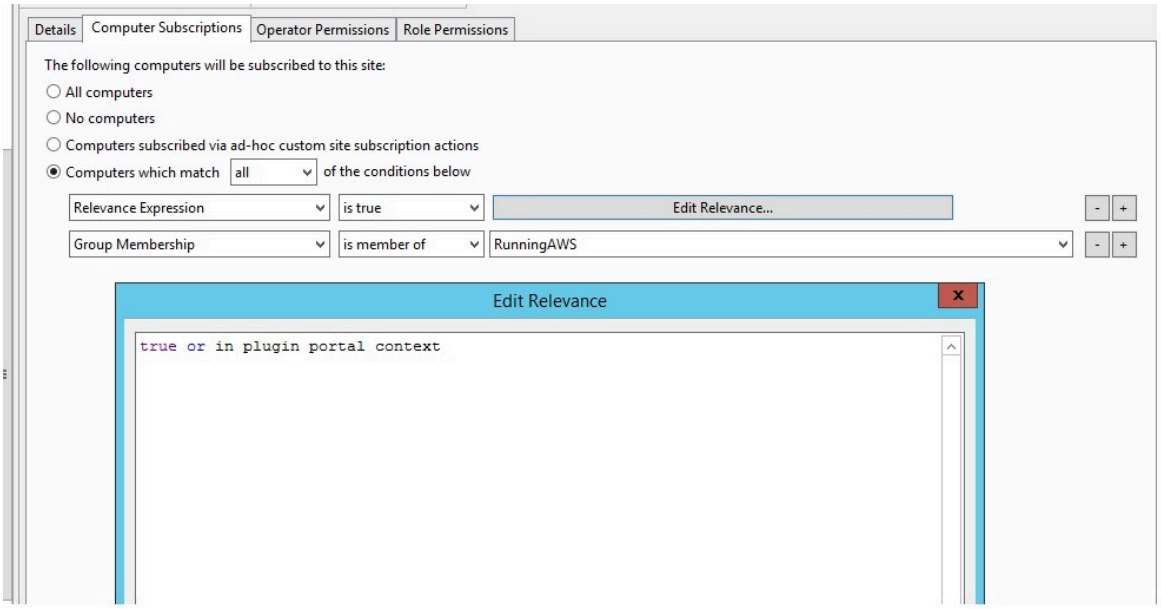
- To subscribe only proxied agents:



- To subscribe all the agents:



- To subscribe all agents of the specified Server Based Computer group:



Filtering the content of subscribed sites

To reduce the load on the Plugin Portal, starting from BigFix 10.0.4, the Plugin Portal adds a new functionality to avoid managing unnecessary content on the devices it discovers.

Reducing the content is the first thing to do to improve performance, speed up operations and reduce costs. It is highly recommended that you follow the instructions below to filter out Fixlets and analyses not needed to devices discovered by the Plugin Portal for all your External Sites that cannot be entirely excluded. This may be achieved by setting an appropriate subscription relevance or your Custom Sites passing the filter described in Custom Site management (on page 219).

The content filtering functionality described below is simpler and more efficient than changing the relevance of each not needed Fixlet, action or analysis, since the Plugin Portal is not forced to evaluate unnecessary relevance.

After a site is downloaded, at gathering time, the Plugin Portal can exclude some Fixlets or analyses from the evaluation process. It is sufficient to specify the excluded content in a json file, named `PluginPortalSubscriptionOptions.json`, and include it to the site itself.

The json file is composed of three arrays:

- **ExcludedFiles**: array of strings defining the list of **FXF names** to be ignored.
- **ExcludedFixletIDs**: array of integers defining the list of **Fixlet/task IDs** to be ignored.
- **ExcludedAnalysisIDs**: array of integers defining the list of **analysis IDs** to be ignored.

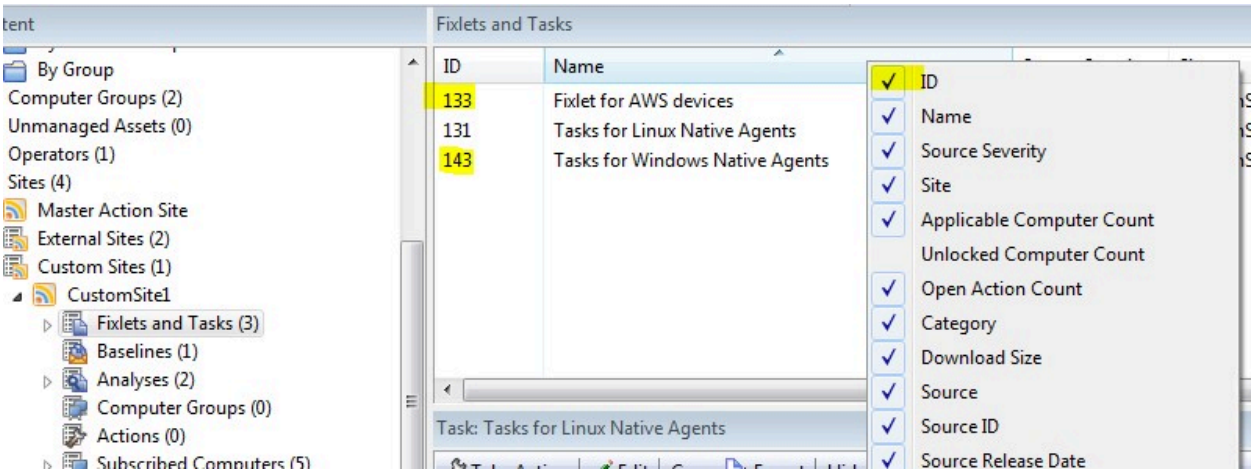
An example of json file:

```
{
  "ExcludedFiles": [
   "My Windows analyses.fxf",
   "Department 121.fxf"
  ],
  "ExcludedFixletIDs": [
   209, 31, 405
  ],
  "ExcludedAnalysisIDs": [
   211, 33
  ]
 }
```

Providing a file name, all the content of that file is ignored by the Portal Plugin during the evaluation phase. If it is not possible to specify a file name, such as for **Custom Sites** or for FXF defining content that must be consumed by proxied devices too, you can provide a list of Fixlet, Task or Analysis IDs to be ignored.

Below is an example using a Custom Site.

To get the IDs of content to exclude, open the **Fixlets and Tasks** BigFix Console panel for the Custom Site and look at the ID column. If the column is not visible, right click the column names and check the ID entry to display the corresponding column.

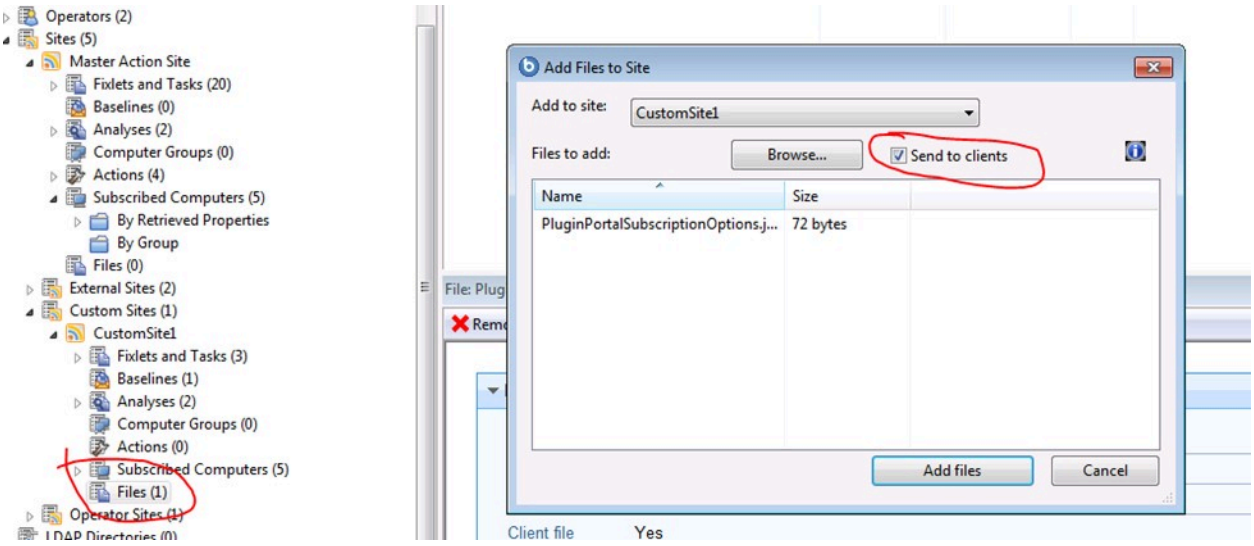


Do the same in the **Analyses** panel.

Once you got the IDs, create the json file anywhere in the BigFix Console machine:

```
{
  "ExcludedFixletIDs": [
   133, 143
  ],
  "ExcludedAnalysisIDs": [
   211, 33
  ]
 }
```

Then, you can add the json file to the Custom Site. Ensure that you check the **Send to clients** option.



By clicking **Add files** the new version of the Custom Site will be distributed to all subscribed computers, including the ones discovered by the Plugin Portal.

Action commands for configuring the BES Plugin Portal plugins

The Plugin Portal plugins can be configured by storing settings in the PluginStore sqlite database. This can be easily accomplished by using some action commands specifically designed for the task.

Introduction

The commands syntax to operate on the PluginStore database through the action scripts is as follows:

```
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}"
```

The **plugin store** command accepts the name of a Plugin, such as `VMwareAssetDiscoveryPlugin`.

The **plugin store key** that follows the **set** keyword always looks like a set of words, called subsections, separated by an underscore, such as `Log_Path`.

The parameter that follows the **value** keyword can be whatever is appropriate to the chosen setting.

An example of how a command should look like:

```
plugin store "AWSAssetDiscoveryPlugin" set "Log_Verbose" value "0" on
 "{parameter "action issue date" of action}"
```

For your reference, consider that the PluginStore database schema is as follows:

Key	Value	Effective Date
TEXT NOT NULL PRIMARY KEY	KEY TEXT NOT NULL	DATE

For more details about the plugin store command, see plugin store.

Configuring the cloud plugins

Several cloud plugins can be installed on the Plugin Portal, in order to manage Amazon Web Services, Microsoft Azure, Google Cloud Platform and VMware cloud environments.

All plugin store commands to configure the plugins should start with the **plugin store** command. For more details on the commands syntax to operate on the PluginStore database, see Introduction (on page 225).

The required plugin names that should follow the "plugin store" keyword for the configuration of the cloud plugins are:

- **AWSAssetDiscoveryPlugin**

- **AzureAssetDiscoveryPlugin**

- **GCPAssetDiscoveryPlugin**

- **VMwareAssetDiscoveryPlugin**

This information is key to configure the settings in the Plugin Store, since the plugin store action commands rely on those names to correctly build up the settings keys.

The options accepted by the **plugin store** command are **set**, **multiple set** and **delete**. If the **set** and **multiple set** keywords are decorated with the keyword **encrypted**, it will result in an encrypted value in the database.

The **set** option must be followed by the Plugin Store key that we want to set, followed by the **value** keyword and the value itself that we want to store. Lastly, the current date is specified after the **on** keyword.

Here follows an example:

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}"

The following settings will display in the Plugin Store:

Key	Value	Effective Date
_AWSAssetDiscoveryPlugin_Credentials_AccessKey_myLabel	myAccessKey	0123456789
_AWSAssetDiscoveryPlugin_Credentials_SecretAccessKey_myLabel	{obf}ABCDEF...	0123456789

The **multiple set** option is used to quickly configure multiple settings at once. The **encrypted** option is available for this as well. The **multiple set** option must be followed by a percent encoded JSON that contains a list of the key-value pairs that must be added to the database. An example of the decoded JSON is the following:

```
{
    "Credentials_AccessKey_myLabel" : "myAccessKey",
    "Credentials_Region_myLabel" : "myLabelRegion",
    "HTTP_ProxyURL" : "myProxyURL",
    "HTTP_ProxyUser" : "myProxyUser"
}
```

For example, the output of the following command:

plugin store "AWSAssetDiscoveryPlugin" **multiple set** <example json> **on** "{parameter "action issue date" of action}"

should be the addition of the following settings to the Plugin Store:

Key	Value	Effective Date
_AWSAssetDiscoveryPlugin_Credentials_AccessKey_myLabel	myAccessKey	0123456789
_AWSAssetDiscoveryPlugin_Credentials_Region_myLabel	myLabelRegion	0123456789
_AWSAssetDiscoveryPlugin_HTTP_-ProxyURL	myProxyURL	0123456789
_AWSAssetDiscoveryPlugin_HTTP_-ProxyUser	myProxyUser	0123456789

The "on" keyword is required by the set commands and should be followed by the date at which the setting is issued:

[...] **on** "{parameter "action issue date" of action}"

The delete option will simply remove a certain key from the plugin store:

plugin store "AWSAssetDiscoveryPlugin" **delete** "Credentials_Region_myLabel"

Key	Value	Effective Date

_AWSAssetDiscoveryPlugin_Credentials_AccessKey_myLabel	myAccessKey	0123456789
_AWSAssetDiscoveryPlugin_HTTP_ProxyURL	myProxyURL	0123456789
_AWSAssetDiscoveryPlugin_HTTP_ProxyUser	myProxyUser	0123456789

Issuing the delete command followed by the keyword **all** will result in all the Plugin Store settings for a given plugin to be deleted.

plugin store "AWSAssetDiscoveryPlugin" **delete all**

Many of the plugins settings are bound to a certain set of credentials, which in turn are identified by what we call a credential label. As such, is a good practice, when building an action script, to store a label in an action parameter in order to re-use it multiple times later in the action script:

```
parameter "credentialsLabel" = "<my label>"
```



Note: The values that you must replace are only the ones wrapped by angled brackets, like <my label>.

If a key or part of it needs to be repreated multiple times, it can be stored in a parameter as well and it can be concatenated later.

```
parameter "accessKey" = "Credentials_AccessKey"
parameter "secretAccessKey" = "Credentials_SecretAccessKey"
```

For example, a new parameter can be defined by the concatenation of the previous parameters:

```
parameter "credentialsLabel" = "<my label>"
```

When configuring the plugin programmatically it could be useful to check whether a parameter exists or if it is not empty, in order to achieve that, simply wrap your code in an if block:

```
if {(exists parameter "myParam")AND (parameter "myParam" != "")}
 // my code
endif
```

Common Plugin settings

The following settings are common among all cloud plugins:

Discovery_Region - The default region for the plugin. This region will be used to retrieve the list of region enabled for all the AWS accounts related to the credentials stored in the plugin. This setting is mandatory.

Log_Path - The path of the log of the plugin.

Log_Verbose - When set to 1, debug logging is enabled. When set to 0, only info logging is displayed.

JSON file settings - Some settings are defined for the cloud plugins through a JSON called settings.json. Here is an example of such a JSON.

```
{
    "ID": <plugin name>,
    "ConfigurationOptions": "",
    "DeviceReportRefreshIntervalMinutes": <refresh interval in minutes>,
    "DeviceReportExpirationIntervalHours": 168,
    "CommandFormat": "JSON",
    "SendSettingsToPlugin": [],
    "ExecutablePath": <executable path>,
    "HandlePartialRefresh": false,
    "FullReportsInRefreshAll": true,
    "NoRefreshBeforeActionIntervalMinutes": 60
}
```

AWSAssetDiscoveryPlugin configuration

Here are the settings needed to fully configure the Amazon Web Services plugin.

IAM User specific settings

Credentials_AccessKey_<label> - The Access Key ID associated to an IAM User. This setting is mandatory.

Credentials_SecretAccessKey_<label> - The Secret Access Key associated to an IAM User. The value of this setting must be encrypted. This setting is mandatory.

Credentials_Region_<label> - The default region for the IAM User credentials with label <label>. This region overrides the Discovery Region.

Credentials_Roles_<label>_<arn> - The region of the role with ARN <arn> to be assumed by the IAM User with label <label>. This region overrides both the Credentials Region and the Discovery Region. The value can be empty.

Credentials_Roles_ExternalId_<label>_<arn> - The external ID of the role with ARN <arn> to be assumed by the IAM User with label <label>. The value must be encrypted. The setting can be omitted if the IAM Roles does not require an external ID.

Advanced settings

HTTP_ProxyURL - The URL of the HTTP Proxy for the plugin.

HTTP_ProxyUser - The User of the HTTP Proxy for the plugin.

HTTP_ProxyPassword - The Password of the HTTP Proxy for the plugin. The value of this setting must be encrypted.

RegionAllowedList_<label> - Forces the plugin to execute discovery only on the listed regions for the user with label <label>. Separate regions with a semicolon ';'.

Example: `eu-central-1;eu-west-1;us-east-1`

When installing the AWS plugin, you can specify the allowed regions. For more details about how to limit the AWS regions, see Limit AWS Regions to restrict the scope of device discovery.

Example of AWSAssetDiscoveryPlugin configuration

Initializing some parameters:

```
parameter "firstLabel" = "foo"
parameter "secondLabel" = "bar"

parameter "accessKey" = "Credentials_AccessKey"
parameter "secretAccessKey" = "Credentials_SecretAccessKey"
```



Note: foo and bar are just invented names for our labels. However, **Credentials_AccessKey** and **Credentials_SecretAccessKey** are real setting names. We are defining these four parameters because, by combining them, we can define the keys that we need to set the user key and password like specified above.

Setting the plugin default region:

```
plugin store "AWSAssetDiscoveryPlugin" set "Discovery_Region" value
 "eu-west-1" on "{parameter "action issue date" of action}"
```

Configuring the first user:

```
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}"
```

Note: We combined the first four parameters into two new parameters, by concatenating them by pairs, in order to fully define the Plugin Store keys. To concatenate the parameters, we are just assigning to a new parameter the string composed by the two parameters separated by an underscore. As such, the content of the parameter “firstUserAccessKey” would be “Credentials_AccessKey_foo”.

Configuring the second user:

```
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}"
```

Setting the log to verbose:

```
plugin store "AWSAssetDiscoveryPlugin" set "Log_Verbose" value "1" on
 "{parameter "action issue date" of action}"
```

Setting the list of allowed regions:

```
plugin store "AWSAssetDiscoveryPlugin" set "RegionAllowedList_{parameter
 "secondLabel"}" value "us-east-1;us-west-1" on "{parameter "action issue
 date" of action}"
```

When installing the AWS plugin, you can specify the allowed regions. For more details about how to limit the AWS regions, see Limit AWS Regions to restrict the scope of device discovery.

An example of the expected output in the PluginStore is:

Key	Value
_AWSAssetDiscoveryPlugin_Credentials_AccessKey_foo	myUs
_AWSAssetDiscoveryPlugin_Credentials_SecretAccessKey_foo	{obf}A
_AWSAssetDiscoveryPlugin_Credentials_Region_foo	eu-ce
_AWSAssetDiscoveryPlugin_Credentials_AccessKey_bar	myUs
_AWSAssetDiscoveryPlugin_Credentials_SecretAccessKey_bar	{obf}A
_AWSAssetDiscoveryPlugin_Credentials_Roles_bar_fakeRoleARN1	us-ea
_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-ea

AzureAssetDiscoveryPlugin configuration

Here are the settings needed to fully configure the Microsoft Azure Plugin.

IAM User specific settings

Credentials_ClientID_<label> - The Client ID for the user with label <label>.

Credentials_ClientSecret_<label> - The Client Secret for the user with label <label>.

Credentials_SubscriptionID_<label> - The Subscription ID for the user with label <label>.

Credentials_TenantID_<label> - The tenant ID for the user with label <label>.

Example of AzureAssetDiscoveryPlugin configuration

```
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}"
```

Where:

myTenantID

Is the tenant ID for the user.

myClientID

Is the Client ID for the user.

myClientSecret

Is the Client Secret for the user.

mySubscriptionID

Is the Subscription ID for the user.

Note: If your Azure credential needs to discover resources located in an Azure Cloud environment different from the public one, see the "Azure cloud plugin environment" section in Configuring cloud plugins *(on page 279)*.

GCPAssetDiscoveryPlugin configuration

Here are the settings needed to fully configure the Google Cloud Platform Plugin.

Service Account specific settings

Credentials_JSON_<label> - The encrypted JSON key related to a service account member of a project on GCP.

The GCP JSON key file will look something like this:

```
{
  "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"
}
```

The JSON key must be percent encoded before feeding it to the plugin store action command. Since all the information required to configure a service account into the GCP Plugin is contained in the JSON, this is the only setting to insert or delete.

The JSON must be encrypted, since it contains the private key.

Example of GCPAssetDiscoveryPlugin configuration

```
parameter "jsonKey" = "<percent encoded json>"
plugin store "GCPAssetDiscoveryPlugin" set encrypted
 "Credentials_JSON_foo" value "{parameter "jsonKey"}" on "{parameter
 "action issue date" of action}"
```

VMWareAssetDiscoveryPlugin configuration

Here are the settings needed to fully configure the VMware Plugin.

IAM User specific settings

Credentials_Username_<label> - The username for the user with label <label>.

Credentials_Password_<label> - The encrypted password for the user with label <label>.

Credentials_URL_<label> - The credentials label for the user with label <label>.

Credentials_CertPath_<label> - The full path of the custom certificate for the user with label <label>.

Example of VMwareAssetDiscoveryPlugin configuration

```
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}"
plugin store "VMWareAssetDiscoveryPlugin" set
 "Credentials_CertPath_{parameter "myLabel"}" value "myCert" on "{parameter
 "action issue date" of action}"
```

Where:

myUsername

Is the username for the user.

myURL

Is the credentials label for the user.

myPassword

Is the encrypted password for the user.

Configuring the BESPluginPortal Plugins through REST API

The plugins configuration can be carried out by issuing BigFix Actions through rest APIs.

The data to be passed to the rest APIs must be a BES XML file.

It is sufficient to create an action script containing the commands for the desired configuration, and then fill the fields of the BES XML for the actions accordingly. An example of such XML is the one below.

```
<?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>
```

The **ActionScript** element will contain the action script text.

The target of this action should be the Native representation of the BESPluginPortal machine on which the plugins to be configured are installed. As such, it is convenient to use the ComputerID element inside the Target field and fill its value with the appropriate computer ID.

After creating the XML file or the XML string by code, the action described in the XML can be issued by executing a POST.

The following command will execute the POST with cURL. The file action.xml will be posted to the server <server> through port <port> (usually 52311 for BigFix) and it will create and execute the action contained in action.xml.

```
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
```

For more information about the Action REST API resource, see Action or the BES.xsd and BESAPI.xsd files.

Alternatively, you can use the IEM CLI to easily issue requests to BigFix REST APIs.

In order to do that, first open a session with the IEM CLI. Open a terminal and locate the IEM CLI in the BigFix Server folder.

Then, run the following command:

```
iem login --server=<server>:<port> --user=<user> --password=<password>
```

Then, after the session is open, simply issue the request with the following command:

```
iem POST <path to action.xml> /api/actions
```

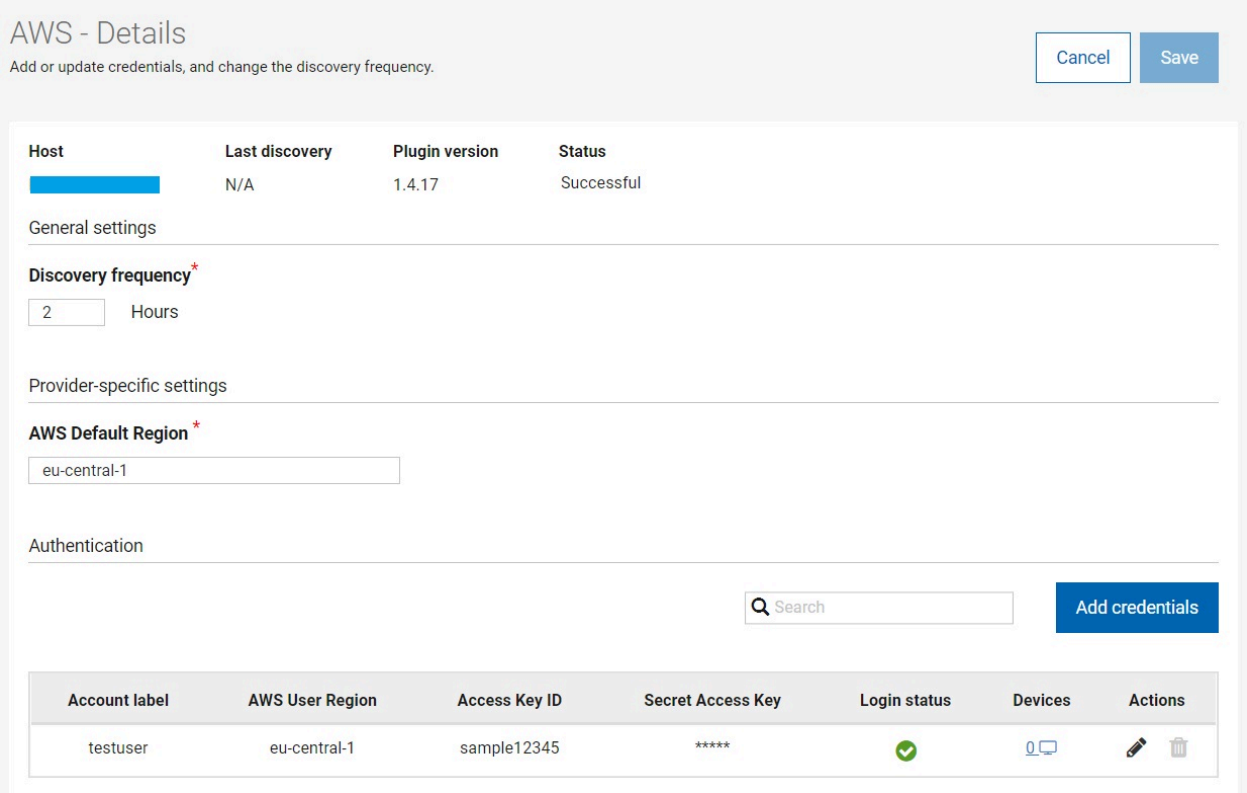
For further reference, see Running Requests from the IEM CLI.

Example 1: Configuring the AWS Plugin

In this example, we are going to configure the AWS Asset Discovery Plugin with the aid of a single action dispatched through the BigFix REST APIs.

Suppose that we already installed the AWS Plugin and have a user already configured but we want to quickly configure a bunch of other users.

As shown in the picture below, we can see that a Plugin is installed and in a Successful status, with a **testuser** configured that correctly executed the login.



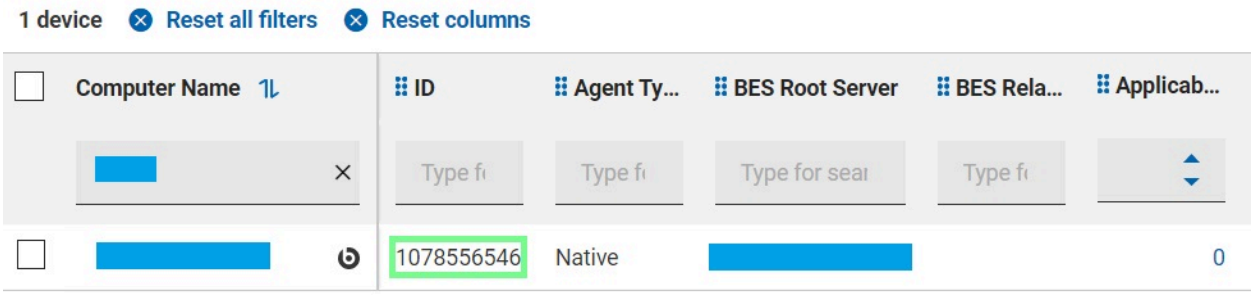
Let us now build an XML file that will be used to update our configuration. Such XML file will be passed as the data file by the REST API command.

Refer to the Configuring the BESPluginPortal Plugins through REST API (on page 238) and The BigFix REST API pages for more information.

We will use the following template for a simple action:

```
<?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>
```

As you can see from the comments in the XML file, we have to fill in at least two fields. Let us begin from locating the **ComputerID**, which would be the ComputerID of the Portal machine upon which the AWSAssetDiscoveryPlugin is installed.



Now let us build the action script for the configuration. Suppose that we want to configure two users, namely **testuser2** and **testuser3**. Those two users will have respectively, **us-east-1** and **af-south-1** as their regions, which should have priority over the default plugin region.

As mentioned in Configuring the cloud plugins *(on page 226)*, let us store useful keys and data in the action parameters:

```
parameter "firstLabel" = "testuser2"
parameter "secondLabel" = "testuser3"

parameter "accessKey" = "Credentials_AccessKey"
parameter "secretAccessKey" = "Credentials_SecretAccessKey"
parameter "region" = "Credentials_Region"
```

Then, write the commands to configure the plugin store settings:



Note: The values that you must replace are only the ones wrapped by angled brackets, like <myUserKey2>.

```
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}"
```

Now that we have what we need, simply save the filled XML file with a custom name, such as **action.xml**:

```
<?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>
```

Note: testuser2 and testuser3 are just invented names for our labels. However, **Credentials_AccessKey**, **Credentials_SecretAccessKey** and **Credentials_Region** are real setting names. We are defining these parameters because, by combining them, we can define the keys that we need to set the user key and password like specified above.

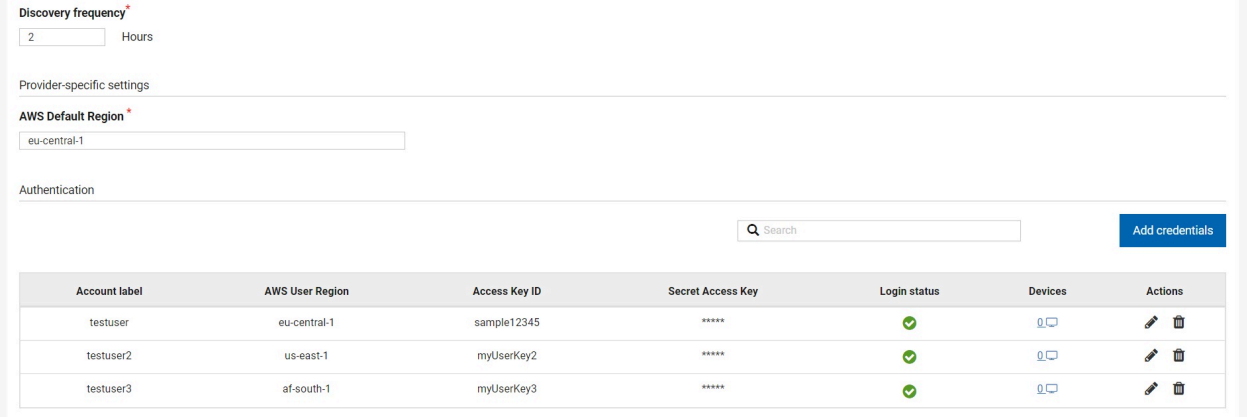
Issue the following command through the IEM CLI in order to launch the action contained in **action.xml** through REST API:

```
iem POST <path to action.xml> /api/actions
```

The output will be something like this:

```
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>
```

Check on the WebUI whether the action was completed successfully. Should that be the case, the two new users will be immediately available:

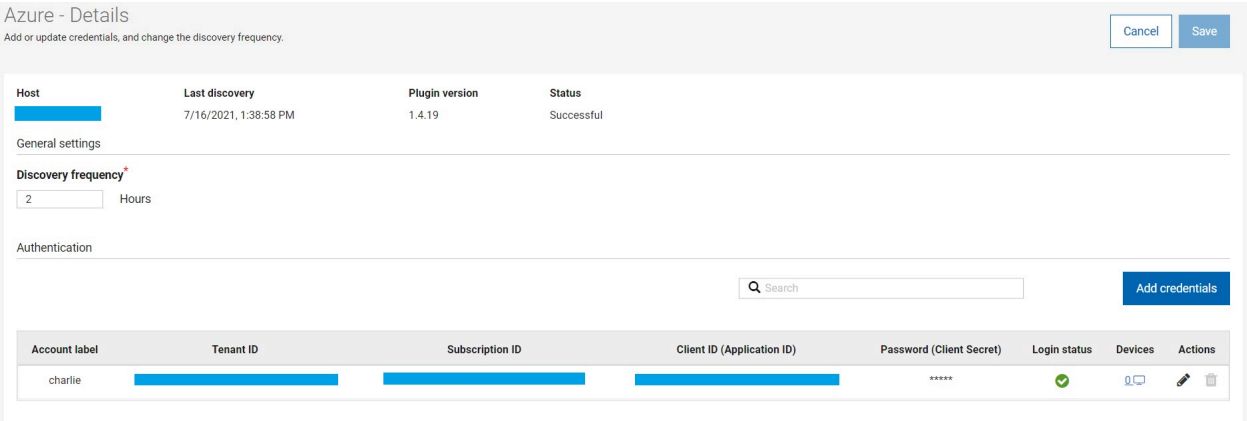


Example 2: Configuring the Azure Plugin with a Custom Action

In this example, we are going to configure the Azure Asset Discovery Plugin by creating and executing a custom action through the BigFix WebUI.

Like in the AWS example described in Example 1: Configuring the AWS Plugin (on page 240) we can assume that we already installed the Azure Plugin.

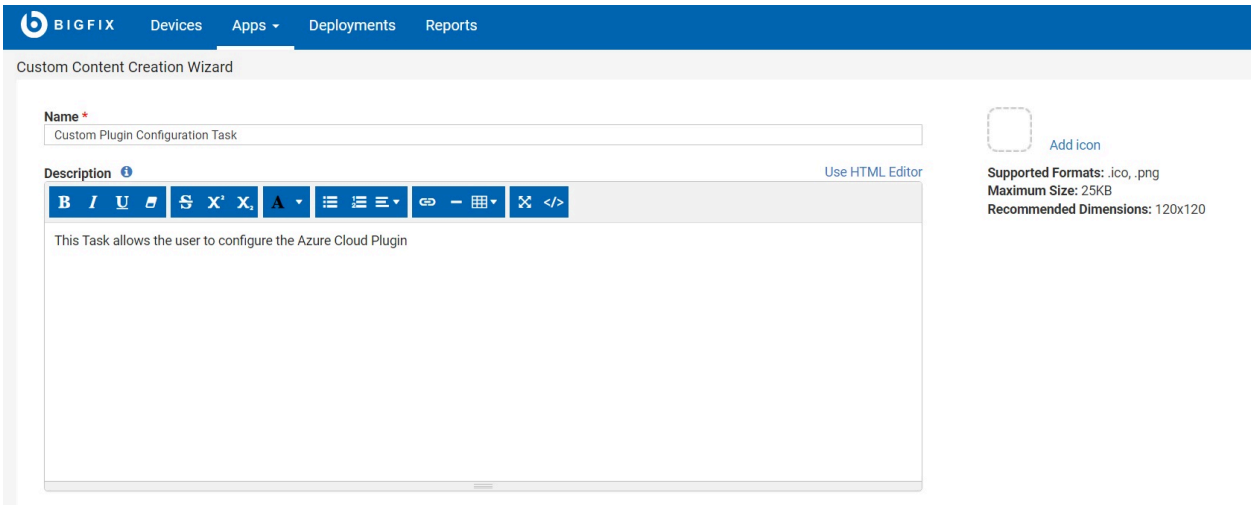
See below the details reported by the WebUI about the Azure Plugin.



We want to create a custom action for configuring two users, **foo** and **bar**, for which we have their **TenantID**, **SubscriptionID**, **ClientID** and **ClientSecret** available.

Under **WebUI > Apps > Custom**, click **Create Custom Content**.

Type in a Name and a Description for the item, if you want.



Then, we could add the following relevances to ensure that we target the right device (which should always be the Plugin Portal on which the Azure Plugin is installed):

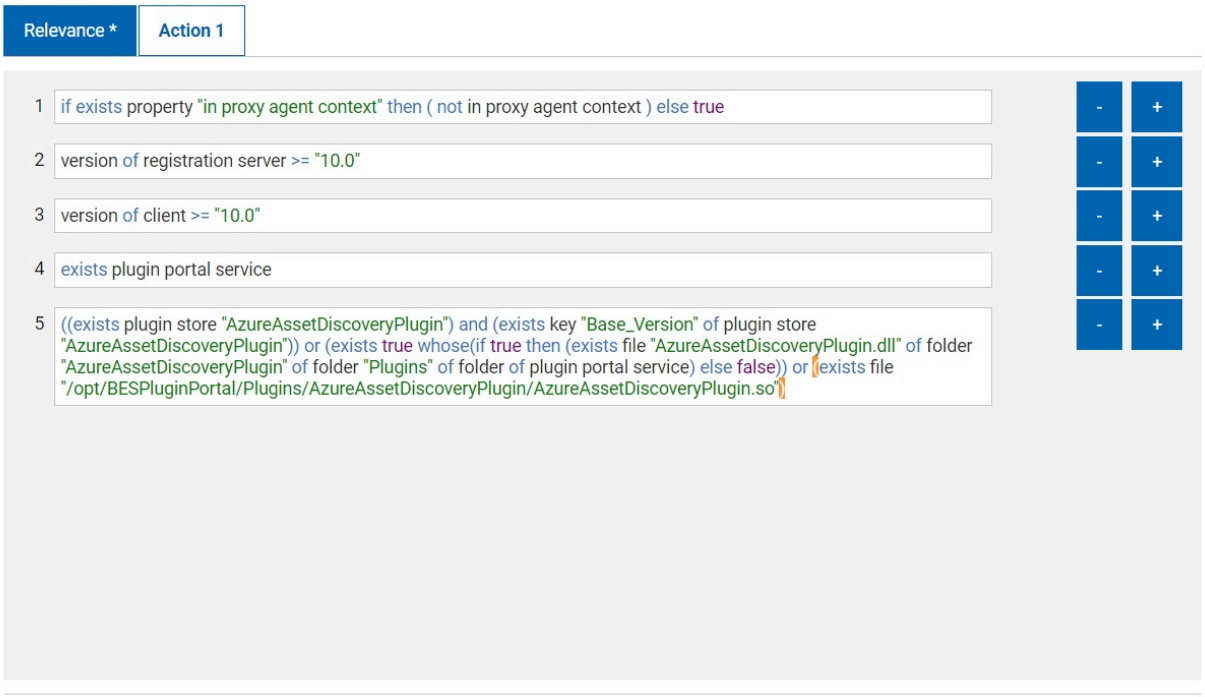
```
if exists property "in proxy agent context" then ( not in proxy agent
 context ) else true

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")
```



Now, similarly to what we did for the AWS Plugin, let us build the action script for the configuration.

Note: The values that you must replace are only the ones wrapped by angled brackets, like <myTenantID1>.

```
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}"
```

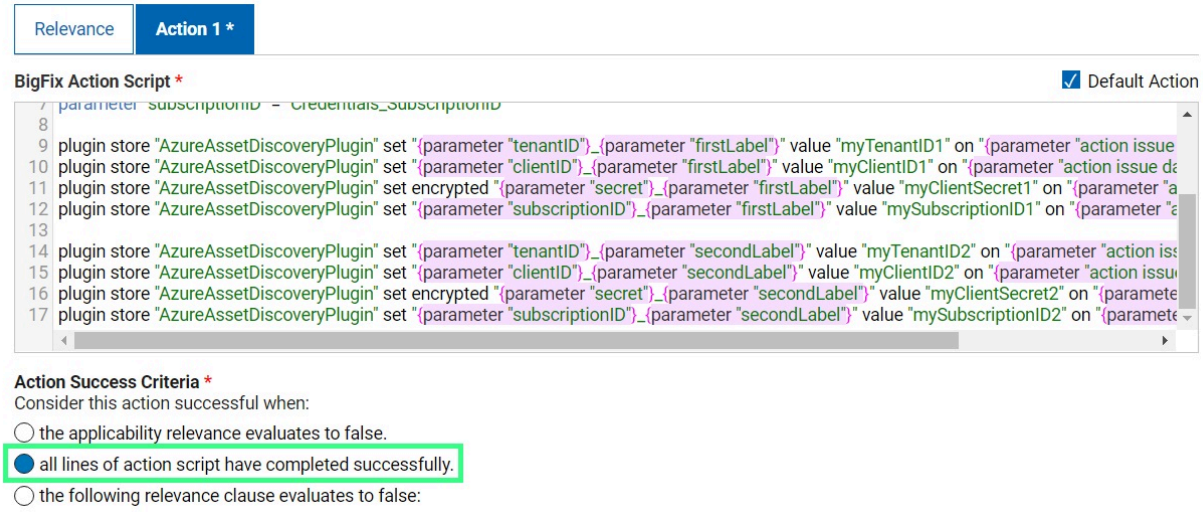
```
plugin store "AzureAssetDiscoveryPlugin" set "{parameter
 "clientID"}_{parameter "firstLabel"}" value "<myClientID1>" on "{parameter
 "action issue date" of action}"
plugin store "AzureAssetDiscoveryPlugin" set encrypted "{parameter
 "secret"}_{parameter "firstLabel"}" value "<myClientSecret1>" on
 "{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}"
```

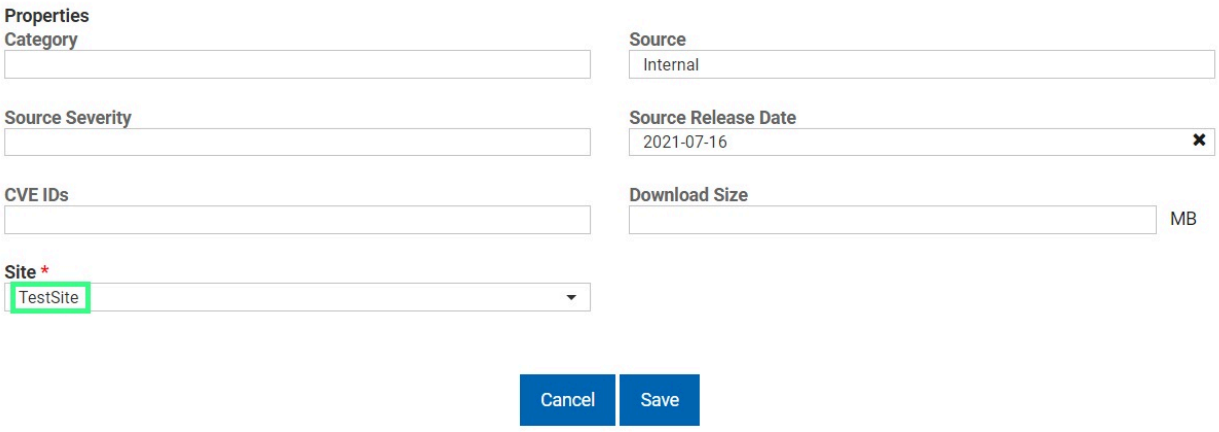


Note: foo and bar are just invented names for our labels. However, **Credentials_TenantID**, **Credentials_ClientID**, **Credentials_ClientSecret** and **Credentials_SubscriptionID** are real setting names. We are defining these four parameters because, by combining them, we can define the keys that we need to set the user key and password like specified above.

Now let us copy and paste the whole script in the Action box, ensuring that we select **all lines of action script have completed successfully**:

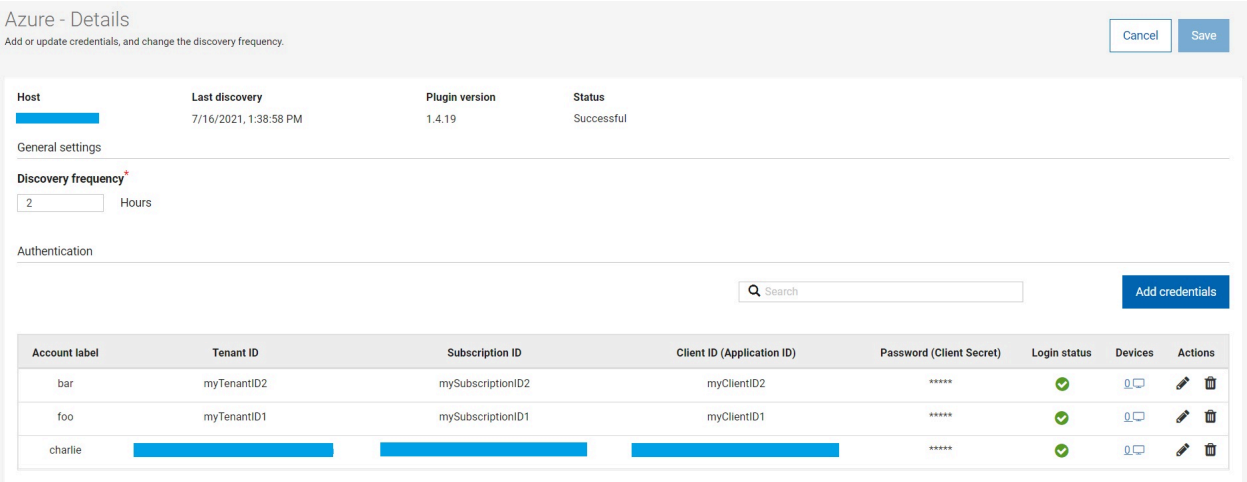


We can now choose the **Site** we want and then click **Save**.



Now we can just go back to the **WebUI > Apps > Custom** application and deploy the newly created Task to configure the Plugin.

After completing this, you should be able to see the new users in the Plugin Management panel:



Example 3: Configuring many settings at once

In this example, we are going to configure many settings at once using the **multiple set** option of the **plugin store** command, in order to quickly dispatch a large configuration for a Plugin.

For more information about the plugin store command and options, see plugin store and Introduction (on page 225).

The syntax for this type of command is the following:

```
plugin store “<plugin name>” multiple set “<percent encoded json>” on
 “<now>”
```

Suppose that we already collected a great number of settings for a Plugin in a JSON file formatted as follows:

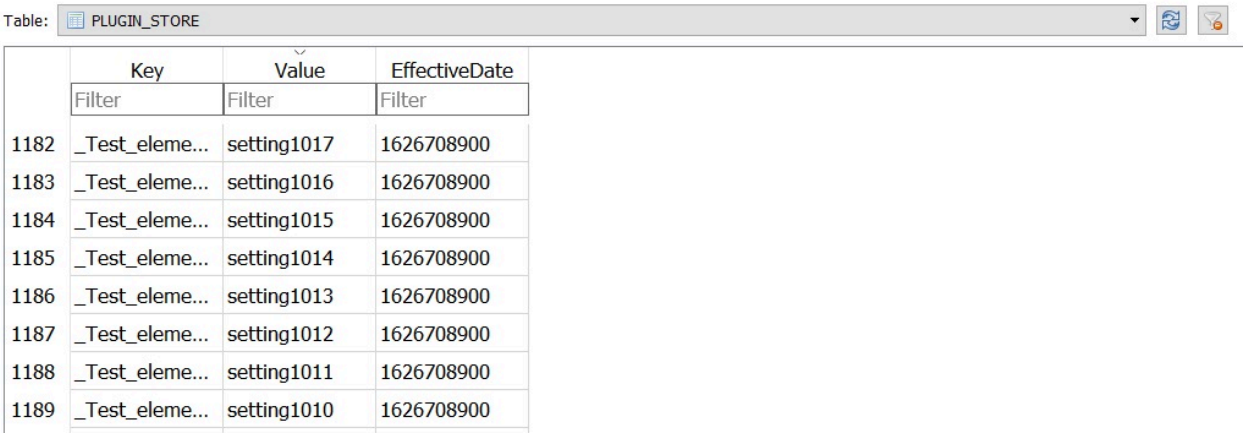
```
{
    “settingKey1”: “settingValue1”,
    “settingKey2”: “settingValue2”,
    “settingKey3”: “settingValue3”,
    …
    “settingKeyN”: “settingValueN”,
}
```

As described in Example 1: Configuring the AWS Plugin (on page 240), we can just create a new action and fill the action script with the commands we desire. In this case, we will issue a multiple set after having copied and pasted the percent encoded JSON in the appropriate section.

```
<?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>
```

Issuing this action through the REST APIs, will result in the addition of all the key-values of the configuration JSON in the database.

As displayed in the image below, hundreds of settings can be quickly added in one single operation.


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

Note that the added settings are in clear text as we have not added the “encrypted” keyword.

This type of approach was meant to be executed from code and, as such, the easier path is creating a dedicated script for encoding the JSON file, creating the action script and then issuing the new action through REST APIs.

Chapter 17. Extending BigFix management capabilities

BigFix 11 delivers a few significant new functions for enhancing the visibility and management of devices on your network regardless of whether the devices are physical or virtual.

Challenges faced in managing modern IT infrastructures

Managing their infrastructures is growing more and more challenging and complex for IT organizations. With the advent of multiple types of servers, different operating systems, software, cloud computing and services, and technology that is changing almost every minute, it becomes difficult to track, control, and manage the IT environments.

- Technologies such as cloud computing and mobility change the IT landscapes fast and it becomes difficult to stay current.
- Catering to new compliance and regulatory requirements while still complying with the old ones has mandated the need for a cost-effective solution.
- As IT organizations continue to increase operations around latest technologies, security becomes a major concern.
- Sophisticated IT infrastructures that support high computing and data analysis need efficient and cost-effective data extraction and data storage techniques.

BigFix 11 features

To achieve transparency across your heterogeneous IT environments, you need a more automated, comprehensive, and robust solution like BigFix 11. This all-new version of BigFix provides you with an accurate view of the resources in your network, key analytics, and detailed insights that can enable your decision makers to make faster and informed decisions about IT management.

Related information

Managing cloud resources *(on page 255)*

Managing cloud resources

The IT ecosystem in every organization keeps expanding to include a variety of endpoints such as servers, Mac and Linux desktops, and laptops. With the advent of cloud computing, more and more IT organizations consider shifting their entire on-premise infrastructure including data centers to the cloud, such as Amazon Web Services, Microsoft Azure, VMware, and Google Cloud Platform (GCP). Consequently, managing endpoints is becoming increasingly complex, which has necessitated endpoint management solutions to also support cross-hosting environments.

BigFix 11 includes capabilities for managing your resources that are in a cloud system as well. With the new feature, BigFix administrators and operators can have complete visibility and control over their Windows, Mac, and Linux endpoints on cloud (public, private, and hybrid) in a secure, cost-controlled manner regardless of their location or whether they are physical or virtual.

The feature extends the capabilities of BigFix to help you discover, manage, and secure your virtual devices (AWS®, VMware, Azure, and GCP to start with) within a single pane of glass with no duplication of information.

- Extends BigFix visibility into all cloud instances across multiple cloud providers including private, public, and hybrid cloud.
- Leverages a combination of cloud native APIs and the BigFix Agent to retrieve information from your virtual endpoints for a complete view of your cloud instances.
- Simplifies cloud management and extends BigFix automation to cloud instances.
- Enables better-informed decisions about cloud instance patching
- Ensures continuous compliance for all your endpoints, regardless of their type or location, within a single solution.
- Helps on-prem and cloud teams work more effectively together.

Understanding cloud setup

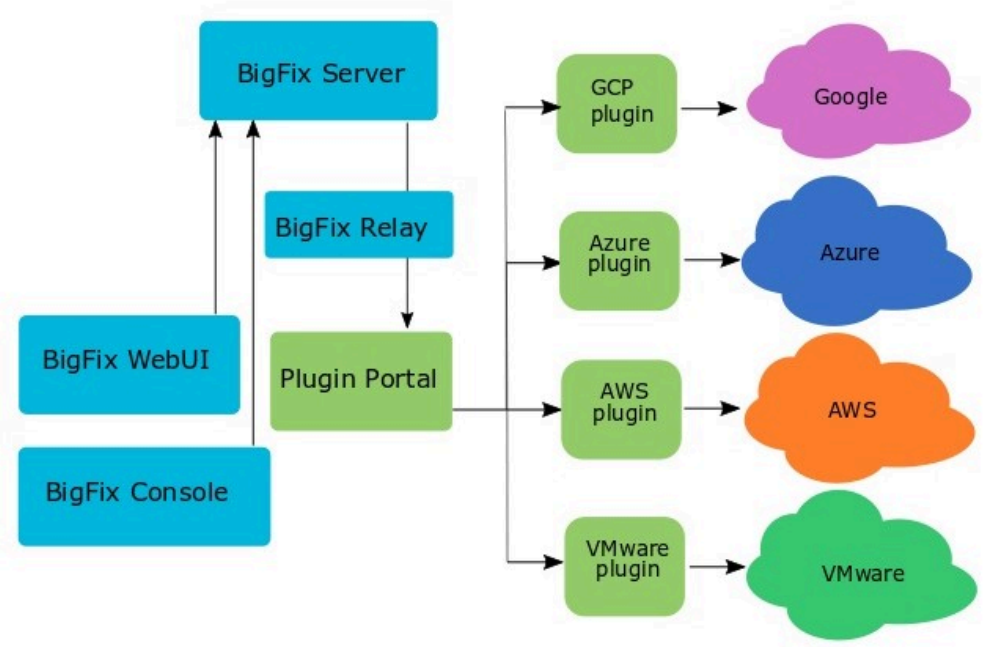
BigFix Console and WebUI provide operators with a single representation of a cloud resource, regardless of whether it is managed by the BigFix Agent, by the The Plugin Portal

(on page 213) through the cloud plugins, or by both. This way, you as an operator can have complete visibility into and control over your cloud infrastructure.

BigFix 11 supports the following use cases for Amazon Web Services (AWS), VMware, Microsoft Azure and Google cloud platforms:

- As a BigFix Master Operator, you can:
  - have all resources under control and monitor what is running on them, with a capability to maintain your environment secure and cost controlled.
  - ensure continuity of service and secure management of your IT enterprise.
  - monitor the cloud instances.
- As a BigFix Operator, you can organize and administer your cloud instances and maintain an inventory of them.
- As an owner of servers that run business-critical applications, you can ensure smooth functioning of your cloud instances.
- As a content creator, you can run analyses to inspect cloud resources and properties without any need to create custom content.

The multicloud management features are available for use in both BigFix Console and WebUI.



In terms of deployment topologies for the Plugin Portal, the following options are available:

- A Portal instance may be deployed geographically remote from the BigFix root server. This may be desirable if there is a large, remote cloud instance that the portal is managing.
- The Plugin Portal may generate large volumes of reports. As such, the higher-level relay infrastructure (if any) should offer high bandwidth. For example, having the Plugin Portal report to a top-level relay, false root relay, or the root server itself is recommended if the Plugin Portal is going to be managing a large number of devices.

Discovering cloud resources

BigFix 11 provides the capability to discover resources that you own on any of the supported cloud providers.

The discovery is performed by the cloud plugins. Each plugin periodically queries the corresponding cloud provider, and retrieves available cloud resource data. Retrieved data is then processed by the Plugin Portal, and eventually sent to the BigFix Server. In case of cloud resources on which the BigFix Agent is installed, the new cloud provider inspector allows retrieving data that is used for computer correlation (on page 259) purposes.

In BigFix Version 10.0.8, a new feature was introduced to delete on the BigFix Database the computer instances no longer discovered. When during a discovery, the Plugin recognizes that a device is no longer present, it sends a report to the Plugin Portal, that removes the device from its database and sends a removal HTTPs Post Request to the Server which will perform either a physical deletion from the database or a logical deletion.

By default, the BigFix Server performs a logical deletion: all the data still remain in the database but the devices are marked as deleted. Devices no longer discovered due to an error, for instance due to wrong credentials, are not removed.

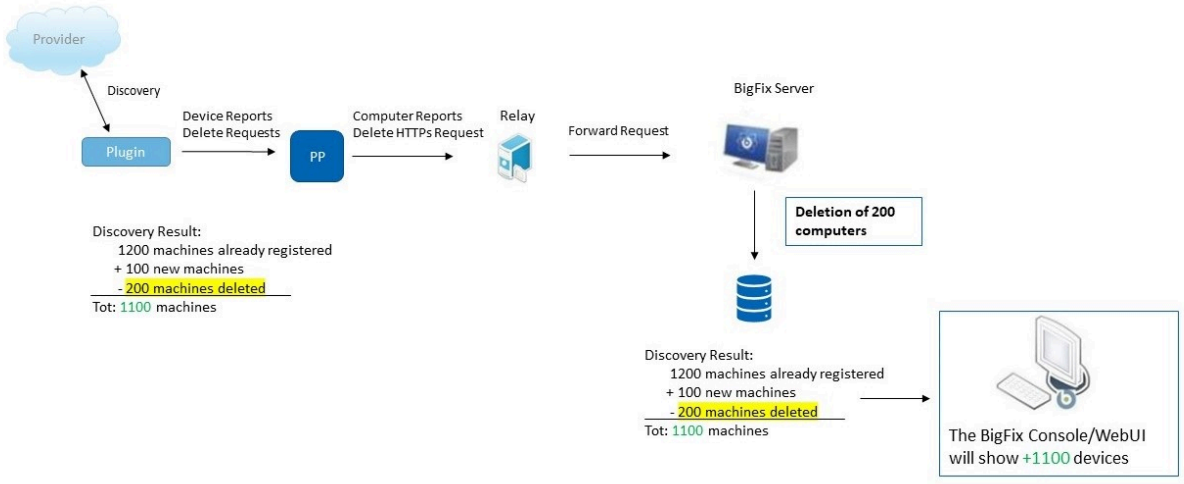
A new setting was added to the Plugins, allowing you to perform the physical deletion from the database. With this option, the BigFix Server physically removes the devices and all its associated data from the database, as the BES Computer Remover Tool does.

To avoid database inconsistencies, we do not support physical delete operations in DSA environments. If the server receiving the physical delete request is within a DSA environment, it logs that the physical delete operations are not supported in DSA environments and switches to a logical delete operation.

Another setting was added to the Plugins, allowing also the deletion of all the correlated instances. Both these settings can be changed from the WebUI. For more details about the settings, see Configuring cloud plugins (on page 279).

In the following example, the Plugin discovers 100 new machines and no longer detects another 200, so the Plugin Portal receives a new report for the 100 new machines and an empty report for the other 200 machines. For its part, the Plugin Portal must register new devices and remove undetected devices (Device alerts / Deletion requests arrow). With this new feature, the Plugin Portal will also send an HTTP delete request for the devices that are no longer detected (Delete HTTPs request arrow). The Relays will forward the request to the BigFix Server, which is in charge of performing either a physical or a logical deletion (depending on the new plugin setting introduced and already described above) of the 200

undetected devices from the BigFix database. What happens is that the removed devices will not be displayed in the BigFix Console and in the WebUI computer list and the BigFix Console and the WebUI will always show the list of the actually detected devices.



Related information

The Plugin Portal *(on page 213)*

Understanding cloud setup *(on page 255)*

Correlated devices

BigFix 11 has the capability of correlating multiple representations of the same computer, allowing operators to manage them as a single entity (also referred to as correlated device throughout this documentation) as well as to operate on a specific representation as needed.

For instance, if the Microsoft Azure cloud plugin discovers a VM created on Microsoft Azure, and at the same time the discovered VM runs the BigFix Agent, then two separate representations of the same computer is reported to the BigFix Server. In this case, BigFix 11 correlates the two representations, and the BigFix Console shows them in a grouped, expandable format.

In this documentation, the term *proxied* refers to computer representations discovered by cloud plugins, and the term *native* refers to computer representations associated to BigFix Agents.

Note: The correlated device requires BigFix Agent version 10 or higher.

Note: Up to BigFix Platform Version 10.0.8, the data required for the correlation in Amazon Web Services are retrieved by the BigFix Agent using Amazon IMDSv1 protocol. Starting from BigFix Platform Version 10.0.9, these data are retrieved using Amazon IMDSv2 protocol.

Note: Starting from BigFix Platform Version 11.0.4, the correlated device can contain up to a maximum of 32 representations.

Enabling device correlation for a cloud plugin

When a cloud plugin is installed, the correlation feature for devices discovered by that plugin is automatically activated.

Displaying correlated devices

The correlated device is a logical entity, and the BigFix Console shows it as an expandable object in the **Computers** view. The root element of the object represents the correlated device itself, and has an own Computer ID. The ID of the correlated device is created at correlation time, and typically falls in a range higher than the IDs of the single representations. Correlated representations are shown slightly indented when expanding the correlated device.

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

The correlated device inherits the properties from all the devices it correlates. In case the devices report different values for the same property, the correlated device inherits the value from the native one as it is the most accurate and meaningful source of data.

Note:

- When one of the correlated representations has not been checking in for the amount of time specified in the BigFix Console preferences, besides appearing itself as offline, it makes the *correlated device* appear offline as well.
- The Agent Version associated with proxied computers corresponds to the version of the Plugin Portal that is managing them.
- The Computer Name of proxied computers discovered by the AWS plugin corresponds to the name of the host taken from the Private DNS Name.

Administering correlated devices

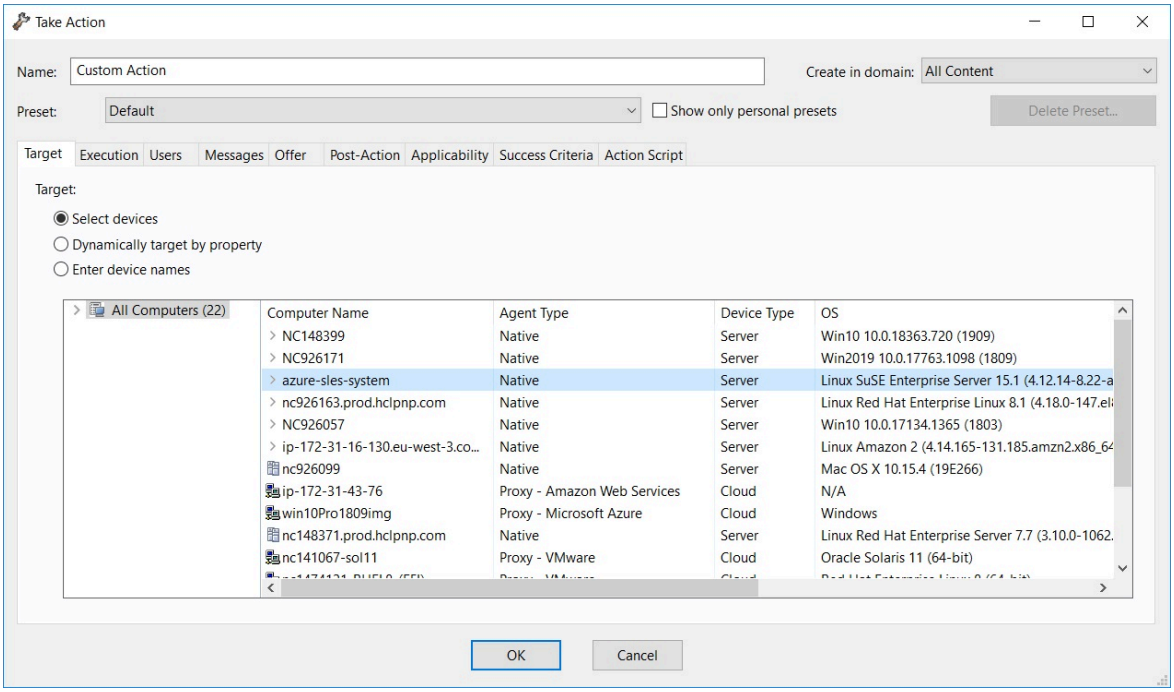
Correlated devices are visible to Master Operators and to Non-Master Operators who administer two or more correlated representations. Operators can administer the correlated

representations independently of one another. No inheritance mechanisms propagate administration rights from a correlated representation to another one.

Performing operations on correlated devices

Operators can target correlated devices, and depending on the type of operation, the BigFix Server takes care of dispatching it to the proper correlated representations. Following are a few examples:

- As shown in the following figure, an Operator takes a custom action, and in the **Target** tab, clicks **Select devices** and selects a *correlated device*:



In this case, the BigFix Server dispatches the action to one of the correlated representations based on the parsing of the actionscript commands. If only one of the correlated representations can run all the commands of the actionscript, then the BigFix Server dispatches the action to that representation. If the whole actionscript is applicable to more than one representation, then the BigFix Server always chooses to dispatch the action to the native computer.

Note:



- If the Operator wants to target a specific computer representation, it is enough to expand the *correlated device* in the **Target** tab, and select the desired representation. In this case, the BigFix Server sends the action to that representation directly.
- If the action is taken from a Fixlet that is applicable to only one of the representations (in this scenario, you can verify it by expanding the *correlated device* in the **Target** tab), then the action is sent to that representation.

- Operator runs a BigFix query and targets a *correlated device*. In this case, the BigFix query is always sent to the native computer because the BigFix Agent is the only component that can execute a BigFix query.
- In the **Computers** view, an operator selects a *correlated device* and adds it to a manual computer group. In this case, all the correlated representations are added to the manual computer group.
- In the **Computers** view, an operator selects a *correlated device* and opts for **Remove From Database...**. In this case, all the correlated representations are set to deleted, and stop being displayed by the BigFix Console both as correlated and standalone computers.
- In the Computers view, an operator selects a *correlated device* and runs a **Send Refresh**. In this case, the refresh notification is sent to all the correlated representations.



Note: Any client relevance expressions referring to the IDs of *correlated device* are not matched by any native or proxied computers, because those IDs represent logical entities that are only known to the BigFix Server. For instance, referring to the IDs of *correlated devices* for defining an automatic computer group or for subscribing computers to a site does not cause any computers to be included in the computer group or be subscribed to the site.

Using REST APIs with correlated devices

While remaining compatible with the XML schema definition of previous BigFix releases, REST APIs in BigFix 11 can handle correlated devices supporting them with methods and applicable resources.

For instance, the ID of a correlated device may be used as target of an action, and in this case the BigFix Server takes care of dispatching the action to the proper target based on the operator permissions and on the commands contained in the actionscript.

Similarly, the ID of a correlated device may be used to retrieve information from the BigFix Server. For instance, if an operator wants to retrieve the settings of a correlated device, the REST API returns an XML made up of a main section with the settings of the native computer, and a subsection named `ManagementExtension` that contains the settings of the proxied computer. For further details, see Computer REST APIs.

Using Session Relevance with correlated devices

When using session relevance, the inspectors `bes computers` and `bes computers set`, in case of correlated devices, will return **only** the BES Computer object related to the CorrelationID. Asking for the value of Properties or applicability of Fixlets on a BES Computer representing a Correlated device will be the same as querying its representations (Native and Azure for example) in order of priority and returning the first available.

Two new session relevance inspectors were introduced, `bes computers with extensions` and `bes computers with extensions set`, where the set of computers returned will include **both** the BES Computer representing the correlated device and its representations.

To check if a BES Computer object represents a correlated device or a representation of a correlated device, two new properties were added to the BES Computer object:

- `correlation flag of <bes_computer> : boolean` returns true for a BES Computer object representing a correlated device.
- `extension flag of <bes_computer> : boolean` returns true for a BES Computer object representing an extension of a correlated device.

Two properties were also added to query an extension for its correlated representation:

- `correlation of <bes_computer> : bes_computer` to return a BES Computer object of the correlation device from an extension.
- `correlation id of <bes_computer> : integer` to return only the Computer ID of the correlation device from an extension.

For example a relevance such as:

```
(name of it, agent type of it, correlation id of it) of bes computer with
 extensions
whose (extension flag of it)
```

returns for all devices that partecipate in a correlated device, their name, their agent type and the ID of the correlated device.
With a deployment like the one in the screenshot of the **Displaying correlated devices** section described above, it would return the following tuples:

```
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
```

For further details, see Computer Inspectors.

Deleting a correlated representation

If, for instance, a correlated device correlates a native and a proxied representation, and at some point one of the two representations is set to deleted (either manually through the BigFix Console, by the BigFix Computer Remover tool, or through the BigFix WebUI or automatically by the BigFix Server, when a resource is no longer discovered by the Cloud Plugin, for details see Discovering cloud resources (on page 257)), the correlated device

is set to deleted as well, and is no longer displayed by the BigFix Console. The remaining representation goes back to being displayed as a standalone computer.

When the BigFix Computer Remover tool removes the deleted representation from the database, the correlated device is removed as well.

Introduction to Cloud Plugins

Starting from Version 10, BigFix provides a set of Cloud Plugins that can be attached to a Plugin Portal, in order to manage the Cloud environments of Amazon Web Services, Microsoft Azure, Google Cloud Platform and VMware. Each plugin has a set of similar capabilities but each of them differs in the support of some features, due to the nature of the correspondent Cloud Provider.

AWS Asset Discovery Plugin

The AWS Asset Discovery Plugin is able to discover and report data about Amazon Web Services EC2 instances. On top of that, the plugin is also able to deploy the BigFix Agent on the whole AWS cloud environment.

Configuration

As of now, the AWS Asset Discovery Plugin supports IAM Users and Roles in its configuration. Single and cross-account scenarios are supported, in line with AWS best practices.

For more information, see Installing cloud plugins (on page 271) and Action commands for configuring the BES Plugin Portal plugins (on page 225).

BigFix Agent Installation

Starting from Version 10 Patch 2, the AWS Plugin supports the deployment of the BigFix Agent through native APIs.

For more information, see BigFix Agent installation on cloud resources (on page 353).

Inspectors

The AWS Plugin reports several properties of the EC2 instances it discovers. **Starting from Patch 2**, several other AWS System Manager properties are available upon further

configuration . It is also possible to discern which set of credentials reported the information, since each instance also reports a credentials label.

For more information, see AWS Asset Discovery Plugin Inspectors (on page 289).

Analyses

Analyses are available for each plugin. See Activating cloud analyses (on page 346) for more information.

Data concerning resources is described in The cloud analyses data (on page 346).

Azure Asset Discovery Plugin

The Azure Asset Discovery Plugin is able to discover and report data about Microsoft Azure Virtual Machines. On top of that, the plugin is also able to deploy the BigFix Agent on the whole Azure cloud environment.

Configuration

The Azure Asset Discovery Plugin requires **Client ID**, **Password**, **Subscription ID** and **Tenant ID** to be configured.

For more information, see Installing cloud plugins (on page 271) and Action commands for configuring the BES Plugin Portal plugins (on page 225).

BigFix Agent Installation

Starting from Version 10 Patch 2, the Azure Plugin supports the deployment of the BigFix Agent through native APIs.

For more information, see BigFix Agent installation on cloud resources (on page 353).

Inspectors

The Azure Plugin reports several properties of the instances it discovers. It is also possible to discern which set of credentials reported the information, since each instance also reports a credentials label.

For more information, see Azure Asset Discovery Plugin Inspectors (on page 300).

Analyses

Analyses are available for each plugin. See Activating cloud analyses (on page 346) for more information.

Data concerning resources is described in The cloud analyses data (on page 346).

GCP Asset Discovery Plugin

Starting from Version 10 Patch 2, the GCP Asset Discovery Plugin is available for installation. The plugin is able to discover and report data about GCP Compute Engine instances.

Configuration

The GCP Asset Discovery Plugin requires a credentials JSON to be configured.

Starting from Version 10 Patch 4, multi-project discovery is also available.

For more information, see Installing cloud plugins (on page 271) and Action commands for configuring the BES Plugin Portal plugins (on page 225).

Inspectors

The GCP Plugin reports several properties of the instances it discovers. It is also possible to discern which set of credentials reported the information, since each instance also reports a credentials label.

For more information, see GCP Asset Discovery Plugin Inspectors (on page 306).

Analyses

Analyses are available for each plugin. See Activating cloud analyses (on page 346) for more information.

Data concerning resources is described in The cloud analyses data (on page 346).

VMware Asset Discovery Plugin

The VMware Asset Discovery is able to discover and report data about VMware Guest Virtual Machines.

Starting from Version 11 Patch 2, VMware Hosts will be available for discovery as well. Furthermore, the VMware Plugin will report a greater number of inspectors and properties for both Guests and Hosts while also allowing the user to take advantage of several action

commands. The capabilities of the VMware Plugin will match the ones of the old ESXi management extender.

Configuration

The VMware Asset Discovery Plugin requires a **User name** and **password**, along with the **vCenter Server URL**, to be configured.

For more information, see Installing cloud plugins (on page 271), Configuring cloud plugins (on page 279) and Action commands for configuring the BES Plugin Portal plugins (on page 225).

Action Commands

It is recommended to rely on the Patch for VMware ESXi, Patch for Virtual Endpoint Manager and Server Automation sites content to use this action commands instead of creating custom actions.

For more information, see VMware Plugin Commands (on page 341).

Inspectors

The VMware Plugin reports several properties of VMware Virtual Machines. **Starting from Version 11 Patch 2**, new inspectors and properties are available for VMware Hosts as well. It is possible to discern which set of credentials reported the information, since each instance also reports a credentials label.

For more information, see VMware Asset Discovery Plugin Inspectors (on page 313).

Analyses

Analyses are available for each plugin. See Activating cloud analyses (on page 346) for more information.

Data concerning resources is described in The cloud analyses data (on page 346).

Managing cloud plugins

BigFix 11 Platform includes a plugin for every cloud provider supported, namely Amazon Web Services (AWS), Microsoft Azure, VMware and Google Cloud Platform (GCP). Each of these cloud providers has its own uniqueness, capabilities, and ways to interface with an external program and they handle access to data and capabilities differently.

To be able to install the plugin portal and the cloud plugins, Master Operator (MO) privilege is required.

The multicloud management features are available for use in both BigFix Console and WebUI.

Related information

Installing the plugin portal

Installing cloud plugins

Planning the installation of cloud plugins

Cloud plugins can be installed on computers running the Plugin Portal (on page 213). Optionally, they may all be installed on the same Plugin Portal instance.

However, this option should be taken into account only if it meets the capacity planning requirements of the Plugin Portal that are included in the BigFix Capacity Planning guide.

It is strongly recommended to have only one instance of cloud plugin per type (AWS, Microsoft Azure, VMware , GCP) in a BigFix deployment. The purpose of this recommendation is to avoid having the same cloud resources discovered more than once, which would cause the correlation of multiple instances of the same proxied computer under one correlation computer, adding unnecessary complexity and load to the displaying and management of correlation computers.

While requiring specific configuration information that depends on the supported cloud platform, the installation of every cloud plugin requires the input of credentials that would allow the embedded SDK to call the APIs of the corresponding cloud platform and access related cloud services. As all cloud plugins perform a credential-based discovery, their capability to discover virtual machines and retrieve related data depends on the permissions granted on the cloud platform side to the user owning the credentials. More information about the required permissions are available in Installing cloud plugins (on page 271).

Note: Each cloud plugin supports multiple sets of credentials. However, only one set may be specified at installation time. Any further credential sets can be added later on using the BigFix WebUI.

When installing the AWS, Microsoft Azure and the Google Cloud plugins, you must ensure that these plugins are able to access related cloud services using HTTPS over the Internet.

The cloud plugins are installed in the following default paths:

Windows:

- C:\Program Files (x86)\BigFix Enterprise\BES Plugin Portal\Plugins*Plugin_name*

Non-windows:

- /opt/BESPluginPortal/Plugins/*Plugin_name*
- /var/opt/BESPluginPortal/Plugins/*Plugin_name*

Installing cloud plugins

Each cloud plugin has a specific installation task on BES Support, which becomes relevant on computers where the The Plugin Portal (on page 213) is installed.

Note: The last published version of the cloud plugins might require an up-to-date version of the Plugin Portal. Older plugins will continue to work properly with the new Plugin Portal.

The task can be run only after filling in all required fields in the **Description** tab.

Only Master Operators (MO) are allowed to install cloud plugins.

The advanced configuration for all cloud plugins can be done using the WebUI.

Amazon Web Services plugin

Account Label

A friendly name for the specified `Access Key ID` / `Secret Access Key` pair. It must contain only alphanumeric characters.

Default region name

It is the name of the AWS region to which the plugin must initially connect to when it performs a discovery. For instance, if you want the plugin to start its discoveries connecting to the Europe (Frankfurt) region, the value to specify is `eu-central-1`. The plugin will, then, automatically complete its discoveries by connecting to all other regions that the specified `Access Key ID` / `Secret Access Key` pair can access.

Note:

- The field is case sensitive, ensure that you input the string with the correct case as documented by AWS.
- When adding a new `Access Key ID` / `Secret Access Key` pair, the BigFix WebUI allows to optionally specify a *user region* value, which for the specified key pair would prevail on the default region.

For more information about available regions, refer to the AWS documentation.

When installing the AWS plugin, you can specify the allowed regions. For more details about how to limit the AWS regions, see Limit AWS Regions to restrict the scope of device discovery.

Access Key ID and Secret Access Key

An `Access Key ID` / `Secret Access Key` pair associated to an IAM user.

Requirements for the IAM user:

- MFA must NOT be enabled
- Must have programmatic access type
- Must have the following permissions at minimum: action "**ec2:Describe***" allowed on resource "***"
    - A suitable predefined AWS policy is *AmazonEC2ReadOnlyAccess*

For more information about AWS access keys, refer to the AWS documentation.

IAM Roles

An ARN / Region / External ID triple associated to an IAM Role.

Starting from cloud plugin version 1.4, released concurrently with BigFix v10.0 Patch 4, IAM roles are supported. An IAM role is an identity that has a set of assigned permissions and it can be assumed temporarily by any trusted user, including an administrative user, depending on your business needs. Roles do not have credentials and as such they are not subject to password expiration. When assuming a role, the logged on user requests temporary credentials for a certain limited amount of time which cannot be bigger than the maximum amount of time assigned by the administrative user.

Note: If you decide to use IAM roles, ensure that the users assuming the role are authorized to perform `sts:AssumeRole` on the roles.

Note: The roles completely replace the users assuming them, which means that each operation managed by the users is performed by the roles and that the roles must have the same permissions which would be required for a user managing the cloud plugins.

Note: Once AWS Roles are inserted, the AWS plugin will use them during its discovery, instead of the credential from which they derive. You must ensure that these roles include all the AWS devices that you want to discover in your cloud environment: otherwise, some machines may not be discovered.

You have to specify the following information:

An `ARN` / `Region` / `External ID` triple associated to an IAM Role.

Where:

ARN

Is the Amazon Resource Name of the role, which is the unique identifier of a resource on AWS.

For more information about ARNs, refer to the AWS documentation.

Region

(Optional): Is the default AWS region for the IAM role. See the **Default region name** section for more information. When adding a new IAM role, BigFix allows you to optionally specify a *role region* value, which prevails on both the default region and the user region for the specified role.

External ID

(Optional): If you need to delegate access to AWS resources to a third party, an IAM role can be used along with an external ID, devised for the purpose of accessing and using the cloud environment resources and services by the third party.
The **external ID** must be provided to the third party by the organization that owns the environment and should be a **GUID**.

For more information about external IDs, refer to the AWS documentation.

HTTP Proxy

Optionally, an HTTP proxy may be specified in case the system where the AWS plugin will be installed does not have a direct connection to the Internet. For supported proxy authentication methods, refer to the AWS documentation.

Microsoft Azure plugin

Account Label

A friendly name for the specified service principal quartet. It must contain only alphanumeric characters.

Client ID, Password, Subscription ID and Tenant ID

A service principal quartet. Requirements for the service principal:

- Must be assigned the built-in *Reader* role.
- MFA must NOT be enabled.

For more information about Microsoft Azure service principals, refer to the Microsoft Azure documentation.

Note: If your Azure credential needs to discover resources located in an Azure Cloud environment different from the public one, see the "Azure cloud plugin environment" section in Configuring cloud plugins *(on page 279)*.

VMware plugin

Account permissions

For the VMware cloud plugin to work correctly, ensure that the following permissions are granted to the users, according to your business needs:

Datastore

- Browse datastore

Global

- Diagnostics
- Global tag

Host

- Configuration
  - Image configuration

vSphere Tagging

- Assign or Unassign vSphere Tag on Object

Virtual machine

- Change Configuration
  - Modify device settings
- Interaction
  - Install VMware Tools
  - Power off
  - Power on
  - Reset
  - Suspend
- Provisioning
  - Clone template
  - Clone virtual machine
  - Create template from virtual machine
  - Deploy template
- Snapshot management
  - Create snapshot
  - Remove snapshot
  - Rename snapshot
  - Revert to snapshot

Account Label

A friendly name for the specified username-password pair. It must contain only alphanumeric characters.

vCenter Server

The hostname or IP address of the vCenter server.

User name and Password

The credentials to access the vCenter server.

Certificate full path

(Optional) The full path of the custom certificate. You can specify if the authentication to the vCenter Server requires a certificate. You can specify the certificate full path on the system where you intend to install the plugin. For example, `C:\Users\Administrator\Downloads\download\certs\win\certificate.0.crt` on Windows systems or `/Downloads/download/certs/lin/certificate.0.crt` on Linux systems.

Note: The VMware plugin uses the govmomi library and its compatibility defines with which version of vCenter the plugin is compliant. For more details, refer to the govmomi documentation.

Google Cloud Platform plugin

Account Label

A friendly name for the specified Service Account credentials. It must contain only alphanumeric characters.

Service Account Credentials

Copy and paste the content of the .json file provided by Google containing the keys of your Service Account. The IAM permissions required are:

- compute.zones.list
- compute.regions.list
- compute.instances.list
- compute.images.list
- compute.disks.list
- compute.machineTypes.list
- compute.subnetworks.list

All these permissions are mandatory.

For more information about Google Cloud Platform service accounts, refer to the Google documentation.

Note: The Google Cloud Platform plugin will discover the VM instances belonging to the project(s) specified in the .json file(s) with which the plugin was configured. Starting from version 1.4 of the Google Cloud Platform plugin, you can manage all additional project(s) that your Service Account is able to list and has permissions on. To be able to list projects, the Resource Manager API must be enabled and the Service Account must be able to issue requests to it, which means having the *resourcemanager.projects.get* permission. For more information about listing projects, refer to the Google documentation.

Related information

The Plugin Portal *(on page 213)*

Understanding cloud setup *(on page 255)*

Verifying the cloud plugin installation

Once the deployment of a cloud plugin has completed, its first discovery attempt is expected to start immediately and the discovered resources should be displayed in the **Computers** view of the BigFix Console or in the **Devices** page of the BigFix WebUI within a few minutes.

As a post-installation sanity check, you can verify the following:

1. The existence of the cloud plugin log file in the following default path:
   - **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. The content of the cloud plugin log file which for a functioning plugin just installed will be such as:

```
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. The running status of the BESPluginPortal process.
4. The content of the plugin portal log file, located in the following default path:
   - **Windows**: C:\Program Files (x86)\BigFix Enterprise\BES Plugin Portal \BESPluginPortal.log
   - **Linux**: /var/log/BESPluginPortal.log

   After the latest plugin portal restart, the log file should contain a line such as:

```
Tue, 31 Mar 2020 19:33:49 +0200 - 1222616832 - Running
 plugin'<Plugin_name>'
```



Note: The presence of the following message in the plugin portal log file is expected as long as no discovered resources exist:

```
Tue, 31 Mar 2020 19:33:49 +0200 - 2383846272 - Error reading
 records from the database.
```

In case of unexpected results, refer to the Troubleshooting (on page 364) section.

Configuring cloud plugins

Cloud plugins are installed with a base configuration that can be updated later on, primarily through the BigFix WebUI.

VMware cloud plugin secure connection

Starting from BigFix Version 11.0.4, you can install the vCenter certificates, on the system where you intend to install the VMware cloud plugin, in order to automatically open secure connections, by performing the following steps:

1. Download the trusted root certificates from the vCenter Server system or vCenter Server Appliance. You can set the URL to <*vCenter URL*>/certs/download.zip
2. The extension of the downloaded file must be .zip (The file is a ZIP file containing all certificates in the TRUSTED_ROOTS store, which is the site of the vCenter Server system or vCenter Server Appliance you are connecting to.
3. Extract the ZIP file.
4. The extracted folder includes files with the extension .0. .1, and so on, which are certificates, and files with the extension .r0, r1, and so on, which are Certificate Revocation List (CRL) files associated with the certificates.
5. The full path of the trusted root certificate must be specified in the Fixlet 5424 - Install BigFix Plugin for VMware Discovery in the **Certificate full path** field. The full path format to be inserted is `<fullPath>/<file.extension>` , an example of full path to insert is: `C:/Users/Administrator/Desktop/certificate/1763543783.0`

For more details about the certificates download and installation, see Download and install vCenter Server root certificates to avoid web browser certificate warning.

For more details about the managed trusted certificates, see Manage Trusted Root Certificates.

AWS cloud plugin FIPS mode connection

Starting from BigFix Version 11.0.3, you can configure the Amazon Web Services (AWS) cloud plugin to leverage the FIPS mode connection.

The Federal Information Processing Standard (FIPS) 140-3 publication is a US and Canadian government standard that defines security requirements for the encryption of modules that protect sensitive information.

The FIPS connection is available for the following US and Canadian AWS regions:

- us-east-1
- us-east-2
- us-west-1
- us-west-2
- us-gov-east-1

- us-gov-west-1
- ca-central-1
- ca-west-1

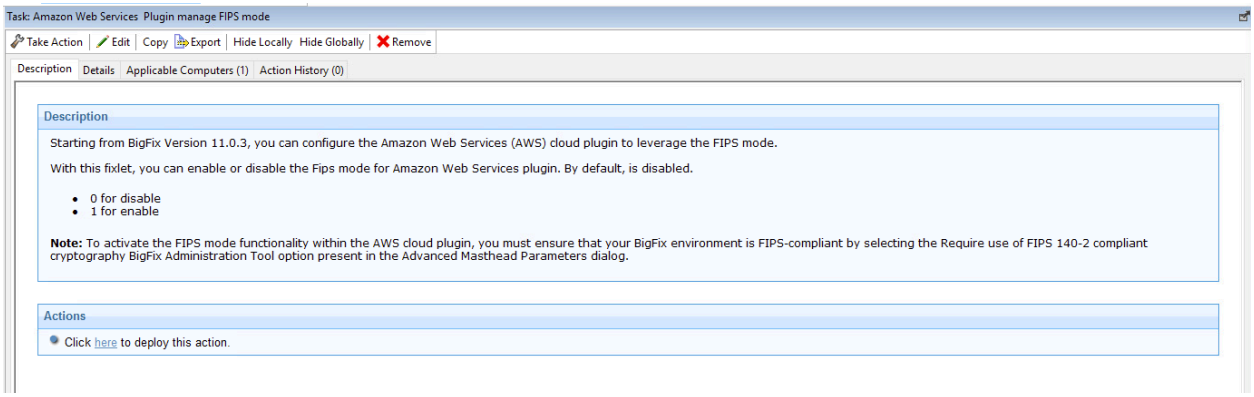
To activate the FIPS connection functionality within the AWS cloud plugin, you must ensure that your BigFix environment is FIPS-compliant by selecting the **Require use of FIPS 140-2 compliant cryptography** BigFix Administration Tool option present in the **Advanced Masthead Parameters** dialog.

After selecting this option, it allows the adoption of FIPS encryption standards across all BigFix components.

The Plugin Portal is configured to use the FIPS encryption, and this configuration is then passed to the AWS cloud plugin.

The FIPS connection can be enabled by using the **Discovery_UseFipsMode** setting, in the **PluginStore.db** configuration file stored in the plugin portal installation folder.

The default setting value is 0 (disabled). You can enable the setting value by running the **Amazon Web Services Plugin manage FIPS mode** Task and specifying 1 as value.



Note: If this setting is enabled but the AWS Region does not support FIPS encryption, the AWS cloud plugin API request fails.

Azure cloud plugin environment

If your Azure credential needs to discover resources located in an Azure Cloud environment different from the public one (the default), such as the US Government or China, you have to configure the Azure environment for this credential. After you have inserted the credential, you may set the environment using the plugin store command:

```
parameter "myLabel" = "<my label>"
plugin store "AzureAssetDiscoveryPlugin" set "Environment_{parameter
 "myLabel"}"
value "<myEnv>" on "{parameter "action issue date" of action}"
```

where <myEnv> is one of the admitted environments by Azure, targeting the Plugin Portal where the Azure cloud plugin is installed.

Take into account that each credential can only have one environment associated with it, and that it will be able to work only there.You can retrieve the allowed Azure environment values using Azure CLI (az cloud) with this command:

```
az cloud list --output table
```

Discovery frequency

It is the frequency with which a cloud plugin runs the discovery of the related cloud resources. Cloud plugins are installed with a default discovery frequency of 120 minutes.

The discovery frequency can be updated from the **Plugin Management** section of the BigFix WebUI. There are also BES Support tasks that allow updating the discovery frequency of cloud plugins.

Discovery optimization

Starting from BigFix Version 11.0.2, the total discovery duration of time was optimized by using the multi-threading solution present in the cloud plugins.

To leverage the multi-threading solution, you can enable the following cloud plugin settings:

- _<*PluginName*>_Discovery_MaxConcurrentRoutinesForCredentials (available for all plugins)
  - This setting specifies the maximum amount of concurrent routines that you can use for a given cloud plugin credentials.
  - The allowed value range is: 1 (minimum value) - 200.000 (maximum value).
  - The default value is 10.
- _<*PluginName*>AssetDiscoveryPlugin_Discovery_MaxConcurrentRoutinesForScans (available for VMware and Amazon Web Services plugins)
  - This setting specifies the maximum amount of concurrent routines for a given cloud plugin discovery.
  - The allowed value range is: 1 (minimum value) - 200.000 (maximum value).
  - The default value is 10.

These settings can be specified using the Fixlet under BES Support named <PluginName>**multithreading discovery settings**.

Remove data for Deleted Devices

This option, available starting from BigFix Version 10.0.8, consists of a setting on the Plugins that can be changed from the WebUI. When the Plugin Portal sends a deletion request to the BigFix Server, and this setting is enabled, the BigFix Server performs a physical deletion. It means that it physically removes the devices and all associated data from the database. If this setting is disabled (it is disabled by default), the BigFix Server performs a logical deletion. It means that the devices are marked as deleted but they are still contained in the BigFix Database.

For details about this feature, see Discovering cloud resources (on page 257).

Remove Correlated Instances

This option, available starting from BigFix Version 10.0.8, consists of a setting on the Plugins that can be changed from the WebUI. When the Plugin Portal sends a deletion request to the BigFix Server, and this setting is enabled, the BigFix Server deletes also the native instances associated to the Proxied ones, coming from the Plugin Portal deletion request. This setting is disabled by default.

For details about this feature, see Discovering cloud resources (on page 257).

Logs

Cloud plugins are installed with a standard log level, and the following default log path:

- **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

The log file rotates when it reaches the size of 10 MB. The latest 10 rotated logs are available in the log directory.

You can update the log path and verbosity in the **Plugin Management** section of the BigFix WebUI.

Support of multiple sets of credentials

A cloud plugin might be configured to use multiple sets of credentials. The first set is specified at installation time, further sets can be added at any time from the **Plugin Management** section of the BigFix WebUI.

At each discovery, the cloud plugin goes through all available credential sets, and retrieves the cloud resources that are available to each set. If a credential set reaches the maximum number of failed discovery attempts, it is no longer taken into account during the next discoveries.

Maximum number of failed discovery attempts

It is the maximum number of consecutive discovery failures for a credential set before it is skipped. Not all kind of discovery failures would increase the counter, only those related to failed login attempts due to incorrect or expired passwords. Successful discovery attempts reset the counter. Restarting the plugin portal would reset the counter as well.

Skipping a failing credential set after three consecutive failures helps reducing the chance of having it affected by possible account lockout policies in place on the cloud platform side.

When a credential set reaches the maximum number of failed attempts, the following message is written in the standard log:

```
[error] Refresh all: user 'Account Label' reached the maximum attempts (3)
 and it will be skipped
```

When this happens, the operator should take advantage of the BigFix WebUI to update the password. The updated credential set is taken into account again starting from the next discovery.

Proxy support

The AWS and the Microsoft Azure plugins must be able to access related cloud services using HTTPS over the Internet. In both cases, it is possible to use a proxy to do that, although the supported proxy configuration differs.

How to configure a proxy for AWS plugin

When installing the AWS plugin, you can configure it to run discoveries through a proxy by filling in the appropriate fields of the installation Fixlet or of the Install Cloud Plugin page on the BigFix WebUI. The WebUI also allows you to specify the proxy at a later time by editing the configuration of an installed plugin.

In case of an HTTPS proxy, it is possible to configure the AWS plugin to validate the SSL certificate of the proxy using a custom CA certificate file. A BES Support task allows you to carry out this configuration.

How to configure a proxy for Microsoft Azure plugin

In order to have the Microsoft Azure plugin go through a proxy, it is necessary to configure the proxy as follows:

Windows

The proxy must be set at system level using the *http_proxy* and *https_proxy* environment variables.

Linux

The following key/value pairs must be specified in the */etc/opt/BESPluginPortal/custom.config* file (create this file manually if it does not already exist):

- https_proxy=*http://proxyHost:proxyPort/*
- http_proxy=*http://proxyHost:proxyPort/*

On both Windows and Linux, restart the BigFix Plugin Portal service to make the proxy configuration effective.

Note: Because the Azure SDK embedded in the Azure cloud plugin requires the proxy to be set at system level, communications from the local Plugin Portal to its parent Relay will also be affected. To restore direct communications, The Plugin Portal must have the proxy defined through the same BigFix configuration settings, that is `_Enterprise Server_ClientRegister_Proxy<attributename>`, generally used to set up a proxy connection on a Relay and with the parent Relay included in the related exception list setting. For more details, see Server/Relay *(on page 663)*.

Related information

The Plugin Portal *(on page 213)*

Understanding cloud setup *(on page 255)*

Cloud Plugins Inspectors

Resources reported by the cloud plugins contain a lot of information related to the instances they represent. Such data is made available through numerous inspectors.

These inspectors are divided between the common section, that all plugins share, and a more specific section for each plugin which is exclusive to the single plugin. In this topic, we describe the **common inspectors**.

Computer Name

```
computer name: string
```

The display name of the device.

CPU

```
main processor: processor
```

Returns the processor object corresponding to the main processor.

```
speed of <processor>: hertz
```

Returns the speed of the processor in Hertz.

```
family name of <processor>: string
```

Returns the family name of the CPU.

Data Source

```
data source: string
```

The data source is the name of the system where data is coming from.

Device ID

```
device id: string
```

The device ID is a unique identifier for each device.

Device Type

```
device type: string
```

The type of device. For cloud plugins resources, it is set to “Cloud” by default.

DNS Name

```
dns name: string
```

The DNS name.

Network

```
network: network
```

The network inspectors report data about the network configuration.

```
ip interfaces of <network>: plural ip interface
```

Returns all the ip interfaces of the network.

```
address of <ip interface>: ipv4or6 address
```

Returns the ip address of the ip interface.

```
subnet address of <ip interface>: ipv4or6 address
```

Returns the subnet address to which the specified interface belongs.

```
loopback of <ip interface>: boolean
```

Indicates whether the particular network ip interface is a loopback interface.

```
adapters of <network>: plural network adapter
```

Returns the one or more network adapter objects of the network.

```
up of <network adapter>: boolean
```

Returns True if the specified network adapter is currently working.

```
loopback of <network adapter>: boolean
```

Returns True if the specified network adapter is a loopback interface.

```
ipv6 interfaces of <network adapter>: plural network adapter interface
```

Returns the IPv6 interfaces of the specified network adapter as a network adapter interface type.

```
address of <network adapter interface>: ipv4or6 address
```

Returns the IP address of the specified network adapter interface as an ipv4or6 address type.

OS

```
operating system: operating system
```

The OS inspectors return data about the operating system. By default the name and version are reported, but additional properties are available depending on the cloud plugin.

```
name of <operating system>: string
```

The name of the OS.

```
version of <operating system>: version
```

The version of the OS.

RAM

```
ram: ram
```

Returns a ram object for inspecting the properties of random access memory of the machine. Additional properties are available depending on the cloud plugin.

```
size of <ram>: integer
```

Returns the number of bytes of random access memory on the current machine.

AWS Asset Discovery Plugin Inspectors

AMI launch index

```
ami launch index: integer
```

The AMI launch index.

Association Overview

```
association overview: association overview
```

The Association Overview inspectors return the status information about the aggregated associations.

```
detailed status of <association overview>: string
```

The detailed status information about the aggregated associations.

```
status count of <association overview>: string
```

The number of associations for the instance.

AWS Image

```
aws image: aws image
```

The AWS Image inspectors represent an AMI and its properties.

```
id of <aws image>: string
```

The ID of the AMI.

```
name of <aws image>: string
```

The name of the AMI.

```
architecture of <aws image>: string
```

The architecture of the AMI.

```
description of <aws image>: string
```

The description that was provided during the AMI creation.

```
creation date of <aws image>: string
```

The time and date of the AMI creation.

```
hypervisor of <aws image>: string
```

The hypervisor of the AMI.

```
location of <aws image>: string
```

The location of the AMI.

```
owner alias of <aws image>: string
```

The AWS account alias or the AWS account ID of the AMI owner.

```
type of <aws image>: string
```

The type of the AMI.

```
kernel id of <aws image>: string
```

The kernel associated with the image, if any. Only applicable for machine images.

```
public of <aws image>: boolean
```

Returns true if the image has public launch permissions or false if it has only implicit and explicit launch permissions.

```
tags of <aws image>: plural tag
```

An array of tags assigned to the AMI.

```
product codes of <aws image>: plural product code
```

An array of product codes.

```
virtualization type of <aws image>: string
```

The virtualization type of the AMI.

```
platform details of <aws image>: string
```

The platform details associated with the billing code of the AMI.

Block Device Mapping

```
block device mapping: block device mapping
```

```
[block device mapping, block device mappings]: plural block device mapping
```

The Block Device Mapping inspectors describes a block device mapping. The return an array of data for each block device mapping.

```
device name of <block device mapping>: string
```

The device name of the block device mapping.

```
ebs of <block device mapping>: ebs
```

Parameters used to automatically set up EBS volumes when the instance is launched.

CPU Package

```
cpupackage: cpupackage
```

The CPU Package inspectors return data about the CPU options.

```
core of <cpupackage>: integer
```

The number of cores for the instance.

```
thread of <cpupackage>: integer
```

The number of threads per core.

Credentials Label

```
credentials label: string
```

The credentials label is a custom string defined by the user when installing the plugin. It uniquely identifies a user’s set of credentials.

Credentials Role

```
credentials role: string
```

The role ARN of the IAM Role through which the instance was discovered. The value is left empty if no role is specified in the credentials.

EBS

```
ebs: ebs
```

```
ebses: plural ebs
```

The EBS inspectors describe the parameters used to set up an EBS volume in a block device mapping.

```
attach time of <ebs>: string
```

The time stamp when the attachment initiated.

```
delete on termination of <ebs>: boolean
```

Indicates whether the volume is deleted on instance termination.

```
status of <ebs>: string
```

The attachment state.

```
volume id of <ebs>: string
```

The ID of the EBS volume.

ENA enabled

```
ena enabled: boolean
```

Specifies whether enhanced networking with ENA is enabled.

Hypervisor

```
hypervisor: string
```

The hypervisor of the instance.

Image ID

```
image id: string
```

The ID of the AMI image used to launch the instance.

Instance ID

```
instance id: string
```

The unique identifier for the instance.

Instance Type

```
instance type: string
```

The type of the instance.

Key name

```
key name: string
```

The name of the key pair the instance was launched with, if any.

Launch Time

```
launch time: string
```

The time the instance was launched.

Owner ID

```
owner id: integer
```

The ID of the instance owner.

Placement

```
placement: placement
```

The location where the instance launched, if applicable.

```
affinity of <placement>: string
```

The affinity setting for the instance on the Dedicated Host.

```
availability zone of <placement>: string
```

The availability zone of the instance.

```
group name of <placement>: string
```

The name of the placement group the instance is in.

```
host id of <placement>: string
```

The ID of the Dedicated Host on which the instance resides.

```
partition number of <placement>: string
```

The number of the partition the instance is in. Valid only if the placement group strategy is set to partition.

```
spread domain of <placement>: string
```

The spread domain of the instance.

```
tenancy of <placement>: string
```

The tenancy of the instance (if the instance is running in a VPC).

Private DNS name

```
private dns name: string
```

The private DNS hostname name assigned to the instance.

Private IP Address

```
private ip address: string
```

The private IPv4 address assigned to the instance.

Product Code

```
product codes: plural product codes
```

The product code inspectors return an object that describes a product code.

```
id of <product code>: string
```

The product code.

```
type of <product code>: string
```

The type of the product code.

Public IP Address

```
public ip address: string
```

The public IPv4 address, or the Carrier IP address assigned to the instance.

Public DNS name

```
public dns name: string
```

The public DNS hostname name assigned to the instance.

Region

```
region: string
```

The AWS Region where the instance resides.

Requested ID

```
requester id: integer
```

The ID of the requester that launched the instances on your behalf.

Root Device Type

```
root device type: string
```

The root device type used by the AMI.

Security Group

```
security groups: plural security group
```

The Security Group inspectors return data about the security groups assigned to an instance.

```
id of <security group>: string
```

The ID of the security group.

```
name of <security group>: string
```

The name of the security group.

State

```
state: string
```

The current state of the instance.

Subnet ID

```
subnet id: string
```

The ID of the subnet in which the instance is running.

System Manager

```
system manager: system manager
```

The System Manager inspectors returns information about the AWS System Manager, if available for the instance. The System Manager inspectors are required for the BigFix Agent deployment on AWS through native APIs. See BigFix Agent installation on cloud resources (on page 353) for further details.

```
activation id of <system manager>: string
```

The activation ID created by Systems Manager when the server or VM was registered.

```
agent version of <system manager>: string
```

The version of the SSM agent present on the instance, if any. Should exist for the BigFix Agent installation.

```
association overview of <system manager>: association overview
```

Returns the overview of the aggregated association.

```
association status of <system manager>: string
```

The status of the association.

```
computer name of <system manager>: string
```

The fully qualified domain name of the instance.

```
ip address of <system manager>: string
```

The IP address of the instance.

```
iam role of <system manager>: string
```

The IAM Role assigned to the on-premises Systems Manager managed instance.

```
instance id of <system manager>: string
```

The instance ID.

```
latest version of <system manager>: boolean
```

The version of the SSM Agent installed.

```
last association execution of <system manager>: string
```

The last date the association was run.

```
last ping of <system manager>: string
```

The time and date the SSM Agent pinged the System Manager service.

```
last successful association of <system manager>: string
```

The last date the association was successfully run.

```
name of <system manager>: string
```

The name assigned to an on-premises server or virtual machine when it is activated as a Systems Manager managed instance.

```
ping status of <system manager>: string
```

The connection status of the SSM Agent. Should be “Online” for the BigFix Agent installation.

```
platform name of <system manager>: string
```

The name of the operating system platform of the instance.

```
platform type of <system manager>: string
```

The type of the operating system platform of the instance.

```
platform version of <system manager>: string
```

The version of the operating system platform of the instance.

```
registration date of <system manager>: string
```

The date the server or VM was registered with AWS as a managed instance.

```
resource type of <system manager>: string
```

The resource type, either EC2 or managed instance.

Tag

```
tags: plural tag
```

The Tag inspectors return the array of tags associated to an instance.

```
key of <tag>: string
```

The key of the Tag.

```
value of <tag>: string
```

The value of the Tag.

Uptime

```
uptime: integer
```

The hours since the time the instance was launched, if known.

Virtualization Type

```
virtualization type: string
```

The type of virtualization used by the instance.

VPC ID

```
vpc id: string
```

The ID of the VPC in which the instance is running.

Azure Asset Discovery Plugin Inspectors

Azure Image

```
azure image: azure image
```

The Azure Image inspectors return data concerning the Azure image of the instance.

```
id of <azure image>: string
```

Specifies the ID of the Azure image.

```
publisher of <azure image>: string
```

Specifies the image publisher.

```
offer of <azure image>: string
```

Specifies the offer of the platform image or marketplace image used to create the virtual machine.

```
sku of <azure image>: string
```

The image SKU.

```
version of <azure image>: string
```

Specifies the version of the platform image or marketplace image used to create the virtual machine. The allowed formats are Major.Minor.Build or 'latest'.

Credentials Label

```
credentials label: string
```

The credentials label is a custom string defined by the user when installing the plugin. It uniquely identifies a user's set of credentials.

Instance ID

```
[instance id, instances id]: string
```

```
instance id: string
```

The instance ID is the unique idenfitier for the instance and it corresponds to the VMID, a 128-bits identifier.

Instance Type

```
instance type: string
```

Specifies the size of the virtual machine.

License Type

```
license type: string
```

Specifies that the image or disk that is being used was licensed on-premises. This element is only used for images that contain the Windows Server operating system.

Linux Configuration

```
linux configuration of <operating system>: linux configuration
```

Specifies the Linux operating system settings on the virtual machine.

```
disable password authentication of <linux configuration>: boolean
```

Specifies whether the password authentication should be disabled.

```
provision vm agent of <linux configuration>: boolean
```

Indicates whether virtual machine agent should be provisioned on the virtual machine.

Network

The Network inspectors are defined in the common section. Here are the additional IP interface properties for the Microsoft Azure Plugin.

```
public address of <ip interface>: ipv4or6 address
```

The public IP address bound to the IP configuration.

Operating System

The Operating System inspectors are defined in the common section. Here are the additional properties for the Microsoft Azure Plugin.

```
admin username of <operating system>: string
```

Specifies the name of the administrator account.

```
windows configuration of <operating system>: windows configuration
```

Specifies the Windows operating system settings on the virtual machine.

```
linux configuration of <operating system>: linux configuration
```

Specifies the Linux operating system settings on the virtual machine.

OS Disk

```
os disk: os disk
```

The OS Disk inspectors report information about the operating system disk used by the virtual machine.

```
caching of <os disk>: string
```

Specifies the caching requirements.

```
create option of <os disk>: string
```

Specifies how the instance should be created.

```
disk size gb of <os disk>: string
```

The size of the disk reported in gigabytes.

```
storage account type of <os disk>: string
```

Specifies the storage account type for the managed disk.

```
name of <os disk>: string
```

The disk name.

```
os type of <os disk>: string
```

The OS type included in the disk.

Provisioning State

```
provisioning state: provisioning state
```

The Provisioning State inspectors return data about the state of the provisioning.

```
status of <provisioning state>: string
```

The short localizable label for the status.

```
time of <provisioning state>: string
```

The time of the status.

Region

```
region: string
```

The resource location.

Resource Group

```
resource group: string
```

The resource group of the virtual machine.

Status

```
status: string
```

The power state of the virtual machine.

Tag

```
tags: plural tag
```

The Tag inspectors return the array of tags associated to an instance.

```
key of <tag>: string
```

The key of the Tag.

```
value of <tag>: string
```

The value of the Tag.

Ultra SSD enabled

```
ultra ssd enabled: boolean
```

The flag that enables or disables a capability to have one or more managed data disks with UltraSSD_LRS storage account type on the VM or VMSS.

Windows Configuration

```
windows configuration of <operating system>: windows configuration
```

Specifies the Windows operating system settings on the virtual machine.

```
provision vm agent of <windows configuration>: boolean
```

Indicates whether virtual machine agent should be provisioned on the virtual machine.

```
enable automatic updates of <windows configuration>: boolean
```

Specifies whether the automatic updated are enabled on the Windows virtual machine.

```
timezone of <windows configuration>: string
```

Specifies the time zone of the virtual machine.

VM Agent

```
vm agent: vm agent
```

The VM Agent inspectors return data about the VM Agent present on the virtual machine, if available. The VM Agent inspectors are required for the BigFix Agent deployment on Azure through native APIs. See BigFix Agent installation on cloud resources (on page 353) for further details.

```
version of <vm agent>: string
```

The version of the VM Agent. Should exist for the BigFix Agent installation.

```
status of <vm agent>: string
```

The status of the VM agent. Should be “Ready” for the BigFix Agent installation.

GCP Asset Discovery Plugin Inspectors

Can IP Forward

```
can ip forward: boolean
```

Flags whether the instance is allowed to send and receive packets with non-matching destination or source IPs.

Computer name

```
computer name: string
```

The name of the resource, provided by the client when initially creating the resource.

CPU Platform

```
cpu platform: string
```

The CPU platform used by the current instance.

Creation Timestamp

```
creation timestamp: string
```

The creation timestamp.

Credentials Label

```
credentials label: string
```

The credentials label is a custom string defined by the user when installing the plugin. It uniquely identifies a user's set of credentials.

Data Source

```
data source: string
```

The data source is the name of the system where data is coming from. For the VMware Plugin, it is set to "Proxy – Google Cloud Platform" by default.

Deletion Protection

```
deletion protection: boolean
```

Whether the resource should be protected against deletion.

Description

```
description: string
```

Optional description of this resource.

Disk

```
disks: plural gcp disk
```

The disk inspectors contains information about the instance storage.

```
type of <gcp disk>: string
```

Specifies the type of the disk, either SCRATCH or PERSISTENT.

```
mode of <gcp disk>: string
```

The mode in which to attach this disk, either READ_WRITE or READ_ONLY.

```
source of <gcp disk>: string
```

Specifies a valid partial or full URL to an existing Persistent Disk resource.

```
device name of <gcp disk>: string
```

Specifies a unique device name. If not specified, the server chooses a default name in the form persistent-disk-x, where x is a number assigned by the Google Compute Engine. This field is only applicable for persistent disks.

```
index of <gcp disk>: integer
```

Zero-based index to this disk, where 0 is reserved for the boot disk.

```
boot of <gcp disk>: boolean
```

Indicates that this is a boot disk.

```
auto delete of <gcp disk>: boolean
```

Specifies whether the disk will be auto-deleted when the instance is deleted, but not when the disk is detatched.

```
interface of <gcp disk>: string
```

Specifies the disk interface to use for attaching this disk, which is either SCSI or NVME.

```
size gb of <gcp disk>: integer
```

The size of the persistent disk, specified in GB.

```
description of <gcp disk>: string
```

An optional description of the resource.

```
zone of <gcp disk>: string
```

The URL of the zone where the disk resides.

```
status of <gcp disk>: string
```

The status of disk creation. Possible values are CREATING, DELETING, READY, FAILED, RESTORING.

```
source image of <gcp disk>: gcp image
```

The source image used to create this disk.

```
creation timestamp of <gcp disk>: string
```

The timestamp fo the creation.

Hostname

```
hostname: string
```

Specifies the RFC1035 compliant hostname of the instance.

Image

```
archive size bytes of <gcp image>: integer
```

The size of the image tar.gz archive stored in the Google Cloud Storage (in bytes).

```
creation timestamp of <gcp image>: string
```

The creation timestamp.

```
source disk of <gcp image>: string
```

The URL of the source disk used to create this image.

```
size gb of <gcp image>: integer
```

The size of the image when restored onto a persistent disk in GB.

```
family of <gcp image>: string
```

The name of the image family to which this image belongs.

```
id of <gcp image>: integer
```

The unique identifier for the resource.

```
name of <gcp image>: string
```

The name of the resource; provided by the client when the resource is created.

```
status of <gcp image>: string
```

The status of the image. Possible values are DELETING, FAILED, PENDING, READY.

```
source type of <gcp image>: string
```

The type of the image used to create this disk.

```
description of <gcp image>: string
```

An optional description of the resource.

Instance ID

```
[instance id, instances id]: string
```

The unique identifier for the resource. This identifier is defined by the server.

Labels

```
labels: plural tag
```

Labels to apply to the instance.

```
key of <tag>: string
```

The key of the label.

```
value of <tag>: string
```

The value of the label.

Machine Type

```
machine type: string
```

Full or partial URL of the machine type resource to use for this instance, in the `zones/zone/machineTypes/machine-type` format. This is provided by the client when the instance is created.

Minimum CPU Platform

```
minimum cpu platform: string
```

Specifies a minimum CPU platform for the VM instance.

Network

```
network tags: plural string
```

The network inspectors are defined in the common section. An array of tags.

```
name of <ip interface>: string
```

The name of the network interface.

```
gateway address of <ip interface>: ipv4or6 address
```

The gateway address for default routes to reach destination addresses outside the subnetwork.

```
external address of <ip interface>: ipv4or6 address
```

An external IP address associated with this instance.

```
access configurations of <ip interface>: plural access configuration
```

The access configurations attached to an instance network interface.

```
type of <access configuration>: string
```

The type of configuration.

```
name of <access configuration>: string
```

The name of this access configuration.

```
nat ip of <access configuration>: ipv4or6 address
```

An external IP address associated with this instance.

```
network tier of <access configuration>: string
```

This signifies the networking tier used for configuring the corresponding access configuration.

```
domain name of <access configuration>: string
```

The DNS domain name for the public PTR record.

Project ID

```
project id: string
```

The ID of the project of the instance.

Self Link

```
self link: string
```

The Server-defined URL for the resource.

Start Restricted

```
start restricted: boolean
```

Whether a VM has been restricted for start because the Compute Engine has detected suspicious activity.

Status

```
status: string
```

The status of the instance.

Status Message

```
status message: string
```

A human-readable message explaining the status.

Zone

```
zone: string
```

The URL of the zone where the instance resides.

VMware Asset Discovery Plugin Inspectors

The Fixlet named **5629 - VMware Plugin manage Hosts discovery** can be used to retrieve information related to the Host VMware Inspectors and display them in the BigFix Console. These information apply to the inspector sections described in this page which contain **(Host)** in their titles.

Bios

```
vm bios: bios
```

The vm bios inspector returns contains the uuid of the bios property.

```
uuid of <bios>: string
```

The 128-bit SMBIOS UUID of a virtual machine represented as a hexadecimal string in the `12345678-abcd-1234-cdef-123456789abc` format.

Bios (Guest)

```
version of <vm bios>: string
```

The current BIOS version of the physical chassis.

```
release date of <vm bios>: time
```

The release date for the BIOS.

Bios (Host)

```
host bios : <host bios>
```

The Host BIOS inspector returns information about the system BIOS of the host.

```
model of <host bios>: string
```

The Host BIOS model identification.

```
platform of <host bios>: platform
```

Returns the platform object of the Host BIOS.

```
serial number of <platform>: string
```

The serial number of the Host BIOS, taken from the platform object.

Computer Name

```
computer name: string
```

The display name of the device. The name of the virtual machine for Guest devices and the host name for Host devices.

Credentials Label

```
credentials label: string
```

The credentials label is a custom string defined by the user when installing the plugin. It uniquely identifies a user's set of credentials.

Custom Attributes

```
custom attributes: plural custom attribute
```

This inspector returns the custom attributes. An array of base types for storing values.

```
attribute of <custom attribute>: string
```

The attribute name of the custom attribute.

```
value of <custom attribute>: string
```

The string value of the custom attribute.

Data Source

```
data source: string
```

The data source is the name of the system where data is coming from. For the VMware Plugin, it is set to “Proxy – VMware” by default.

Device

```
device: device
```

The overall alarm status on this node. In releases after vSphere API 5.0, vSphere Servers might not generate property collector update notifications for this property.

Device (Guest)

```
status of <device>: string
```

The current power state of the virtual machine.

```
state of <device>: string
```

The current state of the virtual machine.

```
vm uuid of <device>: string
```

The VC-specific identifier of the virtual machine.

```
bios uuid of <device>: string
```

The virtual machine BIOS identification.

```
description of <device>: string
```

The description for the virtual machine.

```
vm path name of <device>: string
```

The path name to the configuration file for the virtual machine

```
clean power off of <device>: string
```

For a powered off virtual machine, it indicates whether the virtual machine's last shutdown was an orderly power off or not. Unset if the virtual machine is running or suspended.

```
virtual disk count of <device>: integer
```

The number of virtual disks attached to the virtual machine.

```
ethernet card count of <device>: integer
```

The number of virtual network adapters.

```
modification time of <device>: time
```

The last executed task.

```
template of <device>: boolean
```

This data object type describes a template virtual machine configuration file.

Device (Host)

```
bios release date of <device>: time
```

The release date for the bios.

```
bios version of <device>: string
```

The current bios version of the physical chassis.

```
bios uuid of <device>: string
```

The hardware BIOS identification.

```
file system type of <device>: string
```

Represents the list of supported file system volume types.

```
maintenance mode of <device>: boolean
```

The flag to indicate whether or not the host is in maintenance mode. This flag is set when the host has entered the maintenance mode. It is not set during the entering phase of maintenance mode.

```
model of <device>: string
```

The system model identification.

```
network adapters of <device>: string
```

The number of network adapters.

```
pending restart of <device>: boolean
```

Indicates whether or not the host requires a reboot due to a configuration change.

```
port number of <device>: string
```

The port number.

```
status of <device>: string
```

The overall alarm status of the host.

```
vendor of <device>: string
```

The vendor type.

```
vmotion of <device>: boolean
```

The flag to indicate whether or not VMotion is enabled on this host.

```
vswitches of <device>: plural vswitches
```

The virtual switches.

```
portgroups of <vswitches>: plural string
```

The port groups of the virtual switches.

```
name of <vswitches>: string
```

The name of the virtual switches.

```
hostd log enabled of <device>: boolean
```

The flag to indicate if hostd log file exists.

```
messages log enabled of <device>: boolean
```

The flag to indicate if messages log file exists.

```
vmkernel log enabled of <device>: boolean
```

The flag to indicate if vmkernel log file exists.

Device ID

```
device id: string
```

The device ID is a unique identifier for each device. For VMware Guest devices, it corresponds to the instance UUID, which is the VC-specific identifier of the virtual machine. For VMware Hosts, it corresponds to the Host name with the “Host - “ prefix prepended.

Disk

```
disk: disk
```

The disk inspectors return data concerning the storage used by a machine.

```
provisioned storage of <disk> : string
```

The total storage space, in bytes, both committed and uncommitted to this virtual machine on all datastores.

```
used storage of <disk> : string
```

The total storage space, in bytes, committed to this virtual machine across all datastores.

```
unshared storage of <disk> : string
```

The total storage space, in bytes, occupied by the virtual machine across all datastores, that is not shared with any other virtual machine.

Drive (Guest)

```
drive: drive
```

```
drives: plural drive
```

The drive inspectors return data about the file system.

```
name of <drive>: string
```

The name of the virtual disk in the guest operating system. For example: C:\

```
free space of <drive>: integer
```

The free space on the disk, in bytes. This is retrieved by the VMware Tools.

```
total space of <drive>: integer
```

The total capacity of the disk, in bytes. This is part of the virtual machine configuration.

Guest VM

```
 guest vms: plural guest vm
```

These inspectors retrieve properties for the guest virtual machine.

```
name of <guest vm>: string
```

The guest name configured on the virtual machine.

```
network of <guest vm>: network
```

The network configured on the virtual machine.

```
operating system of <guest vm>: operating system
```

The operating system on the virtual machine.

```
host of <guest vm>: host
```

The host name.

```
 tool of <guest vm>: tool
```

The tool on the virtual machine.

```
processor of <guest vm>: processor
```

The processor on the virtual machine.

```
ram of <guest vm>: ram
```

The RAM on the virtual machine.

```
usage statistic of <guest vm>: usage statistic
```

The usage statistic of the virtual machine.

```
drives of <guest vm>: drive
```

The drives of the virtual machine.

```
device of <guest vm>: device
```

The devices configured on the virtual machine.

```
snapshots of <guest vm>: snapshot
```

The snaphots of the virtual machine.

Hardware

```
hardware: hardware
```

This inspector holds properties about the hardware type of the VMware representation of an instance. By default, the virtual and proxied properties return true on cloud plugins representations.

```
virtual of <hardware>: boolean
```

Returns true if the client is running on a virtual machine.

```
proxied of <hardware>: boolean
```

Returns true if the device reports to the BES Plugin Portal. Otherwise, it returns false.

Host (Guest)

```
host: host
```

The Host inspectors report the name of the host of the current guest.

```
name of <host>: string
```

The name of the host.

Services (Host)

```
[host service, host services]: plural host service
```

The Host Service inspectors return information about each configured service that runs on a host.

```
key name of <host service> : string
```

The brief identifier for the service.

```
label of <host service> : string
```

Displays the label for the service.

```
policy of <host service> : string
```

The service activation policy.

```
required of <host service> : string
```

The flag indicating whether the service is required and cannot be disabled.

```
running status of <host service> : string
```

The flag indicating whether the service is currently running.

Network

The network inspector is defined in the common section. The following properties are available for the VMware cloud plugin.

```
ip interfaces of <network>: plural ip interface
```

Returns all the IP interfaces of the network.

```
address of <ip interface>: string
```

The primary IP address assigned to the guest operating system, if known. Otherwise, the preferred one.

```
loopback of <ip interface>: boolean
```

The routing of electronic signals or digital data streams back to their source without intentional processing or modification.

```
mac address of <ip interface>: string
```

The MAC address of the adapter.

Network (Guest)

The VNIC inspectors provide Guest information about network adapters, if known.

```
vnics of <network>: plural vnic
```

Returns the VNICs of the network.

```
vnic counter of <network>: integer
```

Represents the virtual network identity cards counter.

```
name of <vnic>: string
```

The label for the device.

```
up of <vnic>: boolean
```

The flag indicating whether or not the virtual device is connected.

```
mac address of <vnic>: string
```

The MAC address of the adapter.

```
[address, addresses] of <vnic>: plural string
```

Zero, one or more manual (static) assigned IP addresses to be configured on a given interface.

```
dhcp enabled of <vnic>: boolean
```

Indicates whether or not dynamic host control protocol (DHCP) is used to configure DNS configuration.

```
connected of <vnic>: boolean
```

The flag indicating whether or not the virtual device is connected.

```
device config id of <vnic>: integer
```

The link to the corresponding virtual device.

```
label of <vnic>: string
```

The display label of the VNIC.

```
address counter of <vnic>: string
```

The number of addresses found.

```
vnic counter of <network>: string
```

The number of VNICs found.

OS

The operating system inspector is defined in the common section.

```
name of <operating system>: string
```

The complete product name, including the version information, if known. Otherwise, the short form of the product name.

```
version of <operating system>: string
```

The dot-separated version string.

OS (Guest)

```
[family, families] of <operating system>: string
```

The guest operating system family, if known.

```
id of <operating system>: string
```

The guest operating system identifier short name, if known.

OS (Host)

```
api version of <operating system>: version
```

The version of the API as a dot-separated string.

```
api type of <operating system>: string
```

Indicates whether or not the service instance represents a standalone host. If the service instance represents a standalone host, then the physical inventory for that service instance is fixed to that single host. VirtualCenter server provides additional features over single hosts.

```
boot time of <operating system>: time
```

Represents the time when the host was booted.

```
build number of <operating system>: integer
```

The OS host build number.

```
license product name of <operating system>: string
```

The name of the license product.

```
license product version of <operating system>: string
```

The version of the license product.

```
locale build of <operating system>: string
```

The build number for the current session's locale. Typically, this is a small number reflecting a localization change from the normal product build.

```
locale version of <operating system>: string
```

The version of the message catalog for the current session's locale.

```
major version of <operating system>: version
```

The first number of the OS Version property as a dot-separated string.

```
minor version of <operating system>: version
```

The second number of the OS Version property as a dot-separated string.

```
os type of <operating system>: string
```

The operating system type and architecture.

```
product line id of <operating system>: string
```

The product ID which is a unique identifier for a product line.

```
vendor of <operating system>: string
```

The name of the vendor of this product.

Patch Status (Host)

```
[patch status, patch statuses]: plural patch status
```

This inspector collects the information about the patch enforcement of the ESXi.

```
package name of <patch status>: string
```

The identifier that uniquely identifies the software package.

```
package install date of <patch status>: string
```

The time when the package was installed. On Autodeploy stateless installations, there is no value set.

```
package version of <patch status>: string
```

The version string that uniquely identifies the software package.

```
package vendor of <patch status>: string
```

The corporate entity that created the software package.

```
package acceptance of <patch status>: string
```

The acceptance level definitions. See here for details.

Processor

```
processor: processor
```

The processor inspector is defined in the common section. It returns processor objects defined on the machine.

```
speed of <processor>: hertz
```

Returns the speed of the processor in Hertz. Defaults to -1.

```
family name of <processor>: string
```

Returns the family name of the CPU. For the VMware cloud plugin it is left empty.

Processor (Guest)

```
count of <processor>: integer
```

The number of processors in the virtual machine.

```
resource settings of <processor>: resource setting
```

The resource settings specifies the reserved resource requirement as well as the limit (maximum allowed usage) for a given kind of resource. CPU allocation is specified in MHz.

Processor (Host)

```
logical processor count of <processor>: integer
```

The number of physical CPU threads on the host.

```
physical processor count of <processor>: integer
```

The number of physical CPU packages on the host.

```
power management currenty policy of <processor>: string
```

The information about the current CPU power management policy.

```
power management hardware support of <processor>: string
```

The information about the supported CPU power management.

```
total processor core count of <processor>: integer
```

The number of physical CPU cores on the host.

```
cpu packages of <processor>: plural cpu package
```

The CPU packages of the processor.

```
cpu pkg bus hz of <cpu package>: integer
```

The bus speed in HZ.

```
cpu pkg description of <cpu package>: string
```

The string summary description of the CPU.

```
cpu pkg hz of <cpu package>: integer
```

The CPU speed in HZ.

```
cpu pkg vendor of <cpu package>: string
```

The CPU vendor name. Possible names currently are "Intel", "AMD", "arm", "hygon", or "unknown".

RAM

The RAM inspector is defined in the common section.

```
size of <ram>: integer
```

Returns the number of bytes of random access memory on the current machine.

RAM (Guest)

The amount of memory resource (in bytes) that will be used by the virtual machine above its guest memory requirements. This value is set if and only if the virtual machine is registered on a host that supports memory resource allocation features. For powered off VMs, this is the minimum overhead required to power on the VM on the registered host. For powered on VMs, this is the current overhead reservation, a value which is almost always larger than the minimum overhead, and which grows with time.

```
overhead of <ram>: integer
```

The bytes of memory used by the virtual machine above its guest memory requirements.

```
resource settings of <ram>: resource setting
```

Resource settings specifies the reserved resource requirement as well as the limit (maximum allowed usage) for a given kind of resource. Memory allocation is specified in MB.

Resource Setting

The properties below are common to ram and processor.

```
reservation of <resource setting>: integer
```

The amount of resource that is guaranteed available to the virtual machine or resource pool. Reserved resources are not wasted if they are not used. If the utilization is less than the reservation, the resources can be utilized by other running virtual machines. Units are MB for memory, MHz for CPU.

```
limit of <resource setting>: string
```

The utilization of a virtual machine/resource pool will not exceed this limit, even if there are available resources. This is typically used to ensure a consistent performance of virtual machines / resource pools independent of available resources. If set to -1, then there is no fixed limit on resource usage (only bounded by available resources and shares). Units are MB for memory, MHz for CPU.

```
overhead limit of <resource setting>: string
```

The maximum allowed overhead memory. For a powered on virtual machine, the overhead memory reservation cannot be larger than its overheadLimit. This property is only applicable to powered on virtual machines and is not persisted across reboots. This property is not applicable for resource pools. If set to -1, then there is no limit on reservation. Units are MB.

```
shares of <resource setting>: string
```

The allocation level. The level is a simplified view of shares. Levels map to a pre-determined set of numeric values for shares. If the shares value does not map to a predefined size, then the level is set as custom.

Snapshot (Guest)

```
snapshots: plural snapshot
```

The snapshot inspectors report all the read-only data produced by a snapshot.

```
creation time of <snapshot>: time
```

The date and time the snapshot was taken.

```
description of <snapshot>: string
```

The description of the snapshot.

```
name of <snapshot>: string
```

The name of the snapshot.

```
id of <snapshot>: integer
```

The unique identifier that distinguishes this snapshot from other snapshots of the virtual machine.

```
quiesced of <snapshot>: boolean
```

The flag to indicate whether or not the snapshot was created with the "quiesce" option, ensuring a consistent state of the file system.

```
replay supported of <snapshot>: boolean
```

The flag to indicate whether this snapshot is associated with a recording session on the virtual machine that can be replayed.

```
type of <snapshot>: string
```

The name of the managed object type of this snapshot.

```
value of <snapshot>: string
```

The specific instance of the managed object of this snapshot.

```
vm state of <snapshot>: string
```

The power state of the virtual machine when this snapshot was taken.

```
children of <snapshot>: snapshots
```

The snapshot data for all snapshots for which this snapshot is the parent.

Tool (Guest)

```
tool : <tool>
```

The tool inspectors report data about the VMware Tools.

```
status of <tool>: string
```

The current version status of VMware Tools, if known.

```
running status of <tool>: string
```

The current running status of VMware Tools in the guest operating system, if known.

```
version of <tool>: string
```

The current version of VMware Tools, if known.

```
version status of <tool>: string
```

The current version status of VMware Tools in the guest operating system, if known.

Usage Statistic (Guest)

```
usage statistic: usage statistic
```

The Usage Statistic inspector returns a set of statistics for a Virtual Machine.

```
overall cpu of <usage statistic>: hertz
```

The basic CPU performance statistics, in MHz. Valid while the virtual machine is running.

```
memory of <usage statistic>: integer
```

The guest memory utilization statistics, in MB. This is also known as active guest memory. The number can be between 0 and the configured memory size of the virtual machine. Valid while the virtual machine is running.

```
maximum cpu of <usage statistic>: hertz
```

The current upper-bound on CPU usage.

```
[maximum memory, maximum memories] of <usage statistic>: integer
```

The current upper-bound on memory usage.

```
[host memory, host memories] of <usage statistic>: integer
```

The Host memory utilization in bytes, including the overhead memory of the MV. The value ranges between 0 and the configured resource limit and it is valid while the virtual machine is running. Also known as the consumed host memory.

USB Device (Host)

```
[usb device, usb devices] : plural usb device
```

The USB device inspector returns a set of properties.

```
name of <usb device>: string
```

The USB device name.

```
description of <usb device>: string
```

A user visible name of the USB device.

```
family of <usb device>: string
```

The device class families. For possible values, see VirtualMachineUsbInfoFamily.

```
physicalpath of <usb device>: string
```

An autoconnect pattern which describes the device physical path. This is the path to the specific port on the host where the USB device is attached.

```
product of <usb device>: integer
```

The product ID of the USB device.

```
speed of <usb device>: string
```

The possible device speeds detected by the server. For possible values, see VirtualMachineUsbInfoSpeed.

```
vendor of <usb device>: integer
```

The vendor ID of the USB device.

```
usage summary of <usb device>: usage summary
```

The summary information about the virtual machine that is currently using this device, if any.

```
vm name of <usage summary>: string
```

The name of the virtual machine

```
vm path of <usage summary>: string
```

The path name to the configuration file for the virtual machine.

Virtualization inspectors

```
datacenters: plural datacenter
```

The Virtualization inspectors return a set of datacenter properties.

Virtualization inspectors for datacenters

```
name of <datacenter>: string
```

The name of the datacenter.

```
datastores of <datacenter>: plural datastore
```

A collection of references to the datastore objects available in this datacenter.

```
name of <datastore>: string
```

The name of the datastore.

```
free space of <datastore>: string
```

The free space of this datastore, in bytes. The server periodically updates this value. It can be explicitly refreshed with the Refresh operation.

```
max file size of <datastore>: string
```

The maximum size of a file that can reside on this file system volume.

```
timestamp of <datastore>: string
```

The time when the free-space and capacity values in DatastoreInfo and DatastoreSummary were updated.

```
url of <datastore>: string
```

The unique locator for the datastore.

```
iso images of <datastore>: plural iso image
```

The ISO images of the datastore.

```
path of <iso image>: string
```

The path relative to the folder path in the search results.

Virtualization inspectors for hard drivers

```
disk of <guest vm>: disk
```

The disk.

```
hard driver counter of <guest vm>: string
```

The number of hard drivers.

```
hard drivers of <guest vm>: plural hard driver
```

The hard drivers.

```
hard drivers: plural hard driver
```

Array.

```
unitnumber of <hard driver>: string
```

The unit number of this device on its controller. This property is null if the controller property is null (for example, when the device is not attached to a specific controller object).

```
datastore name of <hard driver>: string
```

The name of the datastore.

```
controllerkey of <hard driver>: string
```

The object key for the controller object for this device. This property contains the key property value of the controller device object.

```
label of <hard driver>: string
```

The label of the disk.

```
capacity of <hard driver>: string
```

The total capacity of the disk, in bytes. This is part of the virtual machine configuration.

```
key of <hard driver>: string
```

A unique key that distinguishes this device from other devices in the same virtual machine. Keys are immutable but may be recycled; that is, a key does not change as long as the device is associated with a particular virtual machine. However, once a device is removed, its key may be used when another device is added. This property is not read-only, but the client cannot control its value. Persistent device keys are always assigned and managed by the server, which guarantees that all devices will have non-negative key values. When adding new devices, it may be necessary for a client to assign keys temporarily in order to associate controllers with devices in configuring a virtual machine. However, the server does not allow a client to reassign a device key, and the server may assign a different value from the one passed during configuration. Clients should ensure that existing device

keys are not reused as temporary key values for the new device to be added (for example, by using unique negative integers as temporary keys). When editing or deleting a device, clients must use the server-provided key to refer to an existing device.

Virtualization inspectors for disk for last time refreshed

```
last time refreshed: string
```

Last time refreshed simply refers to the report production time for the disk discovery. The layout should be: `<year> <month> <day> <hour>:<min>:<sec>`.

```
last time refreshed of <guest vm>: string
```

Last time refreshed simply refers to the report production time for the disk discovery. The layout should be: `<year> <month> <day> <hour>:<min>:<sec>`.

Virtualization inspector for VM name

```
name: string
```

Name of the Guest or Host, unique relative to its parent. Could contain escaped characters.

Cloud plugins Commands

The Cloud plugins have the capability of supporting a wide range of actions.

Commands for the Cloud plugins work like the classical Action Script commands. The main difference is that the targets should be representations of some Cloud Provider instances.

In a single script, more commands can be included, which will be run in sequence on the relevant representations.

The applicability or the relevance for an instance is based on the properties reported by the corresponding Cloud plugin. Similarly, a Cloud plugin's ability to execute a command depends on which commands the plugin supports.

Ensure that the Cloud plugin supports the appropriate inspectors. For further reference, see Cloud Plugins Inspectors (on page 286).

Moreover, check the supported commands in the child pages of this section.

The commands are available starting from BigFix Version 11 Patch 2, while the Cloud Power commands are available starting from BigFix Version 11 Patch 3.

For an introduction to action scripts, see: https://developer.bigfix.com/action-script/

For the action scripts guide, see: https://developer.bigfix.com/action-script/guide/

Amazon Web Services Plugin Commands

Cloud Power Commands

The following table describes the Cloud Power Commands. These commands are available starting from BigFix Platform version 11.0.3.

Name	Syntax	Description
start instance	start instance	This command starts the target Cloud instance.
stop instance	stop instance	This command stops the target Cloud instance.
restart instance	restart instance	This command restarts the target Cloud instance.
stop force instance	stop force instance	This command forces the target Cloud instance shutdown operation.

Note: The exit codes returned by the actions performed in the BigFix Console match with the exit codes provided by Amazon Web Services (AWS) for the corresponding actions.

Required privileges:

- *ec2.StartInstances*
- *ec2.StopInstances*
- *ec2.RebootInstances*

In addition to the above listed privileges, your role also requires KMS permissions in order to perform the actions because EBS volumes, that instances use, are encrypted and without these permissions you cannot perform the actions on the device.

Install agent Command

The following table describes the install agent Command. This command is available starting from BigFix Platform versions 11.0.0 and 10.0.2.

Name	Syntax	Description
install agent	install agent	This command installs the BigFix agent on the target machine.

Action Commands

The following table describes the Action Commands. These commands are available starting from BigFix Platform version 11.0.6.

Name	Syntax	Description
set tag	set tag "<Key>" set tag "<Key>" "<Value>"	Adds or updates a tag with the specified key and value on the target instance.
delete tag	delete tag "<Key>"	Removes the tag with the specified key from the target instance.

Required privileges:

- *ec2:CreateTags* (Required for "set tag")
- *ec2:DeleteTags* (Required for "delete tag")

Google Cloud Platform Plugin Commands

Cloud Power Commands

The following table describes the Cloud Power Commands. These commands are available starting from BigFix Platform version 11.0.3.

Name	Syntax	Description
start instance	start instance	This command starts the target Cloud instance.
stop instance	stop instance	This command stops the target Cloud instance.
restart instance	restart instance	This command restarts the target Cloud instance.

Required privileges:

- *compute.instance.start*
- *compute.instance.stop*
- *compute.instance.reset*

Action Commands

The following table describes the Action Commands. These commands are available starting from BigFix Platform version 11.0.6.

Name	Syntax	Description
set tag	set tag "<Key>" set tag "<Key>" "<Value>"	Adds or updates a label with the specified key and value on the target instance.
delete tag	delete tag "<Key>"	Removes the label with the specified key from the target instance.

Required privilege:

- *compute.instances.setLabels*

Microsoft Azure Plugin Commands

Cloud Power Commands

The following table describes the Cloud Power Commands. The start / stop/ restart instance commands are available starting from BigFix Platform version 11.0.3, while the deallocate instance command is available starting from BigFix Platform version 11.0.4.

Name	Syntax	Description
start instance	start instance	This command starts the target Cloud instance.
stop instance	stop instance	This command stops the target Cloud instance.
restart instance	restart instance	This command restarts the target Cloud instance.
deallocate instance	deallocate instance	This command deallocates the target Cloud instance. It optimizes resource utilization and avoids paying for unused infrastructure.

Required privileges:

- *Microsoft.Compute/virtualMachines/runCommand/action*
- *Microsoft.Compute/virtualMachines/start/action*
- *Microsoft.Compute/virtualMachines/restart/action*
- *Microsoft.Compute/virtualMachines/powerOff/action*
- *Microsoft.Compute/virtualMachines/deallocate/action*

For more details about the required permissions, see Virtual Machine Contributor.

Install agent Command

The following table describes the install agent Command. This command is available starting from BigFix Platform versions 11.0.0 and 10.0.2.

Name	Syntax	Description
install agent	install agent	This command installs the BigFix agent on the target machine.

Action Commands

The following table describes the Action Commands. These commands are available starting from BigFix Platform version 11.0.6.

Name	Syntax	Description
set tag	set tag "<Key>" set tag "<Key>" "<Value>"	Adds or updates a tag with the specified key and value on the target instance.
delete tag	delete tag "<Key>"	Removes the tag with the specified key from the target instance.

Required privilege:

- *Microsoft.Compute/virtualMachines/write*

VMware Plugin Commands

Cloud Power Commands

The following table describes the Cloud Power Commands. These commands are available starting from BigFix Platform version 11.0.3.

Name	Syntax	Description
start instance	start instance	This command starts the target Cloud instance.

stop instance	stop instance	This command stops the target Cloud instance. The command does not complete successfully the operation if the VMware Tools are not installed on the target Cloud instance.
restart instance	restart instance	This command restarts the target Cloud instance.
stop force instance	stop force instance	This command forces the target Cloud instance shutdown operation.

Required VMware privileges:

- *VirtualMachine.Interact.PowerOn*
- *VirtualMachine.Interact.PowerOff* for stop hard instance
- *VirtualMachine.Interact.Reset* for restart instance

Host Commands

The following table describes the Host Commands:

Name	Syntax	Description
shutdown host	shutdown host [force=<force>]	This command shuts down a host.
reboot host	reboot host [force=<force>]	This command reboots a host.
enter maintenance	enter maintenance mode [action=<action> timeout=<time-	This command puts the host in maintenance mode. While this task is running and when the host is in maintenance mode, no virtual machines can be powered on

nance mode	out> evacuate=<evacuate>]	and no provisioning operations can be performed on the host.
exit maintenance mode	exit maintenance mode [timeout=<timeout>]	This command takes the host out of maintenance mode, unless any concurrent running maintenance-only host configurations operations are being performed.

Required VMware privilege:

- *Host.Config.Maintenance*

Guest Commands

The following table describes the Guest Power Commands:

Name	Syntax	Description
power on	power on	This command powers on the target Guest Virtual Machine.
power off hard	power off hard	This command powers off the target Guest Virtual Machine.
power off soft	power off soft	This command shuts down the target Guest Virtual Machine.
reset	reset	This command resets (first powers off, then powers on) the target Guest Virtual Machine.
restart	restart	This command reboots the target Guest Virtual Machine.
suspend soft	suspend soft	This command asks the target Guest Virtual Machine operating system to prepare for a suspend operation.
suspend hard	suspend hard	This command suspends the execution in the target Guest Virtual Machine.

Required VMware privileges:

- *VirtualMachine.Interact.PowerOn*
- *VirtualMachine.Interact.PowerOff* for both power off soft and hard
- *VirtualMachine.Interact.Suspend* for both suspend soft and hard
- *VirtualMachine.Interact.Reset* for both reset and restart

The following table describes the Guest Snapshot Commands:

Name	Syntax	Description
create snapshot	create snapshot <snapshot-name>	This command creates a snapshot with the given name.
revert snapshot	revert snapshot	This command reverts the virtual machine to the current snapshot. If no snapshot exists, the virtual machine state remains unchanged.
go to snapshot	go to snapshot <snapshot-name>	This command changes the execution state of the virtual machine to the state of the snapshot with the given name.
rename snapshot	rename snapshot <snapshot-name>|+| <new-snapshot-name>	This command renames the shapshot with the given name.
remove snapshot	remove snapshot <snapshot-name> [<children>]	This command removes the snapshot with the given name. If the optional boolean is set to true, it will delete all the snapshot children.
remove all snapshots	remove all snapshots	This command removes all the snapshot of a Virtual Machine.

Required VMware privileges:

- *VirtualMachine.State.CreateSnapshot*
- *VirtualMachine.State.RemoveSnapshot* for both remove snapshot and remove all snapshots
- *VirtualMachine.State.RenameSnapshot*
- *VirtualMachine.State.RevertToSnapshot* for both revert snapshot and go to snapshot

The following table describes the Guest Tools Commands:

Name	Syntax	Description
mount tools	mount tools	This command will mount the VMware Tools installer on a Virtual Machine.
upgrade tools	upgrade tools	This command upgrades the VMware Tools for a Virtual Machine.

Required VMware privilege:

- *VirtualMachine.Interact.ToolsInstall* for both mount tools and upgrade tools

The following table describes the Guest VLAN Command:

Name	Syntax	Description
change vlan	change vlan <vlan-name>|+|<adapter-number>	This command will switch the VLAN of a Virtual Machine.

Required VMware privilege:

- *VirtualMachine.Config.EditDevice*

Action Commands

The following table describes the Action Commands. These commands are available starting from BigFix Platform version 11.0.6.

Name	Syntax	Description

set custom attribute	set custom attribute "<Key>" "<Value>"	Sets the custom attribute value for a specific VM only if the attribute name exist globally in vCenter. The plugin will not create the custom attribute automatically.
set custom attribute	set custom attribute "<Key>" or set custom attribute "<Key>" ""	Unsets the attribute value from the specific VM. (It does not delete the Global Field definition).

Required VMware privilege:

- *Global.SetCustomField*

Activating cloud analyses

When a cloud plugin is installed, the following BES Support analyses must be activated to take advantage of the new cloud-related capabilities of BigFix.

Activate:

- BES Component Versions
- Cloud Correlation Data
- *Plugin_name* Plugin Settings
- *Plugin_name* Resources

- Some of the required analyses can be activated also from the Cloud Plugin Dashboard *(on page 350)*.
- Some of the required analyses are activated automatically when installing a cloud plugin from the WebUI.

The cloud analyses data

The analysis is sent to all the cloud virtual instances, discoverable by the specific cloud plugins, which evaluate it for relevance and report the status. You can monitor specific properties of your cloud virtual instances from the BigFix Console.

Data provided by the analysis: *Cloud provider* Plugin Settings

The information reported by these analyses are the following plugin settings:

- Settings: A multi-property which displays a list of all settings.
- Version: The plugin version.
- Refresh Interval: The value, specified in minutes, set for the Task named **Plugin Update Refresh Interval**.

Data provided by the analysis: Amazon Web Services Resources

This analysis reports information that is specific to each provider. In particular, the Amazon Web Services Resources analysis reports data such as:

- State AWS: The state of the virtual machine.
- Launch Time: The time when the instance was launched.
- Type AWS: The specific instance type.
- Image ID: The image ID number.
- Owner ID: The owner ID number.
- Tags AWS: The predefined tags used.
- Region AWS: The region in which the virtual machine is located.
- Tenancy: The tenancy model used. Dedicated or shared. The default tenancy model is the shared tenancy.
- Platform: The value is Windows for Windows instances; otherwise blank (this is a limitation for the API AWS).
- Private / Public IP AWS: The private / public IP address assigned to the virtual machine.
- Private / Public DNS Name: The private / public DNS assigned to the virtual machine.
- Security Groups: The security group, that acts as a virtual firewall for your instance, used to control inbound and outbound traffic.
- Availability Zone: The distinct location within the region that is engineered to be isolated from failures in other availability zones.
- Key Name: The key name associated to the virtual machine.
- VPC ID: The ID number of the specific Virtual Private Cloud (VPC) used.
- Image Name: A unique name for the image used.

- Instance ID AWS: The instance ID number.
- Account Label AWS: The credentials label.
- Correlation ID AWS: The correlation ID number.
- IAM Role AWS: The IAM Role used for discovering the instance.

The cloud plugins might be able to report additional information as well.

Data provided by the analysis: Microsoft Azure Resources

The information reported back from this analysis is specific for each provider. In particular, the Microsoft Azure Resources analysis reports back data such as:

- State Azure: The state of the virtual machine.
- Provisioning State: The provisioning state of the virtual machine.
- Provisioning Time: The provisioning time of the virtual machine.
- Type Azure: The type of virtual machine.
- Image Publisher: The publisher / organization that created the image.
- Image Offer: The name of a group of related images created by a publisher.
- Tags Azure: The predefined tags used.
- Region Azure: The region in which the virtual machine is located.
- Private / Public IP Azure: The private / public IP address assigned to the virtual machine.
- Resource Group: The group of resources the virtual machine belongs to.
- Instance ID Azure: The instance ID number.
- Account Label Azure: The credentials label.
- Correlation ID Azure: The correlation ID number.

The cloud plugins might be able to report back also additional information.

Data provided by the analysis: VMware Resources

The information reported back from this analysis is specific for each provider. In particular, the VMware Resources analysis reports back data such as:

- Custom Attributes: The Custom Attributes defined for the virtual machine.
- Operating System: The operating system used.

- Power State VMware: The power state of the virtual machine.
- Status VMware: The state of the virtual machine.
- BIOS UUID: The BIOS UUID (Universal Unique Identifier) number.
- Host: The host virtual machine.
- VM UUID: The virtual machine UUID (Universal Unique Identifier) number.
- Account Label VMware: The credentials label.

The cloud plugins might be able to report back also additional information.

Data provided by the analysis: Google Cloud Platform Resources

The information reported back from this analysis is specific for each provider. In particular, the Google Cloud Platform Resources analysis reports back data such as:

- Instance ID GCP: The instance ID number.
- Account Label GCP: The credentials label.
- Tags GCP: The predefined tags used.
- Status Message GCP: The status message.
- Status GCP: The status.
- Type GCP: The machine type.
- Creation Time: The creation timestamp.
- CPU Platform: The cpu platform.
- Zone GCP: The zone.
- Network Interface Name: The names of IP interfaces of network.
- Network Subnet Address: The subnet address of IP interfaces of network.
- Private IP GCP: The addresses of IP interfaces of network.
- Public IP GCP: The external addresses of IP interfaces of network.
- IP Forwarding: If the IP address can forward.
- Correlation ID GCP: The correlation ID.
- Project ID GCP: The project ID of the project to which the GCP instance belongs.

Cloud inspectors

Cloud inspectors gather information from the BigFix Agent, such as the following:

- Whether it is a virtual machine or not.
- If yes:
    - The name of the Cloud Provider.
    - Region and availability zone of the instance.
    - Unique ID of the instance (Instance ID or VM ID).
    - Private IP.

For details of various cloud provider configuration properties, see their respective documentation.

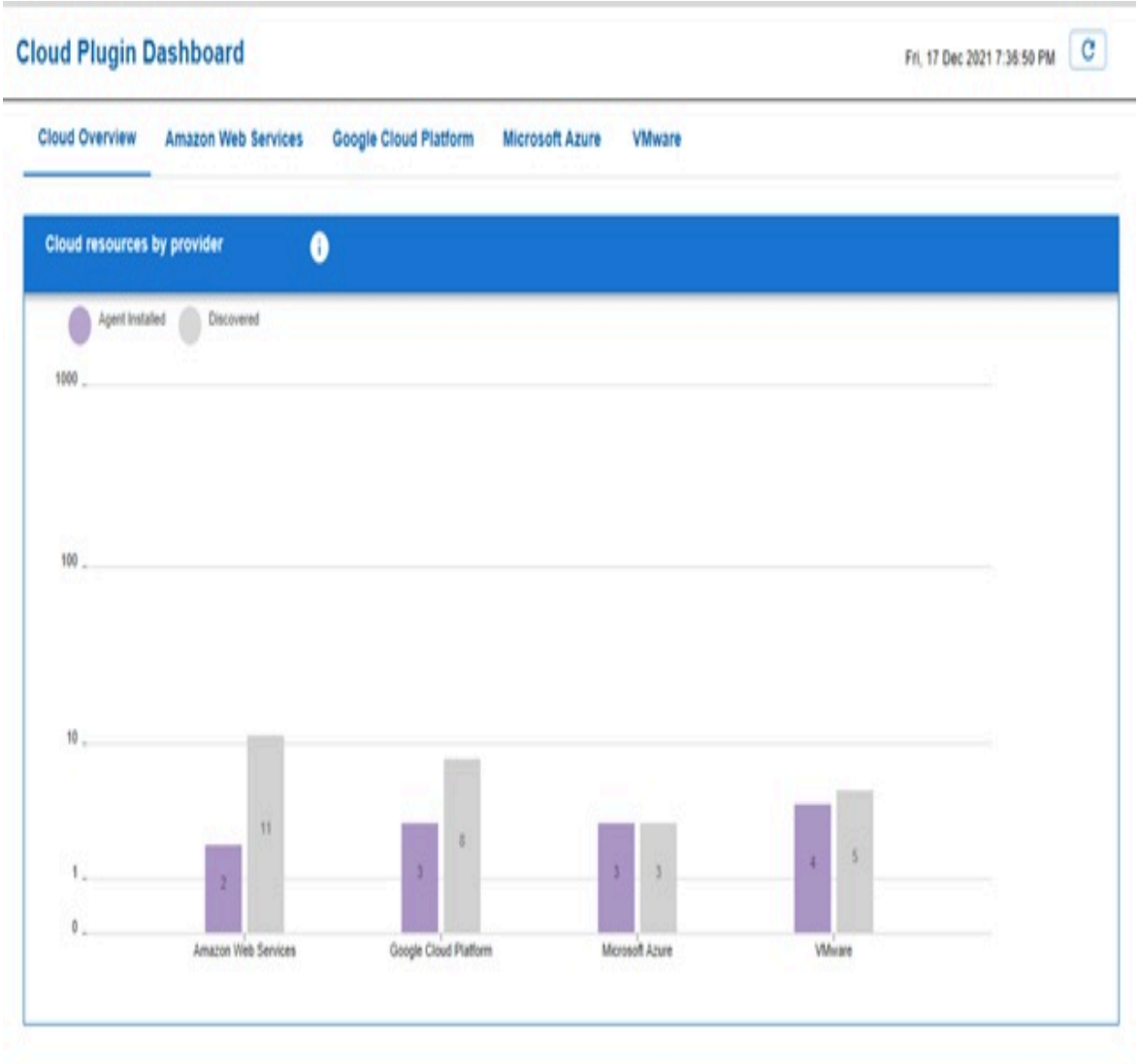
Working with the cloud plugin dashboard

The information displayed by the **Cloud Plugin Dashboard**.

Activate the **Cloud Providers** analysis of BES Support to start using the Cloud Plugin Dashboard.

To access the Cloud Plugin Dashboard, run the following steps on the BigFix Console:

1. Click the **All Content** domain on the leftmost panel of the console.
2. Expand the **Dashboards** entry.
3. Expand the **BES Support** entry.
4. Click **Cloud Plugin Dashboard**. The dashboard is displayed.



The **Cloud Overview** tab shows all discovered cloud resources, both managed by a BigFix Agent and unmanaged. The bar in violet indicates cloud resources on which the BigFix Agent is installed. The bar in gray indicates cloud resources on which the BigFix Agent is not installed yet.

When clicking on a violet bar or on a gray bar, details about all cloud resources represented by that bar are shown in a tabular format. You can then filter the items listed in the table by any column name. The "Name" column contains hyperlinks to the properties of the corresponding computers.

- The **Amazon Web Services** tab contains key information about the cloud resources discovered by the AWS plugin.
- The **Microsoft Azure** tab contains key information about the cloud resources discovered by the Azure plugin.
- The **VMware** tab contains key information about the cloud resources discovered by the VMware plugin.
- The **Google Cloud Platform** tab contains key information about the cloud resources discovered by the Google Cloud Platform plugin.

Installing the BigFix Agent on discovered resources

The resources retrieved by cloud plugins do not necessarily have the BigFix Agent installed. By using the **Cloud Plugin Dashboard** that lists all resources discovered on cloud, you can install the BigFix Agent on those that are still unmanaged.

1. From the **Cloud Overview** tab of the dashboard, click the gray bar for your cloud provider. A table with all unmanaged resources for that cloud provider is displayed.
2. In the table, click the row to select the resource on which you want to install the BigFix Agent. Multi-selection with Ctrl or Shift buttons is supported. If you want to select all the resources at once, click the **Select unmanaged resources** button.
3. Click the **Deploy agent** button.

The dashboard displays the Client Deployment Tool Wizard, with the list of targets to install already filled in. The installation then follows the standard Client Deployment Tool Wizard flow. For details on how to use the Client Deploy Tool Wizard, see Using the Client Deploy Tool.



Note: When launching the Client Deployment Tool Wizard from the **Cloud Plugin Dashboard**:

- In the **Set Target Credentials** page of the wizard, the **Add Targets** button is disabled.
- In the **Set Advanced Settings** page of the wizard, in the **OS families** section, the platforms which are not supported by the cloud infrastructures are greyed out.
- In the **Set Advanced Settings** page of the wizard, in the **Client Version** pull-down menu, only versions 10 or later are available.

Once the BigFix Agent is installed, the selected resources have two representations, the proxied one and the native one, which the BigFix Server correlates under one correlation computer.

BigFix Agent installation on cloud resources

Starting from BigFix Version 10 Patch 2, you can install the BigFix Agent on the discovered cloud resources (AWS and Azure), using the cloud provider services through the corresponding BigFix cloud plugin.

BigFix Platform offers new tasks (specific for each cloud provider) on BigFix Enterprise Suite (BES) Support so that you can deploy the agent on the discovered cloud resources.

The tasks are available from both the BigFix Console and from the WebUI. Use one of these tasks, depending on your cloud provider:

- `5433 Install BigFix Client version 11 through Amazon Web Services`
- `5434 Install BigFix Client version 11 through Microsoft Azure`
- `5951 Install BigFix Client version 11 through Google Cloud Platform` (available starting from BigFix Version 11.0.6)

What the tasks do

The tasks go through the following actions that are completed locally on the target cloud resources:

- Fetch the installation script from software.bigfix.com.
- Fetch the client installer based on the version of the Plugin Portal from software.bigfix.com.
- Fetch the deployment masthead from the non-authenticating relay that is supplied for input to the tasks.
- Install the client and register it with the non-authenticating relay.
- Wait for the client registration to finish.

Requirements

To use this new installation feature, the discovered cloud resources must satisfy a number of cloud provider-specific configuration requirements. A subset of these requirements is automatically checked by the relevances that are contained in the tasks. You must ascertain that your cloud resources are configured correctly.

The primary requirements for installing the Agent on discovered cloud resources are as follows:

- Cloud resources must be active and running.
- Provider-specific cloud agent must be installed on the cloud resource. For details, see Requirements for AWS *(on page 355)*, Requirements for Azure *(on page 356)* and Requirements for Google *(on page 357)*.
- The name of a non-authenticating relay (which you supply as input while you run the tasks). For details, see Relay name as input *(on page 357)*.
- Operating system running on the cloud resource must be supported by the BigFix Client. For details, see Supported operating systems *(on page 357)*.
- The Plugin Portal must be at BigFix Version 10 Patch 2 level at the minimum.

Important considerations

The feature does *not* support the following relays and settings:

- Authenticating relays at agent installation time on AWS and Azure cloud resources. After the client registration finishes, you can configure the authenticating relays and use them.
- Custom client settings that are set during installation.

Requirements for AWS

VM requirements

- The presence of an active and running AWS Systems Manager (SSM) agent on the VM
- An IAM instance profile with the `AmazonSSMManagedInstanceCore` IAM policy that is associated with profile through an IAM Role. The IAM instance profile must be attached to the VM. This requirement enables the AWS Systems Manager to securely run commands on the VM.

IAM Identities requirements

The basic requirements IAM Identities (Users and Roles) should always met, when configured in the AWS Cloud Plugin, are the following:

- MFA must be disabled
- Must have programmatic access type

When installing the BigFix Agent with an IAM User associated with the Access Key ID-Secret Access Key pair that is used as credentials of the AWS Cloud Plugin, the permissions required are as follows:

- Must have the following ec2 permissions at the minimum: the ec2:Describe* action must be allowed on the * resource.
- Must have the following ssm permissions at the minimum: the ssm:DescribeInstanceInformation ,ssm:SendCommand, ssm:GetCommandInvocation actions must be allowed on the * resource

When installing the BigFix Agent with an IAM Role that can be assumed by an IAM user configured in the AWS Cloud Plugin, the permissions required are as follows.

For the User:

- Must be able to perform sts:AssumeRole on the target Role

For the Role:

- Must have the following ec2 permissions at the minimum: the ec2:Describe* action must be allowed on the * resource.
- Must have the following ssm permissions at the minimum: the ssm:DescribeInstanceInformation ,ssm:SendCommand, ssm:GetCommandInvocation actions must be allowed on the * resource
- The IAM User assuming the role should be a trusted identity for the Role



Note: Once AWS Roles are inserted, the AWS plugin will use them during its discovery, instead of the credential from which they derive. You must ensure that these roles include all the AWS devices that you want to discover in your cloud environment: otherwise, some machines may not be discovered.

For a complete list of prerequisites, see Systems Manager prerequisites. Amazon Linux base AMIs that are dated 2017.09 or later include the systems manager by default. On other Amazon Machine Images (AMIs) or custom AMIs, you must install Systems Manager (SSM) Agent manually if it is not present already. For details about operating systems support, see Supported operating systems for Systems Manager.

Requirements for Azure

- The VM Agent must be present on the VM.
  Azure provides two agents that run commands, based on the platform:
  - Azure Linux Agent
  - Azure Virtual Machine Agent
- The operator (discovery credentials) must have at least the action permission.

If the agents are not present, you can install them as specified in the preceding references, provided that the OS requirements are met. To run commands on Azure VMs, an operator must have at least the action permission: `Microsoft.Compute/virtualMachines/runCommand/action`. The contributor role or higher built-in roles have this permission by default. However, you can also define a specific custom role.

Requirements for Google

In addition to the ones for discovering VM instances, to install a BigFix Agent, the **Google Cloud Service Account** set into the Google Cloud Plugin credential needs the permissions to:

- create and delete OS Policy Assignments
- check if the OS config service is enabled
- set labels to VM instances
- get Guest Attributes

This feature, starting from BigFix Version 11.0.6, also provides the `Install BigFix Client version 11 through Google Cloud Platform` Fixlet to install the BigFix agent on the Google Cloud.

Relay name as input

The tasks prompt for the name of a non-authenticating relay. Enter it in one of the following formats:

- The relay host name. For example, enter *`myhostname`*.
- The fully qualified domain name (FQDN) for the relay. For example, enter *`myhostname.mydomain.com`*.
- The relay IP address. For example, enter *`10.10.10.10`*.

Supported operating systems

This feature requires that one of the following provider-specific agents is installed and active:

- The SSM Agent for AWS
- The VM Agent for Azure
- The OS Config Agent for Google

Consequently, some operating systems that are supported by the provider agent might not be supported by BigFix. This subset of operating systems might vary in the future and is subject to change.

OS supported for AWS VMs

To see on which operating system types the AWS SSM agent is currently supported, see AWS SSM supported OSes.

For details about the BigFix agent operating systems currently supported, see Detailed system requirements.

OS supported for Azure VMs

To learn on which operating system types the Azure VM agent is currently supported, refer to the following documentation:

- Windows VM Agent supported OSes
- Linux VM Agent supported OSes
- Support extensions agent version

For details about the BigFix agent operating systems currently supported, see Detailed system requirements.

OS supported for Google Cloud VMs

To see on which operating system types the Google Cloud agent is currently supported, see Operating system details.

For details about the BigFix agent operating systems currently supported, see Detailed system requirements.

Troubleshooting

You have several ways to troubleshoot the agent installation failures:

- You can check a list of custom exit codes which is provided below.
- You can use a troubleshooting feature which is an sqlite database, named ActionResultsStore.db. The database is available for each cloud plugin.
- You can check the action logs. Logs can be checked manually and the default locations are described later.

Installation script exit codes

Table 4. Native agent installation exit codes

Exit code	Cause	How to resolve
209	Client log file not found. This might be because the installed client failed to install within the specified time.	Wait for a few minutes for the client to register. If that does not work, uninstall the client manually and install it again.
210	Client registration failed. This might be because the target cannot contact the relay or the masthead is corrupted.	Check whether the target instance resolves the relay FQDN, IP, or hostname and it resolves the hostname specified in the relay masthead.
211	BESClient service not found.	Check the client installation log on the target at /tmp/BigFix or %TEMP %/BigFix. Uninstall the client and install it again.
212	Client already installed. An installation of the client exists on the target.	
213	Missing input parameter. This might be due to a manual error.	Enter a value for the input parameter when prompted and try again.
215	Incorrect value returned by checksum. The downloaded installation script is corrupted.	Check if the target has network connetivity and can reach the BES Support site.

Table 4. Native agent installation exit codes (continued)

Exit code	Cause	How to resolve
216	Curl and wget utilities not found on Linux target.	Install wget or curl utilities on the Linux targets.
217	The retrieved OS data does not allow the script execution. This might be because the OS installed on the targets is not supported or the information about OS returned by the targets is insufficient.	Install the client manually or by using CDT.
218	Insufficient OS data retrieved. This might be because the OS installed on targets is not supported or the information about OS returned by the targets is insufficient.	Install the client manually or by using CDT.
219	Shasum and sha256sum unavailable on Linux target.	Install the shasum or sha256 utility on the Linux target.
220	Installer file not found.	Check whether the target can reach software.bigfix.com.
221	Masthead file not found.	Check whether the target can reach the relay. In case the relay is authenticating, install the client manually or by using CDT.
222	Error creating the OS Policy Assignment.	These errors apply to Google Cloud only. Check the Google Cloud Plug-in log for errors returned by Google Cloud Platform and try to solve
223	Error creating the temporary working directory.	

Table 4. Native agent installation exit codes (continued)

Exit code	Cause	How to resolve
224	Error executing the Validation script.	them. If not possible, install the client manually or by using CDT.
225	Error executing the Enforce script.	
226	Error getting results from the OS Policy Assignment report.	These errors apply to Google Cloud only. Check if the client is installed and registered to the BigFix server. If not, retry the installation. If the problem persists, check the Google Cloud Plugin log for errors returned by Google Cloud Platform and try to solve them. If not possible, install the client manually or by using CDT. In any case, it would be best to manually remove the OS Policy Assignment reported in the Google Cloud Plugin log using the Google Cloud Console.
227	Error getting the Exit Code from the installation script.	

Table 5. Other common installation exit codes

Exit code	Cause	How to resolve
6	On Linux targets, the curl error "Could not resolve host" occurs.	Check if the target can reach software.bigfix.com, the relay, and BES support site.
22	On Linux targets, the curl error "Request has encontered an error greater than 400" occurs.	Check if target can reach software.bigfix.com, the relay, and BES support site.

ActionResultsStore database

The ActionResultsStore database is stored inside each plugin folder and it contains the action results of the actions that have either a failed or error status.

The database consists of a table, named ACTION_RESULTS, which has the following columns:

- ActionID
- DeviceID
- InstanceID
- Results
- Date

The primary key will be (ActionID, DeviceID).
The Results column in the database contains a JSON string. Each JSON under Results has the following keys:

- **status** represents the status of the action.
- **exitCode** when there is an exit code available.
- **output** the output of the request to the provider.

An example table follows:

```
{
    "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"
   }
}
```

The database can be accessed remotely with the BigFix sqlite inspectors. As an example of this, we can use the Fixlet Debugger, set the query channel feature and the ID of the BES Portal endpoint, and evaluate a query like this example:

```
Q: rows of statement <sql statement> of sqlite database of file
 "ActionResultsStore.db"
of folder <plugin folder>
```

Once a day, the ActionResultsStore database will be automatically cleaned up. Every cleaning removes all entries older than a specified number of days, which can be configured by a setting that is available for each plugin, named `<plugin_name>_ActionResultsStore_CleanupDays`. The value, for this setting, must be larger than 0 (zero) and is specified in days.

Installer, masthead and Log file locations

The installer, masthead and log files, which are part of the installation task outcome, are stored in a folder named "BigFix" which can be found in the following directories, within each target instance:

On Windows

%LOCALAPPDATA%\BigFix

On Linux

/tmp/BigFix

Troubleshooting

This section helps you troubleshoot some common issues or limitations.

BESPluginPortal not running after installing a cloud plugin

Description: After installing a cloud plugin, the BESPluginPortal process is not running.

Reason: This behavior is unexpected and might be due to unpredictable reasons that need to be investigated.

Solution:

- On Windows:
  - Try restarting the BESPluginPortal process from the Windows Services tool.
  - If the BESPluginPortal is still not running, verify the presence of errors related to the BESPluginPortal process in the Windows Event Viewer.
- On non-Windows:

- Try restarting the BESPluginPortal process by issuing the `systemctl start bespluginportal` command.
- If the BESPluginPortal is still not running, verify the presence of error messages related to the BESPluginPortal process in `/var/log/messages`.

AWS plugin cannot discover resources due to authentication failure (status code: 401)

Description: The AWS cloud plugin fails to perform a resource discovery and logs the following error message:

```
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
```

Reason: The error message may be displayed in the following use cases:

1. The specified `Access Key ID` / `Secret Access Key` pair is either wrong or inactive.
2. The specified `Access Key ID` / `Secret Access Key` pair is associated to a SessionToken as part of a temporary credential set created by assuming a IAM Role.
3. Date and time of the computer on which the AWS plugin is installed is not accurate.

Solution: Depending on the use case, the solution differs as follows:

1. Verify that the specified `Access Key ID` / `Secret Access Key` pair is correct and that its status is active.
2. Temporary credentials associated to a Session Token are not supported. Replace the credentials with an `Access Key ID` / `Secret Access Key` pair associated to a IAM User.

3. Adjust the clock (date, time and timezone) of the computer where the AWS plugin is installed (+/- 5minutes should be the maximum deviation tolerated by AWS). For more information about signing incoming requests, refer to the AWS documentation.

Correlation computers not included in automatic computer groups

Description: When creating an automatic computer group with inclusion criteria that refer to the IDs of *correlation computers*, neither the *correlation computers* nor any of the correlated representations are included in the group.

Reason: *Correlation computers* represent logical entities that are only known to the BigFix Server, therefore no BigFix Agent answers for the ID of a *correlation computer*.

Solution: None. This is the expected behavior.

Distinct VMware computers correlated with each other

Description: Two distinct proxied computers of VMware type are correlated under the same *correlation computer*.

Reason: By design, the BigFix Server correlates VMware computers that are reporting the same value for the "BIOS UUID" property of the "VMware Resources" analysis. Although this value is generally expected to be unique in VMware, there are documented cases that might lead to having duplicated values, like converting a VM using the VMware converter or cloning a VM (not a template). For further information about the BIOS UUID duplication, refer to the VMware documentation at https://www.vmware.com/.

Solution: To eliminate a duplicate BIOS UUID, refer to the official VMware documentation. At the next discovery of the VMware plugin, after removing the BIOS UUID duplication, the unexpected correlation should be removed automatically.

VMware plugin does not discover some resources

Description: Some devices are not discovered by the VMware plugin. In the example, two instances are discarded:

```
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
```

Reason: The VMware plugin uses the vm.uuid as the unique key during its discovery. A device discovered that has not this key will not be considered by the plugin.

Solution: None. This is the expected behavior.

Cloud computers often displayed as offline

Description: The BigFix Console often shows proxied computers, discovered by the cloud plugins, offline.

Reason: By default, the cloud plugins have a discovery frequency of 120 minutes. This time interval is bigger than the default value for the "Mark as offline after" preference of the BigFix Console, which is 45 minutes. As a consequence, 45 minutes after the latest discovery performed by the cloud plugins, the proxied computers start displaying as offline on the BigFix Console, until the next discovery occurs.

Solution: This is the expected behavior when using the default values for the discovery frequency of the cloud plugins and for the "Mark as offline after" preference of the BigFix Console.

Chapter 18. Persistent connections

The capability to establish persistent connections was added to the product.

Clients/Relays behind firewall or NAT

Firewalls or NAT might prevent the BigFix Query function from working properly because the UDP notification, with which a parent relay delivers the query to the child clients, cannot usually reach the clients. Unlike other product functions, the BigFix Query cannot take advantage of client polling to overcome this restriction in the downstream communications.

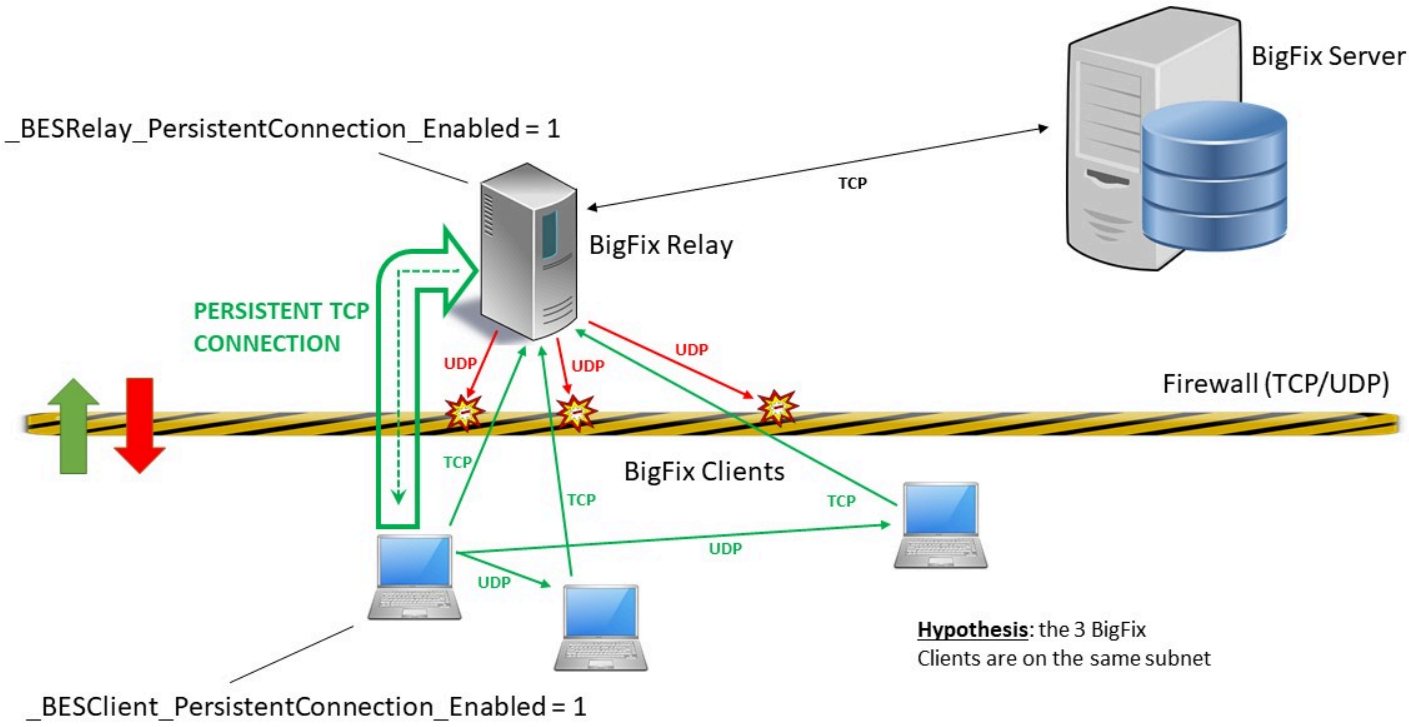
This restriction is overcome by establishing a persistent TCP connection between the parent relay and at least one of its child clients. The persistent connection, which is always initiated by the client, is used by the relay to send UDP notifications to all clients in the same subnet of the persistently connected client (PCC).

The same issues can occur on connections between a parent Relay and its child Relays. Also, in this case, the solution is to establish a TCP persistent connection between the parent relay and the child relay.

Client-Relay persistent connection

Overview

The following picture displays the persistent TCP connection established between client and relay, and the UDP notifications sent from the PCC to other clients of the same subnet:



Enabling persistent connections on the relay

1. Log in as a master operator to the BigFix Console.
2. Locate and right-click the relay computer. Select **Edit Computer Settings...**
3. Add the following setting to the computer:

   ```
   _BESRelay_PersistentConnection_Enabled = 1
   ```

4. Restart the relay process for the setting to become effective.

Note: When adding this setting to a relay computer having a Version prior to 9.5 Patch 11, the behavior of its child clients remains unchanged.

Note: This setting is not effective on the BigFix server computer.

Enabling persistent connections on the client

1. Log in as a master operator to the BigFix Console.
2. Locate and right-click the client computer. Select **Edit Computer Settings...**
3. Add the following setting to the computer:

```
_BESClient_PersistentConnection_Enabled = 1
```

Establishing a persistent connection

After being enabled, a persistent TCP connection between a client and its parent relay is normally established at the next registration of the client.

When the next registration occurs, the relay on which the client is registering checks whether the client is eligible to open a persistent connection, based on the overall number of persistent connections that the relay is already handling, and their partition by subnet. If the client is eligible, the relay notifies it accordingly. The client, then, waits for 60 seconds. If the client does not receive a test UDP notification from the relay within this time interval, it eventually opens the persistent connection.

If the client fails when establishing the persistent connection, it will retry opening the persistent connection after 3 minutes, up to a maximum of 4 attempts in total.

The persistent connection can generally be closed and then established again every time the client performs a new registration, provided that all prerequisites are still satisfied. The persistent connection might also terminate when either the client or the relay must handle restart and shutdown operations.

Communicating on the persistent connection

Directly:

If the relay must send a UDP notification to a persistently connected client (PCC), it uses the persistent connection to send it directly to the target client.

Served by another client of the same subnet:

If the relay must send a UDP notification to a client in a subnet served by a PCC, the relay sends the notification and the target client information (hostname/IP address stored during the registration phase) to the PCC. The PCC reads the notification and sends it through UDP to the target client. The target client processes the notification normally, and sends back a reply directly to the relay, as usually. If there is more than one PCC available, within the same subnet, that can serve the client, the relay sends the notification to one PCC only, not to all available PCCs.

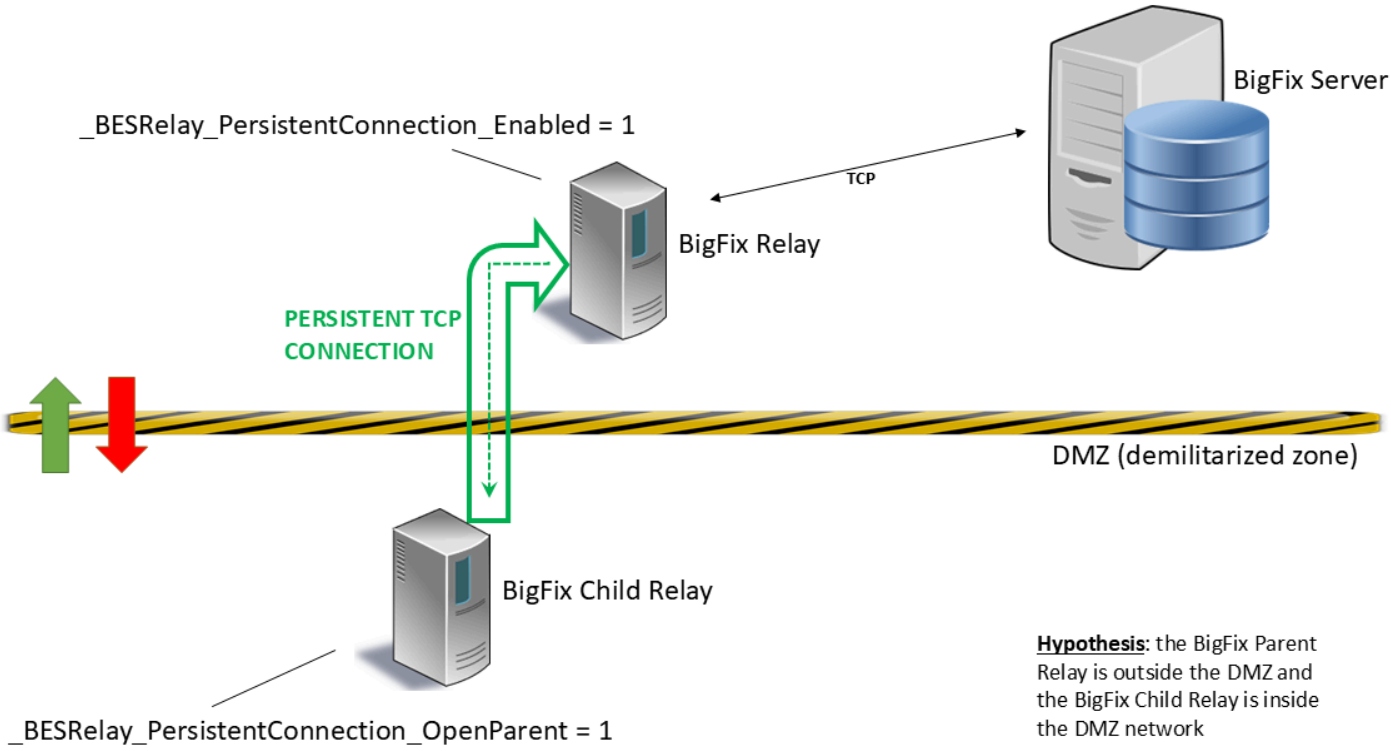
Managing persistent connections

You can manage persistent connections by configuring a few settings. For details, see Persistent TCP connections (on page 733).

Relay-Relay persistent connection

Overview

The following picture displays the persistent TCP connection established between child relay and parent relay.



Enabling persistent connections on the parent relay

1. Log in as a master operator to the BigFix Console.
2. Locate and right-click the parent relay computer. Select **Edit Computer Settings...**
3. Add the following setting to the computer:

```
_BESRelay_PersistentConnection_Enabled = 1
```

4. Restart the relay process for the setting to become effective.

Note: When adding this setting to a relay computer having a Version prior to Version 11, the behavior of its child clients remains unchanged.

Note: This setting is NOT effective on the BigFix server computer.

Enabling persistent connections on the child relay

1. Log in as a master operator to the BigFix Console.
2. Locate and right-click the client computer. Select **Edit Computer Settings...**
3. Add the following setting to the computer:

```
_BESRelay_PersistentConnection_OpenParent = 1
```

4. 4. Restart the relay process for the setting to become effective.

Establishing a persistent connection

The general flow for a Relay establishing a persistent connection with the parent Relay is the same as the Client-Relay one. The main difference is in the technical details for opening the connection and sending down the notification data.

The Relay when registering with the parent Relay, that is when the local Client registers, will be instrumented through a setting to ask for a permanent connection with the parent (_BESRelay_PersistentConnection_OpenParent). In this case, it will ask the parent Relay for a persistent connection. The parent Relay will perform the check on its settings: checking if the persistent connection setting is set and sends back a positive/negative response. In case of a positive response the Relays (both parent and child) will store the connected connection and set it as a keep alive connection.

Every time the parent Relay needs to send a notification to the child Relay, it will use the persistent connection.

A Relay will be persistent connected or not, there is no logic for handling relays in the same subnet, if such a need arise the suggestion is to use one relay of the isolated subnet as parent relay and set it to be persistent connected with its parent.

Managing persistent connections

You can manage persistent connections by configuring a few settings. For details, see Persistent TCP connections (on page 733).

Chapter 19. Relays in DMZ

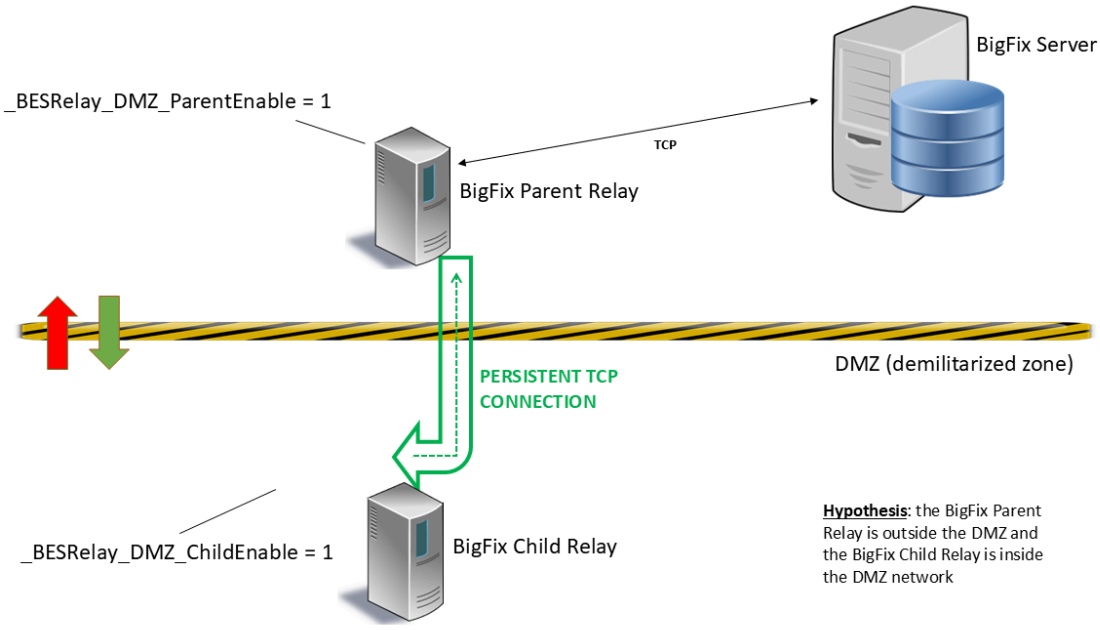
The capability to establish a persistent TCP connection between the parent relay in the more secure zone and its child relay inside the DMZ network was added to the product. This allows you to manage systems in a demilitarized zone (DMZ network).

In an environment where a relay in DMZ reports to a parent relay within its intranet network, it can be assumed that all communications between intranet and DMZ pass through a firewall that does not allow any upstream communication. In this case, any attempt for the child relay in the DMZ to initiate communication with its parent relay will fail.

This restriction is overcome by establishing a persistent TCP connection between the parent relay and its child relay inside the DMZ. The persistent connection is always initiated by the parent relay. The communication cannot be initiated by the child relay due to network restrictions.

Overview

The following picture displays the persistent TCP connection established between parent relay and child relay:



In the picture are displayed:

- In green: The persistent TCP connection established between the parent relay located in the more secure zone and the child relay located in the demilitarized zone.
- In yellow and black: The line of the demilitarized zone (DMZ network).

Enabling persistent connections on both parent and child relay

On a child relay where the BigFix client was not registered on the BigFix server yet

1. Log in to the BigFix Console.
2. Run the `Relays in DMZ: Enable Parent Relay and set Child Relay List` Fixlet on the parent relay computer:

   Note: Before running the Fixlet, you must specify in the text field of the Description tab the list of child relays allowed.

3. Manually install the BigFix client on the child computer. For more details, see Installing the Client on Windows and Installing the Client on Linux.
4. Manually install the BigFix relay on the child computer by downloading the appropriate package depending on your operating system from the following web site: http://support.bigfix.com/bes/release/

   Note: In a typical scenario, run the Fixlet first on the parent relay and then manually configure the child relay.

5. On the child computer, ensure that the client and relay processes are stopped.
6. On a Windows child relay, add the `HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\BigFix\EnterpriseClient\Settings\Client\_BESRelay_DMZ_ChildEnable` key to the Windows registry and set its string REG_SZ value to 1.
7. On a Linux child relay, if the besclient.config file does not already exist, make a copy of the file named besclient.config.default located in the /var/opt/BESClient/ directory

and rename it into besclient.config. Manually edit the besclient.config by adding the following new section:

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

8. Restart first the relay process.
9. At least one minute after restarting the relay process, restart the client process.

Note: If your parent relay was configured as an authenticating relay, it might be necessary to temporarily disable the relay authentication to allow the child relay to register successfully for the first time. Enable again the relay authentication after your child relay was registered successfully.

On a child relay where the BigFix client was already registered on the BigFix server

1. Log in to the BigFix Console.
2. Run the `Relays in DMZ: Enable Parent Relay and set Child Relay List` Fixlet on the parent relay computer:

   Note: Before running the Fixlet, you must specify in the text field of the Description tab the list of child relays allowed.

3. Run the `Relays in DMZ: Enable Child Relay` Fixlet on the child relay computer:

   Note: In a typical scenario, run the Fixlet first on the parent relay and then on the child relay.

4. Both Fixlets will restart the relay process.

Establishing a persistent connection

The parent relay will try to open a socket to the child relay at port 52311.

The child relay can "grab" the socket used by the parent to communicate with it and keep it alive by sending ping messages periodically. At the same time, the child relay will start to listen on a different port such as 52312 only on its loopback address, this will be used to forward all the traffic through the socket opened by the parent that was previously grabbed.

All requests coming to the child relay that must be propagated upstream (for example during the registration of a client below the child relay or for reporting purposes) will be internally routed to the loopback address to be sent to the parent relay within the intranet.

Communicating on the persistent connection

To achieve the requirement, the parent relay initiates a communication with its own child relay and keeps the connection standing and persistent to, later on, use it from the child relay to the parent relay when upstream communication is needed by the child relay.

Managing persistent connections

You can manage the Relays in DMZ persistent connections by configuring a few settings. For details, see Relays in DMZ (on page 895).

Chapter 20. Working with PeerNest

The BigFix Client includes a new feature named PeerNest, that allows to share binary files among Clients located in the same subnet. The feature is available starting from BigFix Version 9.5 Patch 11.

A practical use case is a branch office connected to the data center through a slow link: with earlier BigFix versions, the suggested configuration required a Relay to be installed in the branch office in order to download and cache large payloads.

With PeerNest, the BigFix Clients can share downloaded binaries and therefore reduce the number of communications going outside of the branch office even if a Relay is not installed locally. In this way, multiple Clients generate on the Relay the download load of a single Client, because only one Client downloads from the Relay and then shares the download with the peers.



Note: Up to BigFix 10.0.7, the PeerNest feature does not work if it is enabled on BigFix clients residing on a subnet hosting also clients belonging to a different BigFix deployment. Starting from BigFix 10.0.7, peer requests coming from clients belonging to different deployments are ignored and so this limitation is no longer applicable.

Use of PeerNest can help reducing the number of Relays in some complex BigFix deployment scenarios, therefore reducing infrastructural costs.

An introduction video can be found here: https://www.youtube.com/watch?v=tXRX3zlw1aQ.

Enabling PeerNest

To enable the PeerNest feature, set to 1 the following configuration setting on the Client:

```
_BESClient_PeerNest_Enabled = 1
```

The Client enables all the PeerNest feature in order to locally optimize the download of binaries.

This configuration setting requires a restart of the Client to be effective.

Note: If you require BigFix to optimize the download of the binaries required to execute actions, ensure that the hash of the file is specified inside the *prefetch* statement.

PeerNest configuration settings

For details about all available settings, see Peer to peer mode (on page 915).

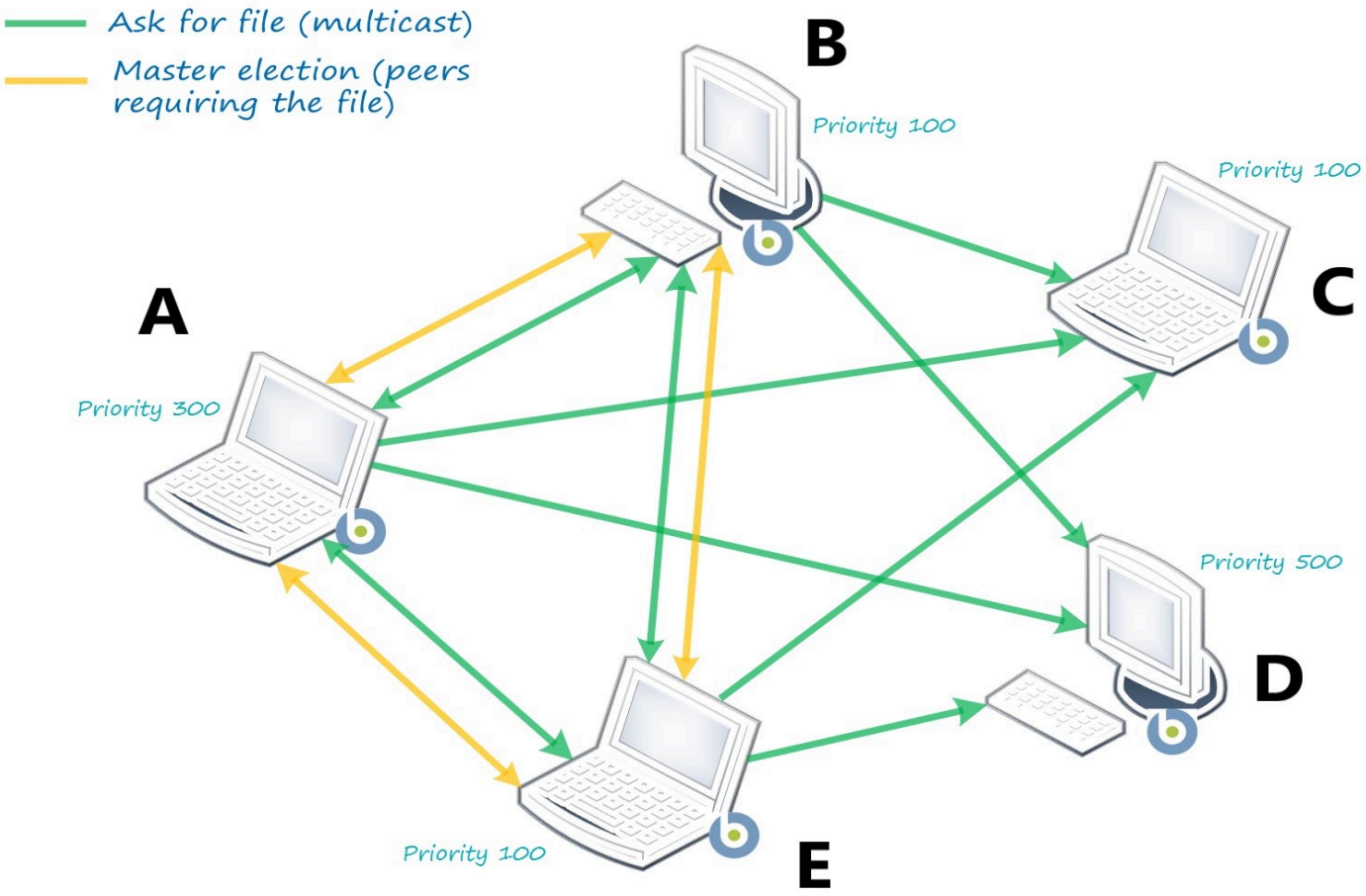
PeerNest in depth

When multiple Clients are executing actions requiring the prefetch of a binary file, they check with their peers if the file is already cached in the subnet. If the binary was not cached then the Clients can elect one of them as responsible for downloading from the Relay (**Figure 1**): the peer with the highest priority, among the peers requiring the file, will manage the download; if all peers have the same priority, the computer with the lowest ID will download the file from the Relay. By doing so, for a given action, all files will likely be downloaded from the Relay by the same computer, and so the downloads will be serialized (one file at a time), thus ensuring minimum bandwidth occupation in the link between Relay and computer.

The following figure shows in details the master election process:

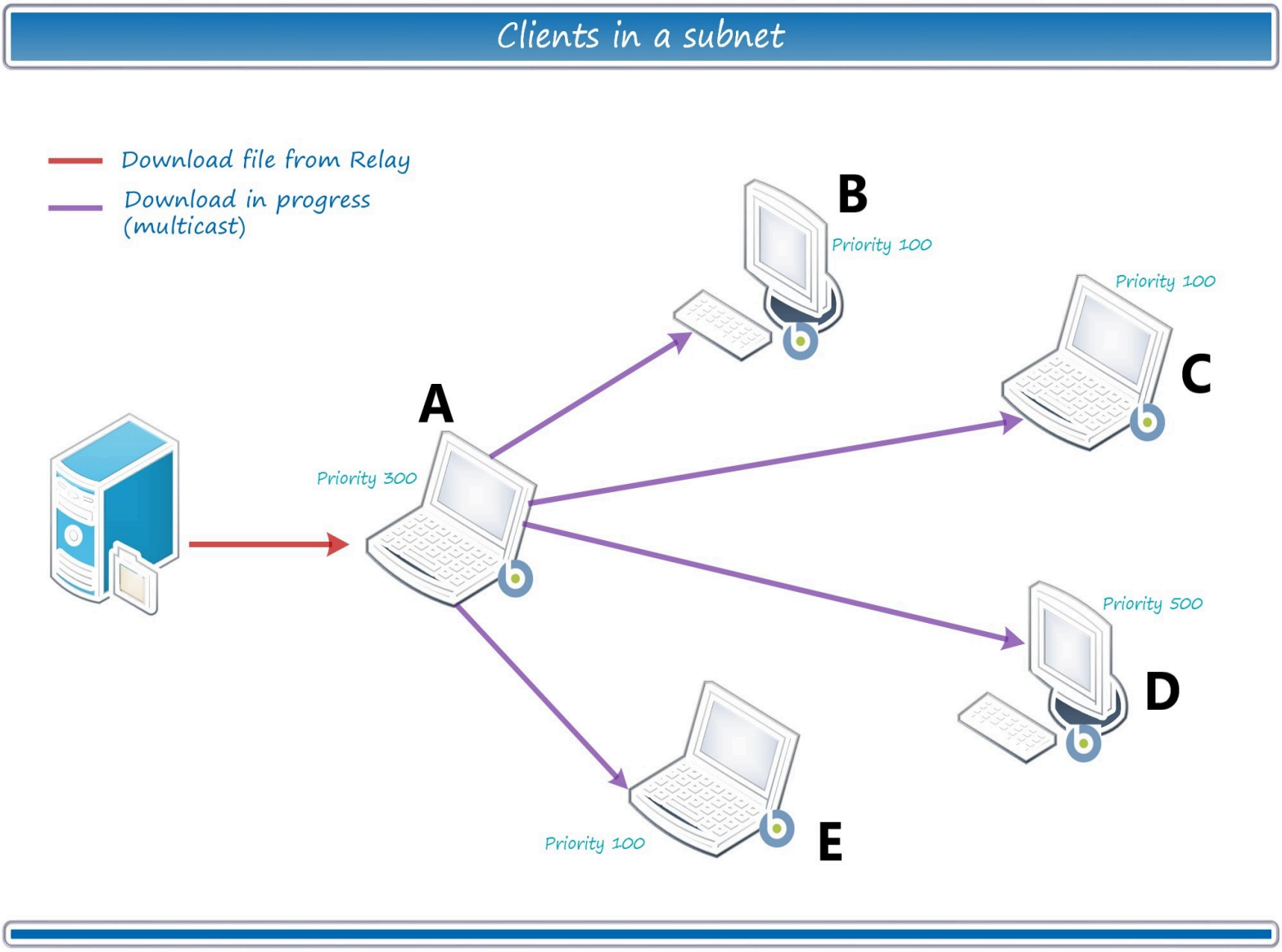
- **A** will be elected as master as it has the highest priority among the peers requiring the file (**A**, **B**, **E**).
- **C** and **D** do not require the file, so even if **D** has higher priority than **A**, **D** is not involved in the master election process.

Figure 1: Master election



As soon as the elected master (**A**) starts downloading the file from the Relay (**Figure 2**) it notifies the other peers that there is a download in progress, so they wait for it without taking further actions. The master sends periodic notifications about the download.

Figure 2: Download in progress

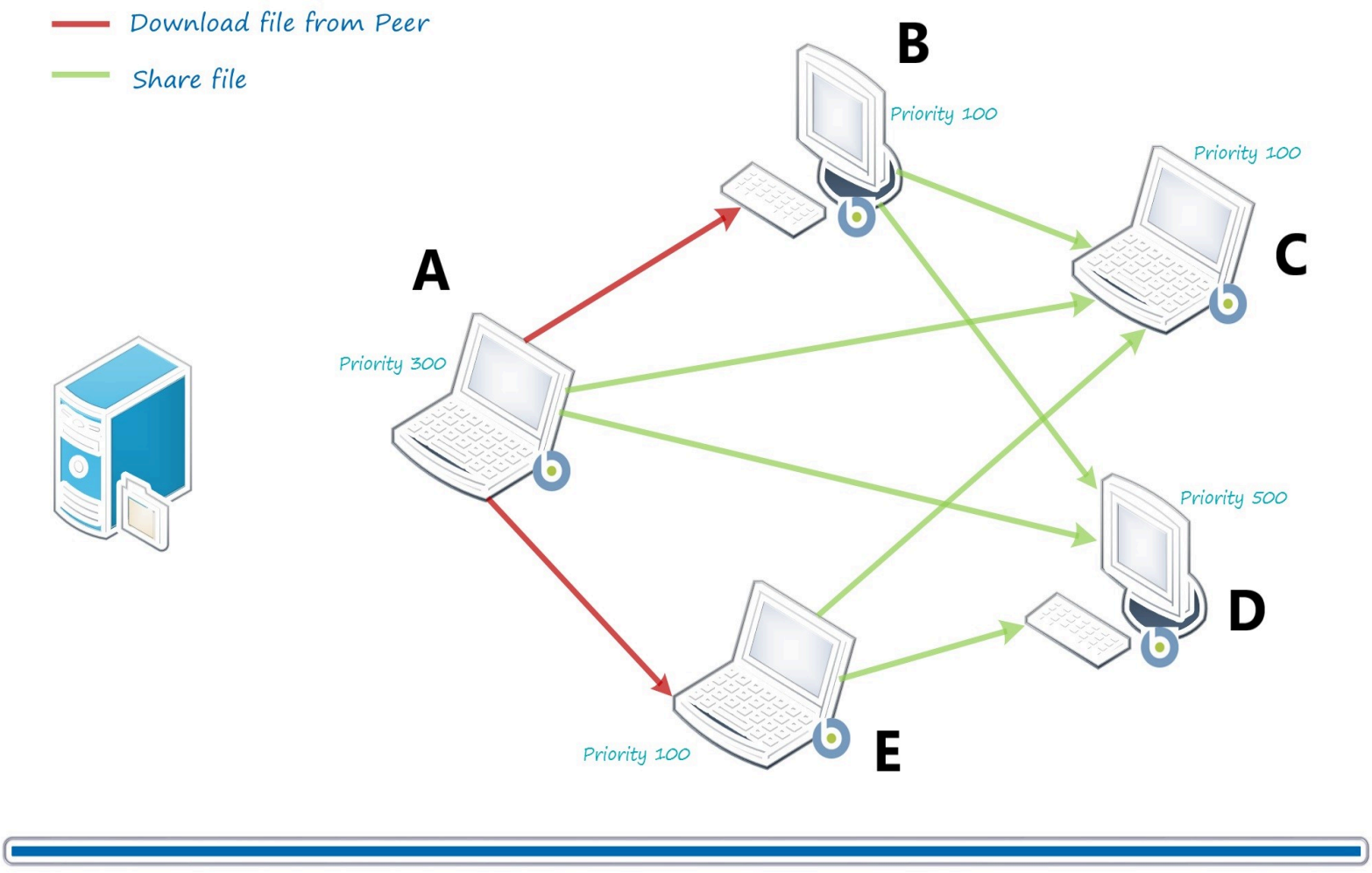


If the peers did not receive the download in progress message (download failed, Client down, network issue), a new election process is started over and another peer becomes master.

When the elected master finishes the download, it moves the file into the PeerNest cache and notifies the other peers about its availability; the peers interested in the file then start downloading it from master and share it, using the same mechanism (**Figure 3**). In this way, multiple Clients generate on the Relay the download load of a single Client, because only one Client downloads from the Relay and then shares the download with the peers.

The following figure shows how the peers interested in the file (**B**, **E**) start downloading it directly from peer **A**, instead of downloading it from the Relay. When their download completes, the file is cached to be available for future usage: so the Clients **A**, **B** and **E** will share the downloaded file with **C** and **D**.

Figure 3: File sharing



Clients with peer enabled will start up an HTTP Server listening on port 52311 for peer transfers. Each Client can serve at the same time a maximum of `_BESClient_PeerNest_MaxActiveFileDownloads` other peers (default value is 5).

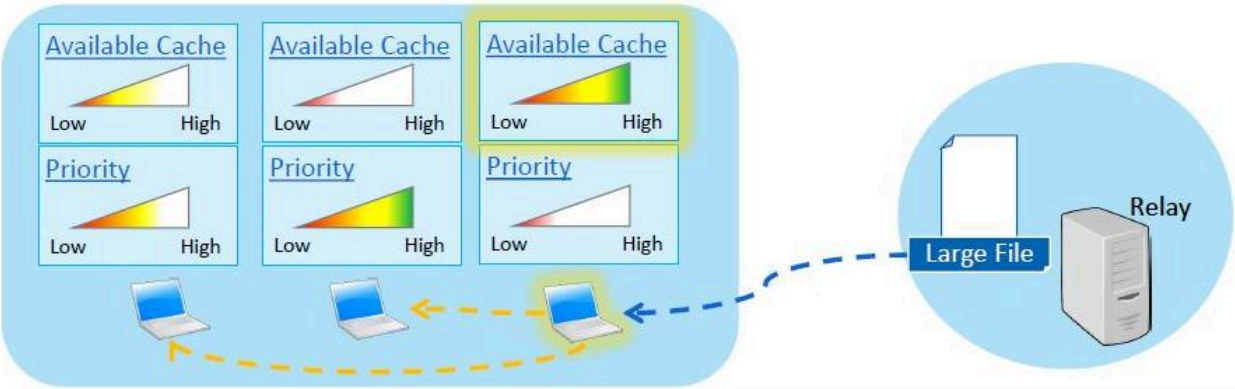
The Client priority comes also into play when there are 2 or more peers available to share the same file. The Client that wants to download the file creates a memory list of peers serving the file; it will pick up the peer randomly with weighted probability, based on priority: for instance, if the memory list is made up of two peers, say C1 with priority W, and C2 with priority 2W, picking C2 will be twice as likely as picking C1. In this case, the legacy retry-behavior applies – ruled by the `_BESClient_Download_RetryMinutes` and `_BESClient_Download_RetryLimit` Client settings – with the following addition:

1. The peer from which a failed download attempt has occurred gets removed from the memory peer list (unless that peer is the only one in memory), and so the next attempt will be done with another peer.
2. If the retry-count reaches the limit, the Client will go to download the file directly from the Relay.

If the Client fails to download the file because the peer that it has tried to connect to is already serving 5 downloads (default case), the retry-count is not increased and the peer is not removed from the memory peer list.

PeerNest behavior for files bigger than the PeerCache size

Starting from Version 10 Patch 2, a new feature allows time optimization for actions that require prefetching of large files. This feature ensures that the peers elected for downloads are those with real ability to store files rather than simply those with highest priority.



In fact, prior to Version 10.0.2, in a PeerNest configuration, when BigFix Clients are performing actions requiring the prefetch of a binary file, they check if the file is already cached in the subnet. If the binary file is not cached yet, the BigFix Clients can elect one of them as responsible for downloading from the BigFix Relay and then sharing the file with the peers. The elected peer is generally the one with the highest priority assigned. If the elected peer does not have enough cache to store the payload, the distribution fails, and all the other peers wait needlessly.

Starting from Version 10 Patch 2, with the `_BESClient_PeerNest_UseNoSpaceDownload` setting, you can configure the PeerNest to elect for downloads only the peers with a cache limit higher than the payload size. The same option forces the peers with high priority but not enough cache limit in passive mode. In this way, you avoid that the peers have a useless waiting and the file is downloaded sequentially. With this option, you also provide the peers with the ability to download the file directly from the BigFix Relay without waiting for the other peers that might eventually fail.

Excluding a list of subnets

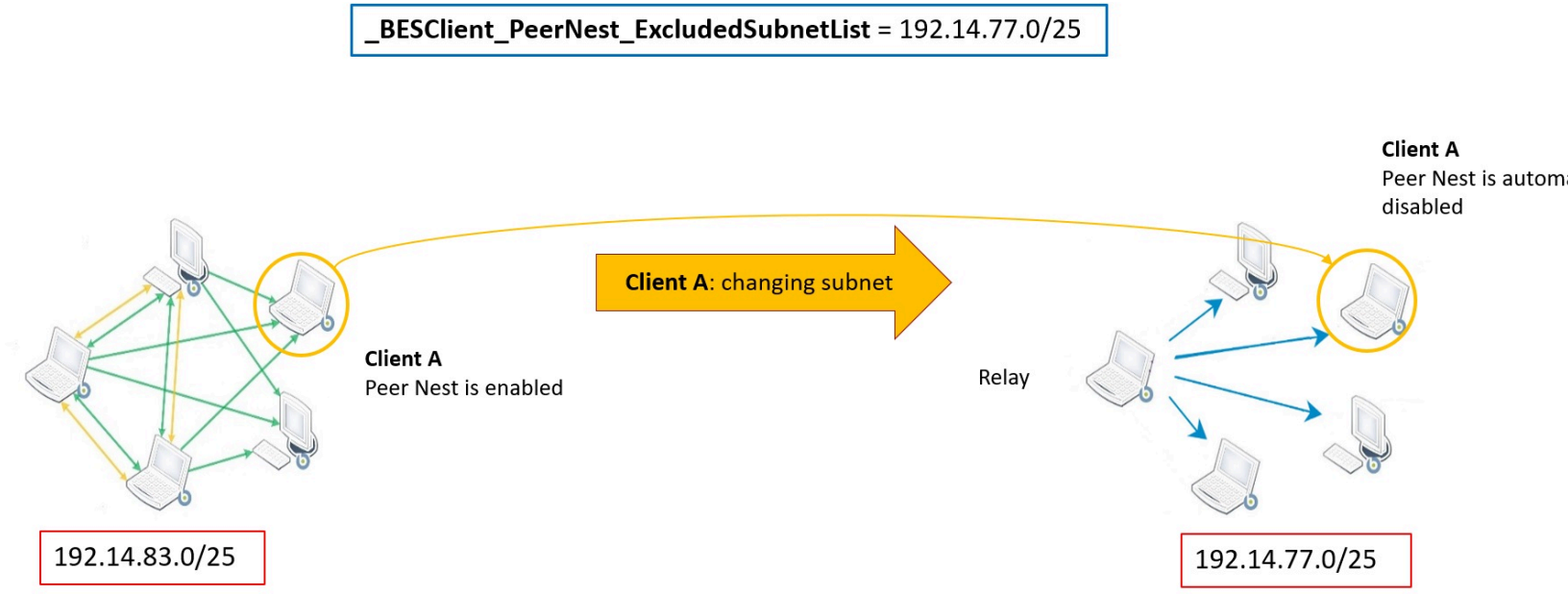
Starting from Version 10 Patch 4, a new feature allows you to disable the PeerNest mechanism for BigFix Clients connected to a specific subnet.

The devices that use the PeerNest mechanism, send and receive messages using the UDP communication protocol. When you have many devices connected on the same subnet, they share network traffic among themselves. This might cause an excessive consumption of network resources. According to your specific needs, you might want to limit the UDP traffic

on some specific network subnets. For example, if you have BigFix Clients running in a VPN infrastructure.

Starting from Version 10 Patch 4, with the `_BESClient_PeerNest_ExcludedSubnetList` setting, that contains a list of subnets (for example: 192.1.77.0/25;192.1.77.129/25), you can exclude from the PeerNest mechanism the Clients that belong to one of the subnets specified in the list. When a Client with PeerNest enabled (`_BESClient_PeerNest_Enabled` = 1) registers with the BigFix Relay, the new feature identifies the subnet of the Client and checks if its subnet belongs to this list. If yes, then the PeerNest mechanism is disabled for the Client.

Moreover, when you need to change the Client subnet, for example when you connect to a VPN, if the new subnet is already included into the Exclusion Subnet List, the P2P mechanism is automatically disabled like shown in the following figure.



Adding/modifying/deleting the value of the new setting requires a restart of the BigFix Client.

Best Practices

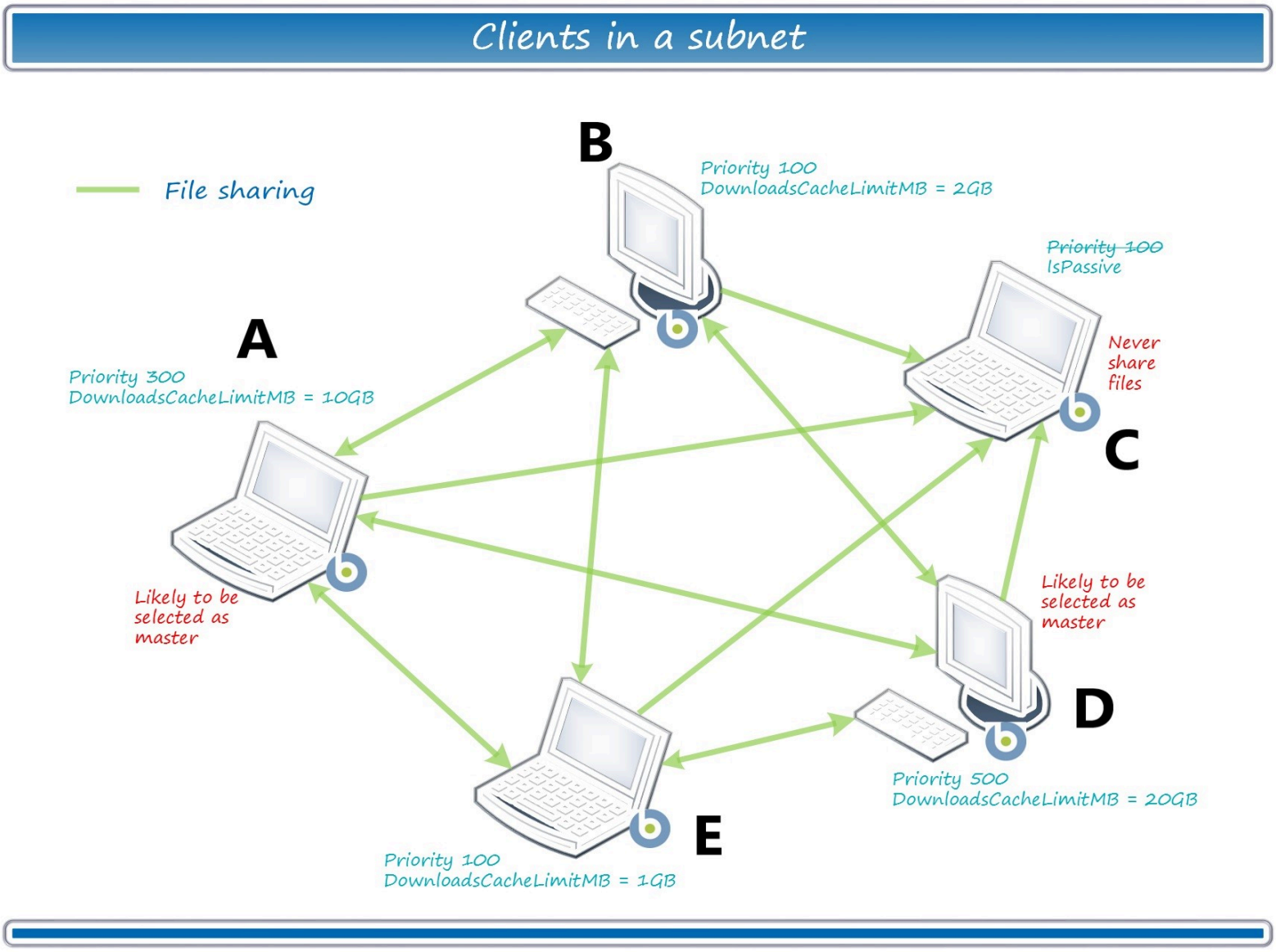
A good practice is to assign higher priorities to computers with a better link to the Relay and enough resources to serve the other peers of the subnet (and possibly a steady power supply).

PeerNest requires an increased disk storage space for caching files. The default PeerNest cache size is 2GB, which is enough for many scenarios; it must be increased in case of transfer of large files (patch management, software distribution, etc.) in order to fit the cache. The PeerNest cache is intended as a temporary storage, so usage and lifetime can be fine tuned with the following parameters: `_BESClient_PeerNest_DownloadsCacheLimitMB`, `_BESClient_PeerNest_MinimumDiskFreeMB`, `_BESClient_PeerNest_MinimumCacheDays`, `_BESClient_PeerNest_MaximumCacheDays`.

PeerNest requires UDP (port 52311) communication to be enabled, in order to allow the BigFix Clients to communicate with each other.

It also requires TCP (port 52311) to allow the BigFix Clients to download files from a peer, and the subnet supporting multicasting.

It is recommended to set PeerNest in passive mode (using the `_BESClient_PeerNest_IsPassive` configuration setting) on Clients that cannot open this port or do not want to use additional disk space for caching. Passive Clients will only download from the other peers but will not share content. The following figure shows an example of configuration.



Bandwidth Throttling

Clients with peer enabled will start up an HTTP Server; the other peers can connect to it for downloading.

Each Client serving files to the other peers can control the amount of bandwidth allocated for that purpose. The `_BESClient_HTTPServer_ThrottleKBPS` setting defines the total number of kilobytes that the Client gives to all of the peers combined per second (0 means

no limit): if its value is 1000 KB/sec and there are 10 peers downloading simultaneously, the Client will send data to each at 100 KB/sec (for a total of 1000 KB/sec).

PeerNest behavior when using an external Certificate Authority (CA)

In a subnet where there are Agents using client certificates issued by a certificate authority set by the customer, and Agents using the original BigFix client certificates, the behavior of the PeerNest feature may deviate from what is normally expected, since the former are not able to download files from the latter, and vice versa.

For example, if a prefetch actionscript command targets Agents in the described situation, it may happen that more than one Agent downloads the file from the parent Relay, and that some of the Agents - typically sharing the same type of client certificate - obtain the file with delay.

When using the PeerNest feature in a BigFix deployment where an external certificate authority has been set, it is therefore recommended, where possible, to try to have subnets populated by Agents that share the same type of client certificate.

For details about how to set an external certificate authority in a BigFix deployment, see Configuring the root server with a custom CA (on page 1207).

Troubleshooting scenario 1

On BigFix Clients hosted on an Operating System that has the Internet Protocol version 6 (**IPv6**) disabled or not configured:

If you want to use the PeerNest feature, you must:

1. Set on these Clients the `_BESClient_Comm_IPCommunicationsMode` configuration setting as follows:

   ```
   _BESClient_Comm_IPCommunicationsMode = OnlyIpv4
   ```

2. Restart the Clients for the changes to take effect.

Troubleshooting scenario 2

On BigFix Clients that have an active polling set using the `_BESClient_Comm_CommandPollEnable` and `_BESClient_Comm_CommandPollIntervalSeconds` configuration settings:

If you want to use the PeerNest feature, you must not configure these Clients to be "passive" PeerNest agents. Do not enable on them the `_BESClient_PeerNest_IsPassive` configuration setting. Otherwise, depending on the timing of the polling, multiple Clients in a subnet can download the same binary, without sharing it.

Peer-to-Peer (PeerNest) Site Gather

Starting from BigFix 11.0.6, PeerNest can be extended to support site gathering in addition to file downloads. When this capability is enabled, BigFix Clients on the same subnet can obtain complete BigFix sites from neighboring peers instead of individually downloading them from their Relay.

This approach reduces Relay traffic and WAN usage, improves distribution speed within branch offices, and lowers infrastructure load while maintaining version correctness for frequently updated content. When PeerNest Site Gather is enabled, the client modifies the traditional site gathering workflow. Before contacting the Relay, the client sends a multicast UDP request over the local subnet to determine whether another peer already has a newer version of the required site or is currently gathering it. If a peer reports that it has a newer version available, the requesting client gathers the site directly from that peer.

If a peer indicates that it is in the process of gathering the site from the Relay, the requesting client waits for completion and then gathers from that peer. If no peer responds, the standard gather process occurs and the highest-priority client retrieves the site from the Relay. Once the gather completes, the client notifies the subnet so that other peers can retrieve the site locally.

New UDP message types are introduced to support site availability requests, responses, gather-in-progress notifications, and gather-complete notifications. A message type identifier differentiates site gather messages from existing P2P file download messages while preserving backward compatibility.

Clients participating in PeerNest run a local TLS-enabled HTTP server on port 52311. With Site Gather enabled, this server can provide the site FileList and individual site files to requesting peers. All connections use TLS with client certificate authentication. Unlike P2P file downloads, site gather does not use the PeerNest cache. Instead, sites are served directly from the client `__BESData` directory where gathered content is already stored. PeerNest Site Gather supports External Sites, Custom Sites, and Action Sites, including the Master Action Site and Operator Sites. Mailbox Sites are excluded because they are specific to individual clients and cannot be shared. Before serving a site to peers, the client verifies with its parent Relay or Server that no newer version is available. If a newer version exists upstream, the client does not respond to the peer request, ensuring that frequently updated operational content is not distributed in an outdated state.

During the process, clients transition through logical states that reflect whether they are waiting for responses, gathering content, waiting for another peer to complete gathering, or preparing to gather from a selected peer. Only one client per subnet gathers from the Relay when necessary, based on the existing priority mechanism used by PeerNest file download.

Other clients then retrieve the site from that peer once it becomes available.If gathering from a peer fails, the client retries with another available peer. After four failed attempts, the client falls back to the traditional Relay-based gather process. If a peer is temporarily busy because it is serving the maximum number of concurrent downloads, the client retries later without increasing its failure count. This ensures reliability while preserving the ability to revert to the Relay when required.

PeerNest must already be enabled for file downloads before enabling Site Gather.

The following client settings apply:

- `_BESClient_PeerNest_EnableGather` Enables P2P site gather functionality. Default value is 0 (disabled).
- `_BESClient_PeerNest_IsPassive` Allows the client to gather from peers but not serve sites. Default value is 0 (disabled).

When `_BESClient_PeerNest_EnableGather` is set to `1`, the client participates in P2P site gathering. Passive peers can download from others but will not act as site servers. PeerNest must already be enabled for file downloads before enabling Site Gather.

The setting `_BESClient_PeerNest_EnableGather` controls whether site gather through PeerNest is active and is disabled by default.

The setting `_BESClient_PeerNest_IsPassive` allows a client to gather from peers without serving content to others. When enabled, passive peers participate in requesting content but do not respond to peer requests.

Older clients ignore site gather–specific UDP messages and continue operating using traditional P2P file download behavior. Newer clients process both legacy and updated message formats, allowing mixed-version environments to function correctly. All peer communications use TLS with mutual certificate authentication. Content is served only from controlled client directories, and no additional ports are required beyond the standard PeerNest configuration. By extending PeerNest to support site gathering, organizations can further reduce Relay load, accelerate content propagation within subnets, and improve scalability while maintaining security and content integrity.

For details about these settings, see Peer to peer mode (on page 915).

Configuring PeerNest without the multicast function

Up to BigFix Platform 11.0.3, the PeerNest implementation is based on multicast UDP messages exchanged between the BigFix Clients (peers). In this old implementation, the BigFix Relay is completely unaware of PeerNest.

Starting from BigFix Platform 11.0.4, in the new PeerNest implementation, the BigFix Relay takes active part in the PeerNest architecture by acting as a file request broker. Each peer (BigFix Client) that needs to download a file asks the BigFix Relay for information about that specific file. The Relay will, then, keep track of "which BigFix Client has what" in the BigFix database by acting as a "custom tracker".

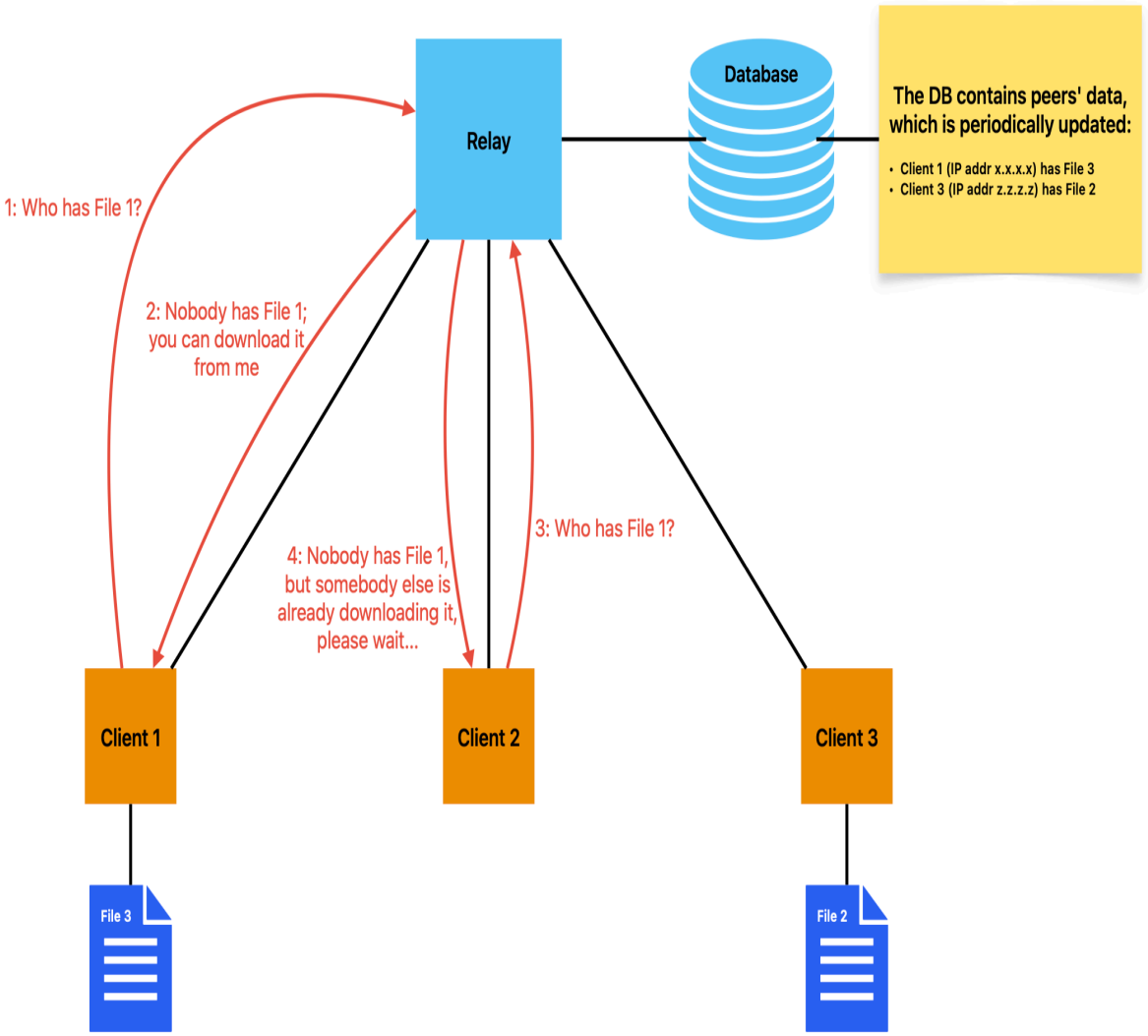
Enable the new PeerNest implementation by setting the `_BESClient_PeerNest_DisableMulticast` configuration parameter to 1 (default value is 0) on the BigFix Clients. A restart is required for the change to be effective. Clients that have multicast disabled (this configuration parameter set to 1) will not send any multicast messages but they will still be able to receive multicast messages from other Clients (if the

network configuration allows to do so) and answer with unicast UDP messages, to ensure backward-compatibility with the old PeerNest implementation.

In this new PeerNest implementation:

1. Clients will request to download files from the Relay; if the file is not available on any peer, the Relay will wait for a defined amount of time before answering, by default 30 seconds and will reply to the client with higher priority that requested the file, by allowing the client to download the file from the Relay. In the meantime, the other clients will be replied that the file is being downloaded by another peer, so they will keep asking for the file periodically until the first client completes the file download.
2. The downloading client will notify the Relay that it completed the downloading and caching. The Relay updates the information in the database. The Relay replies to the other clients, which were still waiting for the file, by providing information about the client that is ready to share the file with them.
3. The PeerNest network can increase the number of clients which share information with the Relay and files with the other peers. BigFix Clients (peers) must be located in the same subnet and they can share files among each other only if they are direct descendants of the same BigFix Relay.

The following picture shows a sample scenario in which three clients start downloading and sharing a file (File 1), while they are already sharing some previously downloaded files.



BigFix Peer-to-Peer Site Gather (No-Multicast)

Starting from BigFix Platform 11.0.6, by combining the multicast disabled configuration and the `_BESClient_PeerNest_EnableGather` setting, it is possible to use peers in no-multicast mode for site gathering. In this state the BigFix Relay acts as the central orchestrator for site distribution within a subnet. This mechanism prevents bandwidth saturation by ensuring only a single "leader" client, selected based on priority or Computer ID, downloads the site from the Relay, while other peers wait to pull the data locally.

In this new PeerNest implementation:

1. When a client requires a site update, it sends a gather request to its parent Relay.
2. If the Relay determines the site version isn't yet available on any local peer, it pauses for a predefined interval. It then designates the client with the highest "entitlement" (priority) to perform the gather.
3. All other requesting clients receive a "Download In Progress" status from the Relay and enter a Waiting state.
4. The designated leader downloads the site from the Relay. Once finished, it sends a message to the Relay to register itself as a local source for that specific site version.
5. The Relay sends a UDP Ping to all clients currently in the waiting state, signaling that the data is ready.
6. The waiting clients resubmit their gather requests. Instead of sending the files directly, the Relay provides a list of peers (the leader) that have the site cached, allowing the clients to complete the gather locally.

PeerNest statistics: Exporting a JSON file with aggregated data

Starting from BigFix Version 11.0.6, the BigFix Platform supports the collection of data to monitor the PeerNest activity, both prefetch and gather activities.

The BigFix Server collects the PeerNest activity information from all BigFix Clients 11.0.6 and later versions, and exposes the aggregated data through some REST APIs in JSON format. For more information about the available APIs, see PeerNest.

Using the new `_BESClient_PeerNest_StatisticsEnabled` configuration setting, you can manage the PeeNest information collection on the BigFix Client. For more information about this setting, see Peer to peer mode (on page 915).

Chapter 21. Archiving Client files on the BigFix Server

You can collect multiple files from BigFix clients into an archive and move them through the relay system to the server.

This allows the BigFix Administrator to automatically log data from specific managed computers.

To do this, a new component called the **Archive Manager** has been added to the BigFix Client which can collect files periodically or on command. It passes the resultant compressed tar-ball to the **Upload Manager** on the BigFix Client. The Upload Manager has an input directory that queues the files for uploading.

Additionally, a new component called the **Scanner Manager** has been introduced since Patch 4 to support real-time archive uploads in BigFix Platform for on-demand data collection from external tools. The Scanner Manager facilitates the immediate collection and upload of data, streamlining the process of retrieving data from the external sources.

The Upload Manager performs one upload operation at a time, moving the data in manageable chunks to reduce network traffic. It sends these chunks to the nearest BigFix relay or server, where the **PostFile** program reassembles the chunks back into the original file.

PostFile then passes the file up the chain, to the next BigFix relay or to its ultimate destination at the BigFix server. It again uses the Upload Manager to slice the file into chunks and send them on to the next PostFile program in the hierarchy. When the file finally arrives at the BigFix server, it is saved in a special directory location based on the ID of the client computer. Along the way, both the Upload Manager and the PostFile program can alter the chunk size or throttle the upload speed to smooth out network traffic.

For information about configuration settings related to these components, see Archiving client files (on page 472).



Note: When it encounters an unregistered BigFix Client, the Upload Manager pauses. This can happen for a variety of reasons, including a downed network, a

busy server, or a disconnected client. As soon as the BigFix client can register with the BigFix system again, it restarts the Upload Manager and continues from where it stopped.

Archive manager settings

A typical archive is a collection of logs and configuration files that are compiled regularly and posted to the server. There are many settings available to help you customize your logging needs.

For details about the configuration settings related to this component, see Archive Manager (on page 472) in BigFix Configuration Settings (on page 403).

Creating a Custom Action

You can create custom actions that can post attributes about the BigFix client to an archive file.

To create a custom action:

1. Start the BigFix Console.
2. Select the **Computers** tab.
3. From the filter/list, select the set of computers that you want to target for the action.
4. Select **Take Custom Action** from the **Tools** menu.
5. Select the **Action Script** tab.
6. Enter the desired **Action Script** in the text box provided.

Archive Manager

Archive Manager is a component of the BigFix Client that can collect files periodically or on command. It passes the resultant compressed tar-ball to the Upload Manager on the BigFix Client.

For details about the configuration settings related to this component, see Archive Manager (on page 472) in BigFix Configuration Settings (on page 403).

Archive Manager internal variables

These are the internal variables of the Archive Manager component.

_BESClient_ArchiveManager_LastArchive

The Archive Manager updates this setting whenever it posts an archive. The value of the setting is the secure hash algorithm (sha1) of the file that was posted.

__BESClient_ArchiveManager_LastIntervalNumber

The BigFix Client updates this setting whenever it posts an archive. It represents the interval number from 1970 to the time when the archive was last collected. If the interval is a day long (the default), then the setting indicates the number of days from 1970 to the day when it created the last archive. It is calculated such that when the interval number changes, it is time to create a new archive.

The value is also offset by a time corresponding to the computer id, to stagger the collecting of archives.

Archive Manager Index File Format

During the building of the archive, the Archive Manager creates an index containing metadata about the archive.

This is a sample index from an archive with a single file:

```
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
UTF8URL: file:///c:/temp/log/newfile.log
NAME: (LOG)newfile.log
UTF8NAME: %28LOG%29newfile.log
SIZE: 105
TYPE: FILE
HASH: 3a2952e0db8b1e31683f801c6384943aae7fb273
MODIFIED: Sun, 14 Mar 2004 18:32:58 +0000

--===--
```

Archive Manager Server Index File Format

When the BigFix Server receives files from the Archive Manager, it creates a server index containing metadata about all files present on the Server.

The server index file is created when you have enabled one or more settings named **_BESClient_ArchiveManager_KeepFileSet-<tag>** on the BigFix client.

This is a sample index from an archive with a single file:

```
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
KeepFileSet-(LOG): 1

--===
URL: file:///c:/temp/log/newfile.log
UTF8URL: file:///c:/temp/log/newfile.log
NAME: (LOG)newfile.log
UTF8NAME: %28LOG%29newfile.log
```

```
SIZE: 105
TYPE: FILE
HASH: 3a2952e0db8b1e31683f801c6384943aae7fb273
MODIFIED: Sun, 14 Mar 2004 18:32:58 +0000

--===--
```

Upload Manager

The Upload Manager coordinates the sending of files in chunks to the Post File program. You can throttle the upload dataflow to conserve bandwidth. The file system uses 64-bits, sufficient for file sizes of up to 2^{64} – 1 bytes in length.

For details about the configuration settings related to this component, see Upload Manager (on page 489).

PostFile

The PostFile program receives the chunks of files posted by the Upload Manager and appends them to its own copy of the file. The Upload Manager specifies the range of bytes being posted and the sha1 of the file, which is used as the filename. For details about the configuration settings related to this component, see Post File (on page 483) in BigFix Configuration Settings (on page 403).

These parameters are appended to the URL as in the following example:

```
postfile.exe?sha1=51ee4cf2196c4cb73abc6c6698944cd321593007&range=1000,1999,
20000
```

Here the sha1 value identifies the file, and the range in this case specifies the second 1,000 byte chunk of a 20,000 byte file.

When PostFile receives a chunk of the file it first checks to make sure it is the correct segment. If so, it appends the posted data to its local copy of the file. It returns the size of this file, as well as the current chunk size and throttle BPS settings.

PostFile has to handle several BigFix clients feeding into it at the same time. To balance that load, it adjusts the throttle rate. The effective throttling rate is determined by dividing the limiting PostFile rate by the number of concurrently uploading files.

For example, if PostFile has a throttle setting of 100 KBPS and 50 clients are simultaneously uploading files, the throttle value returned to each client would be adjusted to 2 KBPS. By setting custom throttle values to specific BigFix relays, you can efficiently deal with any bottlenecks in your network.

PostFile stores the partially uploaded files in the Upload Manager's buffer directory with an underscore in front of them (the Upload Manager does not upload files that begin with underscore). When PostFile receives the last chunk of the file, it calculates the sha1 of the file and checks that it matches the sha1 parameter in the URL. If so, it removes the leading underscore.

The Upload Manager can then upload the file to the next relay up the hierarchy (or any other server, if so specified).

PostFile determines whether or not the Upload Manager is running. If not, PostFile assumes that it has reached its root server destination. It renames the uploaded file, extracts the files from the archive, and deposits them in a subfolder of the Upload Manager's buffer directory.

The program calculates the subfolder path using a modulus of the computer ID. This has the effect of spreading out file directory accesses and preventing an overpopulation of any single directory.

For example, the path to file "log" from computer ID1076028615 is converted to the path "BufferDir/sha1/**15**/1076028615/log" where 15 is the remainder modulo 100 (the lower two digits) of the id.

If the uploaded file is a valid BigFix archive and is successfully extracted, then the original uploaded file is deleted.

Resource Examples

Some concrete examples.

Example 1

In this example, we want to collect all the files in the `c:\log` folder and all the .ini files in the `c:\myapp` folder once an hour. Send up only the differences and do not send the archive if it exceeds 1,000,000 bytes in size. To set this up, create the following settings in the BigFix Console:

```
_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
```

Example 2

In this example, we want the same set of files as above, but we also want to collect some useful attributes (retrieved properties) from the client computer. A custom action can generate these attributes and trigger an archive when it completes. It uses the same settings as above, but sets the operating mode to 2 to enable the **archive now** action command:

```
_BESClient_ArchiveManager_OperatingMode = 2
```

You can then create a custom action, specifying the attributes you want to collect. For example, to append the operating system, computer name, and DNS name to the log file, create a custom action like this:

```
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
```

The **appendfile** command creates a temporary text file named **__appendfile** . Each time you invoke the command, it appends the text you specify to the end of this temporary file.

The **delete** and **copy** commands clear out the old log file (if any) and copy the __appendfile to the log. This has the effect of creating a new properties.log file. The **archive now** command immediately creates an archive, as long as the OperatingMode is set to 2.

You can then target this action to any subset of BigFix Clients, using whatever scheduling you choose. Using variations on this scheme, you could perform a full archive once a week, in addition to nightly differences.

Chapter 22. BigFix Configuration Settings

A number of advanced BigFix configuration settings are available that can give you substantial control over the behavior of the BigFix suite. These options allow you to customize the behavior of the BigFix server, relays, and clients in your network.

Overview

The configuration settings apply to the BigFix server, relays, and clients. You can administer them through the Custom Settings configuration in the console's **Computer Status** dialog.

- Settings take on their default values unless you modify them.
- If you specify an invalid value for a setting, it reverts to its default value.
- All configuration values are stored as strings in the registry (or a configuration file).
- To prevent older settings from overriding newer settings, each setting has an "effective date" associated with it. An action with an older effective date does not overwrite settings with newer effective dates. Effective dates are set to the time that the action was taken.
- Numeric values are stored as strings, and assumed to fit in unsigned integers with a maximum of 32 bits (max value is 4,294,967,296).
- Numeric values that are negative, greater than the maximum value, or that contain non-numeric characters are treated as invalid values. The settings revert to their default values in such cases.
- Boolean values are stored as strings - either as *"1"* or *"0"* corresponding to *true* and *false* respectively. Boolean values that do not contain either of these allowed values are treated as invalid. The settings revert to their default values in such cases.



Warning: Use the configuration settings with caution. If misused, they can cause non-optimal behavior or prevent BigFix from functioning properly. When in doubt, consult with your support technician.

For details, see BigFix Configuration Settings (on page 403).

List of settings and detailed descriptions

Some of the configuration settings are commonly used and they have tasks already documented in the BigFix Support site for your assistance. If such a task exists, it is indicated along the **Task available ?** row in the tables which you can use. If no task exists, create the configuration setting manually.

Note: The component restart is required ONLY if explicitly mentioned in the **Component restart required ?** field. If the **Component restart required ?** field does not exist in the setting details, then the component restart is NOT required.

Making a configuration setting

You can make a configuration setting in two ways: through the BES Console or manually on the endpoint.

Through the BES Console

1. Open the BES console and navigate to the **Computer** section under the **All Content** domain.
2. Select the computer(s) to which you want to apply the configuration settings.

   Note: To change a configuration setting for the server or relays using this mechanism, you must select the computer(s) that have the server or relays installed.

3. Right-click the computer(s) and choose **Edit Computer Settings**.
4. Create a custom setting using a **Name** and **Value** pair from the configuration table.
5. Click **OK** to send the configuration setting through an action named **Change Multiple Settings** or a similar one.

   The setting takes effect after the action is complete.
6. View the computer's updated setting. To view the updates, right-click the computer and select **Edit Computer Settings** or click the computer in the computer list and then select the **Summary** tab. Then, scroll down to the **Client Settings** section.

Manually on a Windows endpoint

The client configuration settings are maintained as keys in the Windows registry at the following location: `HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\BigFix\EnterpriseClient\Settings\Client`. To update the settings, do the following steps:

1. Create a key at the registry location and give it a name.
2. Create a string value (REG_SZ type) named value within the key.
3. Set the value data within the value.

For guidance, review the already existing client setting keys.

Note: On Windows, you do not need to stop the BESClient service to apply the client setting manually. However, some settings may not take effect until the BESClient service is restarted. On Unix/Linux, you must stop the BESClient service to manually apply the configuration setting in the `besclient.config` file.

Manually on a Mac endpoint

On MacOS, you can manually change your BigFix Client settings from the command line.

Stop the BigFix Client service.

```
sudo /Library/BESAgent/BESAgent.app/Contents/MacOS/BESAgentControlPanel.sh
 -stop
```

Run the -setSettings command and provide an appropriate JSON file.

```
sudo "/Library/BESAgent/BESAgent.app/Contents/MacOS/BESAgent" -setSettings
 "/path/to/file.json"
```

Start the BigFix Client service.

```
sudo /Library/BESAgent/BESAgent.app/Contents/MacOS/BESAgentControlPanel.sh
 -start
```

Here is a JSON sample file that shows how to set (or change) two client settings, named "setting1" and "setting2".

```
{
 "client_settings": [
  {
   "key":"setting1",
   "value":"1"
  },
  {
   "key":"setting2",
   "value":"setting value 2"
  }
 ]
}
```

Description, values, and references

The configuration settings are categorized by the following BigFix functions:

- Plugin Portal *(on page 408)*
- Inspector behavior *(on page 439)*
- Action execution *(on page 451)*
- Action management *(on page 461)*
- Archiving client files *(on page 472)*
  - Archive Manager *(on page 472)*
  - Post File *(on page 483)*
  - Upload Manager *(on page 489)*
- Gathering content *(on page 530)*
  - HTTPS *(on page 575)*
- Deployment Encoding *(on page 582)*
- Logging *(on page 591)*
  - Auditing *(on page 611)*
- CPU Usage *(on page 619)*
- Proxy agent *(on page 648)*
- Proxy configuration *(on page 650)*

    - Client *(on page 650)*
    - Server/Relay *(on page 663)*
- Registration *(on page 674)*
    - Relay selection *(on page 684)*
    - Relay management *(on page 703)*
    - Affiliation *(on page 707)*
    - Child Number Limit *(on page 710)*
- Client notification *(on page 714)*
    - Command polling *(on page 714)*
    - UDP notifications *(on page 720)*
    - Persistent TCP connections *(on page 733)*
- Communication *(on page 754)*
    - Authentication *(on page 757)*
- Download *(on page 766)*
- Reports management *(on page 799)*
- Query *(on page 807)*
- Application Usage Collection *(on page 813)*
- Bandwidth throttling *(on page 818)*
- Airgap Mode *(on page 825)*
- HTTP server *(on page 827)*
    - HTTPS *(on page 832)*
    - Logging *(on page 838)*
- Relay diagnostics *(on page 844)*
- User authentication *(on page 851)*
- Explorer *(on page 854)*
- Web Reports *(on page 857)*
- Deployment Telemetry *(on page 869)*
- UI History *(on page 880)*
- Relays in DMZ *(on page 895)*
    - Parent relay *(on page 895)*
    - Child relay *(on page 906)*
- Peer to peer mode *(on page 915)*

- Automatic Server Clean Up *(on page 968)*
- User base management *(on page 971)*
- External Repository Sites *(on page 989)*
- Miscellaneous *(on page 995)*

For information related to parallel FillDB configuration, see Configuring parallel FillDB (on page 1040).

Plugin Portal

Name/Description	References	Values	Component(s) affected	Version(s) applicable
_BESPluginPortal_Log_Verbose If set to 1 it enables verbose logging on the Plugin	The Plugin Portal *(on page 213)*	Default value: 0 (disabled) Setting type: Boolean Value range: 0 (disabled) or 1 (enabled) Component restart required ? No	Plugin Portal	10 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
Portal for troubleshooting purposes. This setting increases the information written to the existing BES-Plug-in-				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
`Portal.log` file and should not be left on during normal operation.				
_BESPluginPortal_HTTPServer_LogFilePath The full path to the Plug-	The Plugin Portal *(on page 213)*	Default value `<Plugin Portal Install-Dir>\BES-Plugin-Portal.log`	Plugin Portal	10 and later

Name/Description	References	Values		Component(s) affected	Version(s) applicable
in Portal log file.			(Windows) `/var/log/BES-Plugin-Portal.log` (Linux)		
		Setting type	String		
		Component restart required ?	Yes		
BESPlugInPortal-HTTPServer_LogFileSizeLimit This setting		Default value	52,428,800 (50MB)	Plugin Portal	10 and later
		Value range	0 - 4,294,967,295		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
specifies the maximum size in bytes that a log file can reach before rotation. A maximum of 10 rotated		Setting type: Numeric (bytes) Task available? No		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
copies of the log file are maintained in addition to the active log file (for example, the active log file, BES-Plug-in-				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
Portal.log file and rotated copies, BES Plugin Portal.log_0, BES Plugin Portal.log_1,- .- .., BES Plugin Portal.log_9)- . The				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
plug-in portal log file rotates automatically even without setting the limit, it uses its default val-				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
ue. The active log file is filled in with logged data until it reaches the specified maximum size. Its content then				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
overwrites the content of the oldest log file. The active log file is then emptied ready to continue to collect new logged				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
data. For example, if you want your log to rotate across 10 files, each one being at most 100 KB, you must set "Log-File-				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
Size-Limit" to 100000.				
BESPluginPortal-HTTPServer_LogFile-Rotation-Limit This setting is used to define the maximum amount of files that the Plu-		Default value: 10 Value range: 0 - 4,294,967,295 Setting type: Numeric Task available?: No Component restart required?: Yes	Plugin Portal	10 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
g-in Portal should write.				
_BESPluginPortal_ManagementRights_MinRefreshInterval When enabled, it controls the minimum frequency at which	The Plugin Portal *(on page 213)*	Default value: 0 (disabled) Setting type: Numeric (Hours) Value range: 1 - 8,760 (1 hour - 365 days) Task available?: No Component restart: Yes	Plugin Portal	10 Patch 1 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
the Plug-in Portal evaluates the management right actions. By default, it is disabled. If enabled, the devices man-		required ?		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
aged by the Plug-in Portal will report faster on the BigFix console but changes against the operator management rights				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
will not be evaluated immediately.|				
_BESPluginPortal_Performance_ExcludeCustomSites-Subscription When enabled (1, default value), the	Custom Site management *(on page 219)*	Default value: 1 (enabled) Setting type: Boolean Value range: 0 (disabled) - 1 (enabled) Task available? No	Plugin Portal	10 Patch 4 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
Plug-in Portal adds an additional filter to exclude the subscription to Custom Sites for the devices it dis-		Component restart required ? Yes		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
covers. For more details, see Custom Site management *(on page 219)*.				
_BESPluginPortal_Register-Devices_-MaxNumberPerIterationPerPlugin Determines the max-	The Plugin Portal *(on page 213)*	Default value: 0 (disabled) Setting type: Numeric (Number of devices)	Plugin Portal	10 Patch 8 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
imum number of devices that the Plug-in Portal can register for each plug-in during a single iter-		Value range: >= 0 Component restart required ?: No		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
a-tion loop. If the num-ber of de-vices to reg-is-ter ex-ceeds the thresh-old, their reg-is-tra-tion is queued for the sub-se-				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
quent it- er- a- tions. For ex- am- ple, if the lim- it is 10, this means each plu- g- ins can reg- is- ter a max- i- mum of				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
10 devices at once, and more later. Before the next iteration starts, the Plugin Portal can execute other				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
operations on the registered devices, like: the evaluation of Fixlets, the execution of actions, etc. This set-				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
ting can help smooth operations in environments with slow network or low storage access performances. **Note:** This set-				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
 ting is valid up to BigFix Version 11.0.6.				
_BESPluginPortal_Computers-Groups-Gather_Pre-Process-Fixlets When the Plugin portal receives an		Default value: 0 (disabled) Setting type: Boolean Value range: 0 (disabled) - 1 (enabled) Task available? No	Plugin Portal	11 Patch 3 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
action "Dynamically targeted by property", it requests a refresh to all the devices submitted to the		Component restart required ? No		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
site containing the action in order to get the more recent value of the device properties. Belonging to a				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
computer group is a particular device property; when this setting is enabled and the "targeting relevance"				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
of the associated Fixlet refers to the properties that indicate membership of a group of computers, the Plu-				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
g-in Portal executes the targeting relevance to get the list of devices belonging to the group, and requests				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
the refresh only to the resulted devices. When disabled (0, default value), the Plug-in Portal requests the re-				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
fresh to all the devices submitted to the site.				

Top (on page 406)

Inspector behavior

Name/Description	References	Values	Component(s) affected	Version(s) applicable
_BESClient_Inspector_DebianPackageForceCacheRefresh Set to 1 to force the refresh of the Debian-		Default value: 0 (disabled) Setting type: Boolean Value range: 0 (disabled) - 1 (enabled)	Client	11 Patch 3 and later

Name/Description	References	Values		Component(s) affected	Version(s) applicable
Package inspector cache. In this case, when the Debian-Package inspector is invoked, if the last time the Debian-Package cache was refreshed exceeded 60 minutes, it forces a refresh to the		Platform	Linux		
		Task available ?	No		
		Component restart required ?	Yes		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
Debian-Package cache to exclude any mismatch between the cached information and the actual operating system Debian-Package status. A restart is needed for the setting to				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
be applied.				
_BESClient_Inspector_DebianPackageDisableCache Set to 1 to disable the DebianPackage inspector cache. In this case, seriously degraded client performance should be expected. A restart is need-		Default value: 0 Setting type: Boolean Value range: 0 (disabled) - 1 (enabled) Platform: Linux Task available ?: No Component restart required ?: Yes	Client	11 Patch 3 and later

Name/Description	References	Values		Component(s) affected	Version(s) applicable
ed for the setting to be applied.					
_BESClient_Inspector_ActiveDirectory_Refresh_Seconds Controls the frequency at which the BigFix client polls the Active Directory information.	Active Directory property updating	Default value	43,200 (12 hours)	Client	All
		Setting type	Numeric (Seconds)		
		Value range	1,200 - 2,147,483,648 (20 minutes - ~25K days)		
		Task available ?	No		
		Component restart	No		

Name/Description	References	Values		Component(s) affected	Version(s) applicable
		re-quired ?			
_BESClient_Inspector_AdminPrivilegeFromToken Enable this setting if you have user accounts belonging to a local domain group and the local domain group belongs to the administrators group. If this set-		Default value Setting type Value range Task available ? Component restart required ?	0 (disabled) Boolean 0 (disabled - 1 (enabled) No No	Client	9.5.9 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
ting is not enabled, the administrative privileges of these user accounts are not correctly retrieved. This setting is not enabled by default, set this setting to "1" to enable. Among other business				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
needs, you should enable this setting also when the Active Directory security groups information is not correctly retrieved by the BigFix client.				
_BESClient_Inspector_RPMForceCacheRefresh Set to 1 to force the re-		Default value: 0 (disabled) Setting type: Boolean Value range: 0 (disabled)	Client	All

Name/Description	References	Values		Component(s) affected	Version(s) applicable
fresh of the rpm inspector cache. In this case, when the rpm inspector is invoked, if the last time the rpm cache was refreshed exceeded 60 minutes, it forces a refresh to the rpm cache to exclude			- 1 (enabled)		
		Platform	Linux/AIX		
		Component restart required ?	No		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
any mismatch between cached information and actual operating system rpm status.				
_BESClient_Inspector_RPMDisableCache Set to 1 to disable the rpm inspector cache. In this case, if the child process is en-	Managing Downloads (on page 1099) Problems experienced when attempting to start the Unix/Linux BigFix Client after installation	Default value: 0 (disabled) Setting type: Boolean Value range: 0 (disabled) - 1 (enabled) Platform: Linux/AIX Component restart: No	Client	All

Name/Description	References	Values	Component(s) affected	Version(s) applicable
abled, seriously degraded client performance should be expected. **Note:** The cache operation is independent from the child process. In par-	Stopping the BES-ClientUI from running on every local user session on a Citrix Server	required ?		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
 tic- u- lar, the cache may be en- abled even if the child process is dis- abled.				
_BESClient_Inspector_DisableWMI This setting will disable the use of any inspectors that make calls to the Windows WMI. WMI has been found to cause problems on some computers, especially older versions of Win-	High Disk I/O and/or CPU on the client related to WMI queries	Default value: 0 (disabled) Setting type: Boolean Value range: 1 (true) - WMI inspectors disabled	Client	9.2 and later

Name/Description	References	Values		Component(s) affected	Version(s) applicable
dows (95 and 98). Problems include high CPU usage and 'blue screen' crashing of Windows. WMI is used by some optional retrieved properties.			0 (false) - WMI inspectors enabled		
		Task available ?	No		

Top (on page 406)

Action execution

Name/Description	References	Values		Component affected	Version(s) applicable
_BESClient_ActionManager_PrefetchPlugInTimeoutSeconds This configuration setting enables to customize the amount of	Prefetch plug-in took too long Linux Prefetch Plug-in and black list The client logs contains a	Default value	60	Client	All
		Setting type	Numeric (seconds)		
		Value range	60 - 6,000		

Name/Description	References	Values	Component affected	Version(s) applicable
time that the client waits, after executing a prefetch plug-in, for the plug-in to generate an answer. The time is expressed in seconds.	prefetch plug-in error that prevents the Fixlet from completing successfully. What is causing the error? What should I do?	Platform: All Component restart required: No ?:		
_BESClient_ActionManager_PresentOfferAfterAllConstraints To control the Client behavior when managing actions configured as an offer. If set to 1 (True),		Default value: 0 (false) Setting type: Boolean Value range: 0 or 1 Platform: Windows and Mac Component: No	Client	9.5.7 and later

Name/Description	References	Values	Component affected	Version(s) applicable
the offer is presented to the user on the Client UI at action start time. If set to 0 (False), the offer is presented to the user on the Client UI as soon as the action is received by the Agent (BESClient).		nent restart required?		
_BESClient_ActionManager_UIMissingHoldMode To control client behavior for actions	User interface process unable to launch	Default value: "failed" Setting type: String	Client	All

Name/Description	References	Values		Component affected	Version(s) applicable
with "don't care" user constraint and UI to display. This setting is very useful when you have remote clients connecting to the computer and the remote sessions with an error of User interface process unable to launch (XX.YYYY) for user " Example User interface		Values	"none" "failed" "all" • "none" - automatically run the action if the only UI sessions are disabled or failed. • "failed" -		

Name/Description	References	Values	Component affected	Version(s) applicable
process unable to launch (22.1008) for user " This usually indicates an error condition of the connection to the remote connection was present, but was terminated. We are not able to detect the user name being used because the abandoned user connection		hold the action in a pending message state if there are UI sessions that are failed. • "all" - hold the action in a pending		

Name/Description	References	Values		Component affected	Version(s) applicable
was left active but not terminated correctly. Other examples and information related to this are conditions of abandoned session tokens.			message state if there are UI sessions that are failed or disabled.		
		Task available ?	No		
		Platform	All		
		Component restart required	Yes ?		

Name/Description	References	Values	Component affected	Version(s) applicable
_BESClient_ActionManager_-SSAv2Mode This setting was implemented to allow the BESClient to display the Client UI dashboards in the Self Service Application (SSA). If SSA is not installed, this setting must be deleted, if present. For more information, refer to the		Default value: "SSAV2UIAll" Setting type: String Values: "SSAV2UIAll" Task available ?: No Platform: Windows/Mac Component restart required: Yes ?	Client	9.5.3 and later

Name/Description	References	Values	Component affected	Version(s) applicable
Software Distribution Self Service Application documentation.				
_BESClient_Action-Manager_Pending-RestartExclusions String(s) residing in the registry key X which are to be ignored by the BigFix Client when determining if a restart is needed. The strings must be separated by semi-	Determining if a restart is needed	Default value: Blank Value range: NA Task available ?: No Platform: Windows Component restart required ?: No	Client	All

Name/Description	References	Values	Component affected	Version(s) applicable
colons and the last string must have a terminating semicolon (for example "exclude1;exclude2;exclude3;"). With the setting _BESClient_Action-Manager_PendingRestartExclusions=:; all entries in the Microsoft registry key HKLM\System\CurrentControlSet\Con-				

Name/Description	References	Values	Component affected	Version(s) applicable
trol\Session ManagerPendingFileRename-Operations are ignored because every entry is a path containing always a colon. **Note:** The strings are case sensitive.				
_BESClient_ActionManager_Locale-Enable This configuration		Default value: 1 (enabled)	Client	All

Name/Description	References	Values		Component affected	Version(s) applicable
setting will enable BigFix Clients to display non-English messages if the translation files are in place (IEM 6.0+).		Setting type	Boolean		
		Value range	1 (enable), 0 (disable)		
		Task available ?	Yes		
		Platform	Linux, Unix		
		Component restart required	Yes ?		

Top (on page 406)

Action management

The settings described in this section are made on the BigFix Client to configure how actions are run locally.

Settings affecting data download on the Client

Whenever an action requiring the download of one or more than one files is taken, the target agent prevents the action from running on the client if the total size of the downloads

associated to the action exceeds the value set in _BESClient_Download_SizeLimitMB, and the Client is not connected to a preferred relay.

The Client installed on the Server and the Client/Relay do not consider the setting _BESClient_Download_SizeLimitMB, allowing always the download (on the Relay/Client the condition is true when the Relay is active).

The connected relay is a preferred relay if:

- The relay selection associated to the Client is the Manual Relay Selection, and the connected relay is the primary or the secondary relay.
- The relay selection associated to the Client is the Automatic Selection with Affiliation, and the connected relay is member of an affiliation group belonging to the affiliation list. The number of affiliation groups belonging to the affiliation list can be configured using the setting _BESClient_PreferredRelay_MaxAffiliationsToCheck. The special affiliation group "*" is never used for setting a preferred relay.
- The connected relay is either in the same sub-network of the client, or the number of network hops from the Client to the relay is lower than the value configured in the setting _BESClient_PreferredRelay_MaximumHopCount. In this case there is no dependency from the relay association method selected on the Client.

Name/Description	References	Values	Component(s) affected	Version(s) applicable
_BESClient_PreferredRelay_MaxAffiliationsToCheck Use this setting		Default value: 1 Setting type: Numeric	Client	9.5.9 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
to specify the number of affiliation groups, belonging to the affiliation list, to consider when		Value range: 1 - 255 Task available ?: No Component restart required ?: No		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
evaluating the preferred relay. By default any member of the first group in the affiliation list is a pre-				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
ferred relay.				
_BESClient_PreferredRelay_-MaximumHopCount Use this setting to specify the maximum number of network hops from the client		Default value: 1 (adjacent subnetwork) Setting type: Numeric (number of hops) Value range: 0 - 255 Task available?: No Component restart: No	Client	9.5.9 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
to the preferred relay. The number of network hops from the client to the preferred relay must be lower than the value required ?				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
specified in this setting. By default, if the connected relay is in the same subnetwork of the Client, it is a				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
preferred relay. **Note:** By setting the value to 0, this setting considers the Relays in the same subnetwork as				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
not preferred.				

Settings affecting wait and waithidden commands on the client

You can use the wait command, as part of an action, to wait for the completion of a specific process or program before continuing with the next actionscript command. You can also use the waithidden command to run the process or program in a hidden window. When defining the wait command, you can optionally specify an override section to change some of the default behaviors, represented by keywords, that are applied at runtime on the target Clients. If you do so the different behavior applies only to the program or process triggered by that wait or waithidden command on all the clients where the action runs. The settings listed in the following table allow you to define the default behavior for timeout and disposition on a specific client for all the programs or processes triggered by any wait or waithidden commands, unless it is specified differently in an override section of that specific wait or waithidden command definition.

Name/Description	References	Values	Component(s) affected	Version(s) applicable
_BESClient_Action-Manager_Override-TimeoutSeconds Use this setting to define on the specific Client how many		Default value: 0 Setting type: Numeric (seconds) Value range: 0 - 4,294,967,295	Client	9.5.11 and later

Name/Description	References	Values		Component(s) affected	Version(s) applicable
seconds the action processing must wait for the completion of the wait or waithidden command's process before timing out. When the timeout elapses, the behavior specified in the disposition is applied to the wait or waithidden command's process, the overall action stops processing and exits in *Timeout Reached* status.		Task available ?	No		

Name/Description	References	Values		Component(s) affected	Version(s) applicable
_BESClient_Action-Manager_OverrideDisposition Use this setting to define on the specific Client what to do with the processes or programs triggered by any wait or wait-hidden command once the timeout elapses. The available values are: • abandon, to disassociate the wait or wait-hidden com-		Default value	abandon	Client	9.5.11 and later
		Setting type	String		
		Value range	abandon, terminate		
		Task available ?	No		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
mand's process from the remainder of the actions. • terminate, to kill the wait and wait-hidden command's process.				

Top (on page 406)

Archiving client files

This allows the BigFix Administrator to automatically log data from specific managed computers.

For details, see Archiving Client files on the BigFix Server (on page 395).

Archive Manager

For detailed description about Archive Manager, see Archive Manager (on page 396).

The BigFix Client which uses the Archive Manager must be located on a different machine than the BigFix Server, for the Archive Manager to function correctly.

Name/Description	References	Values		Component(s) affected	Version(s) applicable
_BESClient_Archive-Manager_Operating-Mode The OperatingMode dictates the style of archiving, allowing periodic or triggered archiving.	Archive Manager *(on page 396)*	Default value	0	Client	9.2 and later
		Setting type	Numeric		
		Value range	• 0 - Disables all archival operations • 1 - Automatic with a period = BESClient_-		

Name/Description	References	Values		Component(s) affected	Version(s) applicable
			Archive-Manager_-Interval-Seconds • 2 - Enables the archive now action command. To allow a custom action		

Name/Description	References	Values		Component(s) affected	Version(s) applicable
			to post client attributes to an archive file, make sure the OperatingMode is set to 2.		
		Task available ?	No		
_BESClient_ArchiveManager_FileSet-<tag>	Archive Manager *(on*	Default value	None	Client	9.2 and later

Name/Description	References	Values		Component(s) affected	Version(s) applicable
This setting (or a group of settings with optional tags) specifies the files to be archived. This technique lets you specify multiple named batches of files. Each setting starts with "_BESClient_ArchiveManager_FileSet-" and ends with a batch name (the <tag> part). The value of each setting is a path on the client file system. It can be a single file, in which	*page 396)*	Setting type	String		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
case that file is part of the archive; a single directory, in which case all files in the directory will be part of the archive; or a directory path ending with wild cards, in which case all files in the directory matching the wild cards will be part of the archive. For example, the setting _BESClient_ArchiveManager_FileSet-(log), representing all the log files in a tempo-				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
rary log folder, could have a value like c:\temp\log. Everything after the dash (-) is used as the default prefix of the files as they are unpacked on the root server. Therefore a file named x.log in the c:\temp\log folder would be unpacked as (Log)x.log.				
_BESClient_Archive-Manager_KeepFile-Set-<tag> This setting (or a group of settings with optional tags)	Archive Manager *(on page 396)*	Default value: 0 (Disabled) Setting type: Boolean	Client	11.0.2 and later

Name/Description	References	Values		Component(s) affected	Version(s) applicable
specifies the files to be kept on the BigFix Server. Each setting starts with "_BESClient_ArchiveManager_KeepFileSet-" and ends with a batch name (the <tag> part). The batch name must match with the batch name defined for the setting named _BESClient_ArchiveManager_FileSet-<tag>. This setting only works		Value range	• 0 - Disabled. • 1 - Enabled.		
		Task available ?	No		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
when used in conjunction with the setting named _BESClient_ArchiveManager_FileSet-<tag>.				
_BESClient_ArchiveManager_SendAll This setting allows you to send just the archives that have changed, avoiding redundant uploads. There are two possible values for this setting. The Default value of 0 is recommended for most applications.	Archive Manager *(on page 396)*	Default value: 0 Setting type: Boolean Value range: • 0 - Only send files that have changed since the last archive operation	Client	9.2 and later

Name/Description	References	Values		Component(s) affected	Version(s) applicable
			• 1 - Send all files, even if they have not changed.		
		Task available ?	No		
_BESClient_ArchiveManager_MaxArchiveSize This setting limits the size (in bytes) of the uploaded archive. Because a typical archive might be composed of several files, the archive	Archive Manager *(on page 396)*	Default value	1,000,000 (one million bytes). Starting in BigFix version 8.0, the file system is 64-bit. This means that the actual	Client	9.2 and later

<table>
<tr><th>Name/Description</th><th>References</th><th colspan="2">Values</th><th>Component(s) affected</th><th>Version(s) applicable</th></tr>
<tr><td>size corresponds to the sum of the file sizes. If the limit is exceeded, an archive that contains only the index file is created and uploaded by the Archive Manager. The index contains the following header line: <code>MaxArchiveSize: Exceeded Copy</code>.</td><td></td><td></td><td>maximum file size is 2^{64} – 1, sufficient for any reasonably sized file.</td><td></td><td></td></tr>
<tr><td></td><td></td><td>Setting type</td><td>Numeric (bytes)</td><td></td><td></td></tr>
<tr><td></td><td></td><td>Value range</td><td>0 - (2^{64}– 1)</td><td></td><td></td></tr>
<tr><td></td><td></td><td>Task available ?</td><td>No</td><td></td><td></td></tr>
<tr><td>_BESClient_Archive-Manager_IntervalSeconds
When the OperatingMode is set to 1, this setting</td><td>Archive Manager (on page 396)</td><td>Default value</td><td>86,400 seconds (24 hours)</td><td>Client</td><td>9.2 and later</td></tr>
<tr><td></td><td></td><td>Setting type</td><td>Numeric (seconds)</td><td></td><td></td></tr>
</table>

Name/Description	References	Values		Component(s) affected	Version(s) applicable
determines the interval at which the client triggers an archive.		Value range	0 or a range from 120 to 4,294,967,295		
		Task available ?	No		

Top (on page 406)

Post file

The PostFile program receives the chunks of files posted by the Upload Manager and appends them to its own copy of the file. The Upload Manager specifies the range of bytes being posted and the sha1 of the file, which is used as the filename. For details about the configuration settings related to this component, see Post File (on page 483) in BigFix Configuration Settings (on page 403).

For details, see PostFile (on page 399).

Name/Description	References	Values		Component(s) affected	Version(s) applicable
_BESRelay_PostFile_-PassThrough You can use this setting	Post-File (on page 399)	Default value	0 (disabled)	Relay	11 Patch 4 and later
		Setting type	Boolean		

Name/Description	References	Values		Component(s) affected	Version(s) applicable
on the BigFix Relay in order for the Relay to immediately forward received file chunks upstream (this setting should not be applied to the Root server). When this setting is disabled (**0**), the BigFix Relay continues to function as it always did, receiving files in chunks through the PostFile program, storing them locally until files are complete, and then us-	Upload Manager (on page 399)	Value range	0 (disabled) - 1 (enabled)		
		Task available ?	No		
		Component restart required ?	Yes		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
ing the Upload Manager to send them upstream through the communication chain. When this setting is enabled (**1**), the BigFix Relay operates in pass-through mode, forwarding the received file chunks directly upstream without a local storage. **Note:** When enabling this setting				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
on a BigFix Relay, any `_-BESRelay_-Post-File_-*` or `_-BESRelay_-Upload-Manager_-*` settings you may have defined on it will				

<table>
<tr><th>Name/Description</th><th>References</th><th>Values</th><th>Component(s) affected</th><th>Version(s) applicable</th></tr>
<tr><td> be ignored.</td><td></td><td></td><td></td><td></td></tr>
<tr><td>_BESRelay_PostFile_-ThrottleKBPS
The PostFile component of BigFix uses this setting for controlling throttle values for the incoming data.
The value of the setting can be adjusted for varying connection speeds or other network anomalies. When PostFile communicates with the Upload Manager, it passes along this value. If there</td><td>Bandwidth throttling (on page 1090)</td><td><table><tr><td>Default value</td><td>0 (disabled)</td></tr><tr><td>Setting type</td><td>Numeric (KBPS)</td></tr><tr><td>Value range</td><td>0 - 4,294,967,295</td></tr><tr><td>Task available ?</td><td>No</td></tr></table></td><td>Relay, Root server</td><td>9.2 and later</td></tr>
</table>

<table>
<tr><th>Name/Description</th><th>References</th><th>Values</th><th>Component(s) affected</th><th>Version(s) applicable</th></tr>
<tr><td>is a conflict between any two computers over these settings, it favors the smaller value.</td><td></td><td></td><td></td><td></td></tr>
<tr><td>_BESRelay_PostFile_-ChunkSize
The PostFile component of BigFix uses this setting for controlling the chunk size of incoming data.
The value of the setting can be adjusted for varying connection speeds or other network anomalies. When PostFile communi-</td><td>Bandwidth throttling (on page 1090)</td><td><table><tr><td>Default value</td><td>0 (disabled)</td></tr><tr><td>Setting type</td><td>Numeric (in bytes)</td></tr><tr><td>Value range</td><td>1,024 - (2^{64}- 1)
Any value between 1 and 1,023 is reset to 1,024.</td></tr><tr><td>Task available ?</td><td>No</td></tr><tr><td>Component</td><td>Yes</td></tr></table></td><td>Relay, Server</td><td>9.2 and later</td></tr>
</table>

Name/Description	References	Values	Component(s) affected	Version(s) applicable
cates with the Upload Manager, it passes along this value. If there is a conflict between any two computers over these settings, it favors the smaller value.		restart required ?		

Top (on page 406)

Upload Manager

The Upload Manager coordinates the sending of files in chunks to the Post File program. You can throttle the upload dataflow to conserve bandwidth. The file system uses 64-bits, sufficient for file sizes of up to 2^{64} – 1 bytes in length.

Note: BigFix Inventory and BigFix License Metric Tool upload a lot of data from the BigFix Clients. If you plan to use any of these two applications you are suggested to specify a value for the settings `_BESRelay_UploadManager_BufferDirectoryMaxSize` and `_BESRelay_UploadManager_BufferDirectoryMaxCount` described in the following table to limit the disk space usage.

Name/Description	References	Values		Component(s) affected	Version(s) applicable
_BESRelay_UploadManager_-AllowConsoleUploads Controls the ability to upload files as "console uploads" to the relay. Set it to 1 to re-enable. **Note:** Enabling this	Upload Manager *(on page 399)*	Default value	0 (disabled)	Relay, Root server	9.5 and later
		Setting type	Boolean		
		Value range	0 - 1		
		Task available ?	No		
		Component restart required ?	No		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
type of upload is not secure because the relay does not require any sort of authentication.				

Name/Description	References	Values		Component(s) affected	Version(s) applicable
_BESClient_UploadManager_-BufferDirectory The input buffer directory of the Upload Manager. This directory is on the client computer, in the BigFix Client folder.	Upload Manager *(on page 399)*	Default value	None	Client	9.2 and later
		Setting type	String		
		Task available ?	No		
		Component restart required ?	No		
_BESClient_UploadManager_-ChunkSize	Upload Man-	Default value	131,072 (128KB)	Client	9.2 and later

<table>
<tr><th>Name/Description</th><th>Ref-er-ences</th><th colspan="2">Values</th><th>Com-ponen-t(s) af-fected</th><th>Ver-sion(s) ap-plic-able</th></tr>
<tr><td>Up-loads are done one chunk at a time. In a con-flict be-tween this com-put-er and the up-stream com-put-er, the size of the chunk is set to the</td><td>ager (on page 399)</td><td colspan="2"><table>
<tr><td>Setting type</td><td>Numeric (bytes)</td></tr>
<tr><td>Value range</td><td>1,024 - 4,294,967,295
Note: If you set a val-ue less than 1,024, it is au-tomati-cally re-set to 1,024.</td></tr>
<tr><td>Task available ?</td><td>No</td></tr>
<tr><td>Component restart re-quired?</td><td>Yes</td></tr>
</table></td><td></td><td></td></tr>
</table>

<table>
<tr><th>Name/Description</th><th>References</th><th>Values</th><th>Component(s) affected</th><th>Version(s) applicable</th></tr>
<tr><td>smaller of the two. The local chunk size setting is specified in bytes.</td><td></td><td></td><td></td><td></td></tr>
<tr><td>_BESClient_UploadManager_ThrottleKBPS
After each chunk is uploaded, the Upload Manager calcu-</td><td>Upload Manager (on page 399)</td><td><table><tr><td>Default value</td><td>0 (Disabled)</td></tr><tr><td>Setting type</td><td>Numeric (KBPS)</td></tr><tr><td>Value range</td><td>0 - 4,294,967,295</td></tr><tr><td>Task available ?</td><td>No</td></tr><tr><td>Component restart required ?</td><td>No</td></tr></table></td><td>Client</td><td>9.2 and later</td></tr>
</table>

Name/Description	References	Values	Component(s) affected	Version(s) applicable
lates the amount of time to sleep to maintain the throttle speed in kilobytes per second (Throttle-KBPS). This allows you to compensate for net-				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
work bottlenecks. For example, a BigFix client that is connected over a slow VPN to the relay might have a low upload throttle rate to minimize the bandwidth				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
on that network segment. In a conflict between this computer and the upstream relay (or server), the throttle KBPS is set to the smaller of				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
the two.				
_BESRelay_UploadManager_-BufferDirectory Like the BigFix Client, the BigFix Relay also has an Upload Manager, and it also has a buffer directory, whose path is spec-	Upload Manager *(on page 399)*	Default value: None Setting type: String Task available ?: No Component restart required ?: No	Relay, Root server	9.2 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
ified by this setting. The Upload Manager uploads the files in the `sha1` subdirectory of the specified directory. It sorts the files by modification time				

Name/Description	References	Values		Component(s) affected	Version(s) applicable
and then, just like the BigFix Client, it uploads them in chunks to smooth out the bandwidth requirements.					
_BESRelay_UploadManager_BufferDirectoryMaxSize Denotes the	Upload Manager *(on page 399)*	Default value	• 1 Gbyte, on relay and on the server until 9.5.4 • 2^{64} – 1, on the	Relay, Root server	9.2 and later

<table>
<tr><th>Name/Description</th><th>References</th><th colspan="2">Values</th><th>Component(s) affected</th><th>Version(s) applicable</th></tr>
<tr><td>maximum amount of space on disk that can be used to store the data uploaded from the BigFix Clients using the Upload Manager. You can specify this</td><td></td><td colspan="2"><table><tr><td></td><td>server starting from 9.5.5</td></tr><tr><td>Setting type</td><td>Numeric (bytes)</td></tr><tr><td>Value range</td><td>0 - 2^{64} – 1</td></tr><tr><td>Task available ?</td><td>Yes</td></tr><tr><td>Component restart required ?</td><td>No</td></tr></table></td><td></td><td></td></tr>
</table>

Name/Description	References	Values	Component(s) affected	Version(s) applicable
setting on the BigFix Server or on the BigFix Relays. Depending on the role of the system in the BigFix topology, the behavior of this setting differs as follows:				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
BigFix Relay You can set the maximum file size to be as large as 2^{64} – 1 bytes. The default value is 1 GB. A check against this setting is run every time a new				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
file is received. **BigFix Server** By default, the setting is not specified on the system, meaning that the maximum size of the Buffer Directory is				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
unlimited. Specify this setting to define a threshold size for the Buffer Directory. You must remove the setting to restore the default value. The BigFix				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
Server checks every 15 minutes if the size of the Buffer Directory exceeds the value set in _BESRelay_UploadManager_BufferDirectoryMaxSize.				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
If this check is true, the BigFix Server does not accept any additional uploaded files. Change the value of the setting either manually or using the appropriate Fixlet,				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
or reduce the content of the Buffer Directory to resume uploading files. **Note:** Starting from version 9.5 Patch 5, the _-				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
BESRelay_Upload_Manager_BufferDirectory_MaxSize checking is listed among the prerequisite checks				

Name/Description	References	Values		Component(s) affected	Version(s) applicable
for running the upgrade. For more information, see Upgrading.					
_BESRelay_UploadManager_BufferDirectoryMaxCount Denotes the maxi-	Upload Manager *(on page 399)*	Default value	• 10,000 on both server and relay, up to version 9.5.4. • 2^{64}- 1 on the server	Relay, Root server	9.2 and later

<table>
<tr><th>Name/Description</th><th>References</th><th colspan="2">Values</th><th>Component(s) affected</th><th>Version(s) applicable</th></tr>
<tr><td>mum number of files that the Upload Manager Buffer Directory is allowed to store. You can specify this setting on the BigFix Server or on the BigFix</td><td></td><td colspan="2"><table><tr><td></td><td>(later than version 9.5.4)</td></tr><tr><td>Setting type</td><td>Numeric</td></tr><tr><td>Value range</td><td>0 - 2^{64} – 1</td></tr><tr><td>Task available ?</td><td>Yes</td></tr><tr><td>Component restart required ?</td><td>No</td></tr></table></td><td></td><td></td></tr>
</table>

Name/Description	References	Values	Component(s) affected	Version(s) applicable
Relays. Depending on the role of the system in the BigFix topology, the behavior of this setting differs as follows: **BigFix Relay** A check against this setting				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
is run every time a new file is received. **BigFix Server** By default the setting is not specified on the system, meaning that the maximum num-				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
ber of files in the Buffer Directory is unlimited. Specify this setting to define a threshold to the number of files stored in the Buffer Directory. You must re-				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
move the setting to restore the default value. The BigFix Server checks every 15 minutes if the number of uploaded files stored in the Buffer Directory				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
exceeds the value set in _BESRelay_UploadManager_BufferDirectoryMaxCount. If this check is true, the BigFix Server does not accept any additional up-				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
loaded files. Change the value of the setting either manually or using the appropriate Fixlet, or reduce the content of the Buffer Directory to resume up-				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
loading files. **Note:** Starting from version 9.5 Patch 5, the _-BESRelay_-Upload-Manager_-BufferDirecto-				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
ry-MaxCount checking is listed among the prerequisite checks for running the upgrade. For more information,				

Name/Description	References	Values		Component(s) affected	Version(s) applicable
see Upgrading.					
_BESRelay_UploadManager_-CompressedFileMaxSize This setting denotes the amount of space of the largest compressed file the Upload Manager is allowed	Upload Manager *(on page 399)*	Default value	20,971,520 (20 MB)	Relay, Root server	9.2 and later
		Setting type	Numeric (bytes)		
		Value range	0 - 2^{64}- 1		
		Task available ?	No		
		Component restart required ?	No		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
to handle. You can set the maximum file size to be as large as 2^{64} – 1 bytes. It applies only to the server and it is evaluated during the decompression of				

<table>
<tr><th>Name/Description</th><th>References</th><th colspan="2">Values</th><th>Component(s) affected</th><th>Version(s) applicable</th></tr>
<tr><td>the uploaded archive.</td><td></td><td colspan="2"></td><td></td><td></td></tr>
<tr><td>_BESRelay_UploadManager_-ChunkSize
Uploads are done one chunk at a time. In a conflict between this computer and the upstream com-</td><td>Upload Manager (on page 399)</td><td>Default value</td><td>131,072 (128 KB)</td><td>Relay, Root server</td><td>9.2 and later</td></tr>
<tr><td></td><td></td><td>Setting type</td><td>Numeric (bytes)</td><td></td><td></td></tr>
<tr><td></td><td></td><td>Value range</td><td>1,024 - 4,294,967,295
Note: If you set a value less than 1,024, it is automatically reset to 1,024.</td><td></td><td></td></tr>
<tr><td></td><td></td><td>Task available ?</td><td>No</td><td></td><td></td></tr>
<tr><td></td><td></td><td>Component restart required ?</td><td>Yes</td><td></td><td></td></tr>
</table>

<table>
<tr><th>Name/Description</th><th>References</th><th>Values</th><th>Component(s) affected</th><th>Version(s) applicable</th></tr>
<tr><td>puter, the size of the chunk is set to the smaller of the two.</td><td></td><td></td><td></td><td></td></tr>
<tr><td>_BESRelay_UploadManager_ThrottleKBPS
After each chunk is uploaded, the Upload Manager calculates the</td><td>Upload Manager (on page 399)</td><td><table><tr><td>Default value</td><td>0 (Disabled)</td></tr><tr><td>Setting type</td><td>Numeric (KBPS)</td></tr><tr><td>Value range</td><td>1 - 65,536</td></tr><tr><td>Task available ?</td><td>No</td></tr><tr><td>Component restart required ?</td><td>No</td></tr></table></td><td>Relay, Root server</td><td>9.2 and later</td></tr>
</table>

Name/Description	References	Values	Component(s) affected	Version(s) applicable
amount of time to sleep to maintain the throttle speed in kilobytes per second (Throttle-KBPS). This allows you to compensate for network bottle-				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
necks. For example, a BigFix relay that is connected over a slow VPN to the server might have a low upload throttle rate to minimize the bandwidth on that net-				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
work segment. In a conflict between this computer and the upstream server (or relay), the throttle KBPS is set to the smaller of the two.				

<table>
<tr><th>Name/Description</th><th>References</th><th>Values</th><th>Component(s) affected</th><th>Version(s) applicable</th></tr>
<tr><td>The default value is 0, which disables throttling.</td><td></td><td></td><td></td><td></td></tr>
<tr><td>_BESRelay_UploadManager_CleanupHours
This setting controls when a partial upload will be removed from the BufferDir/Temp</td><td>Upload Manager (on page 399)</td><td><table><tr><td>Default value</td><td>72</td></tr><tr><td>Setting type</td><td>Numeric (hours)</td></tr><tr><td>Value range</td><td>0 - 4,294,967,295</td></tr></table></td><td>Relay, Root server</td><td>9.2 and later</td></tr>
</table>

Name/Description	References	Values	Component(s) affected	Version(s) applicable
directory. These files can remain in this folder due to a network outage, a downed server or other communication issues. To avoid overloading the sys-				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
tem, these old files are deleted or cleaned up. Completed downloads (which have been moved to BufferDir/sha1) are not affected by this setting.				

Top (on page 406)

Gathering content

Name/Description	References	Values	Component(s) affected	Version(s) applicable
_BESRelay_Site-Clean_Enable This setting controls the enablement of the Relay-Site-Clean task. The task runs every _-BESRe-		Default value: 1 Setting type: Boolean Value range: 0-1 Task available? No	Relay	9.5.7 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
lay_-Site-Clean_-Interval-Hours hours on the Relay, removing sites that have not been requested for gathering for more than				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
-BESRelay-Site-Clean_-Site-Not-Gathering-Days days. Setting this value to 0 disables the task entirely. The default val-				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
ue is 1.				
_BESRelay_SiteClean_IntervalHours This setting controls the frequency, specified in hours, on which the RelaySiteClean		Default value: 24 Setting type: Numeric (hours) Value range: 0 - 4,294,967,295 Task available? No	Relay	9.5.7 and later

<table>
<tr><th>Name/Description</th><th>References</th><th>Values</th><th>Component(s) affected</th><th>Version(s) applicable</th></tr>
<tr><td>task runs. The default value is every 24 hours.</td><td></td><td></td><td></td><td></td></tr>
<tr><td>_BESRelay_SiteClean_-SiteNotGathering-Days
This setting controls the number of days a site</td><td></td><td><table><tr><td>Default value</td><td>7</td></tr><tr><td>Setting type</td><td>Numeric (days)</td></tr><tr><td>Value range</td><td>0 - 4,294,967,295</td></tr><tr><td>Task available?</td><td>No</td></tr></table></td><td>Relay</td><td>9.5.7 and later</td></tr>
</table>

Name/Description	References	Values	Component(s) affected	Version(s) applicable
is not requested for gathering before the Relay-Site-Clean task deletes it. The default value is 7 days.				
_BESRelay_GathererMirror_-		Default 360	Relay	9.2 and later

Name/Description	References	Values		Component(s) affected	Version(s) applicable
UpstreamCheckPeriodMinutes This setting controls the minimum amount of time a relay (or a root server in DSA deployments) will wait		value			
		Setting type	Numeric (number of minutes)		
		Value range	0 - 4,294,967,295		
		Task available?	No		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
between checking for new versions of sites. In a network with full connectivity, this polling behavior will be unimpor-				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
tant, because relays will always receive notifications when new sites become available. But when notifications get				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
missed, this polling behavior allows a relay to "catch up". The downside to polling too frequently is that it can drive un-				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
necessary load into the parent. The polling will only be done in response to a client request, so if none of a re-				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
lay's children are asking for a site, it won't go upstream to check for the site, even if its polling period has expired. Note				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
that a "request" in this case means any query about the status of the site -- so a command polling request from a Client can trig-				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
ger this upstream check even though it's not directly requesting any sites. This setting does not directly control a "relay				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
gather interval", as there is no such thing. The Relay only initiates gather requests in response to notifi-				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
cations received from its parent or gather requests received from one of its children. However, if a Relay has many				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
children, gather requests will come in frequently, and the amount of time between upstream checks will end up being very close to				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
the minimum amount of time specified by this setting.				
_BESGather_Download_CheckParentFlag When a file is requested from the BigFix	Setting up a proxy connection **Relay downloads patch from Internet directly instead of gathering from BigFix Server** How to have Relays download content directly from the Internet	Default value: BigFix Server: 0 (False) BigFix Relay: 1 (True) Setting type: Boolean	Server, Relay	9.2 and later

Name/Description	References	Values		Component(s) affected	Version(s) applicable
Fix Relay, it can either download the file from the BigFix Server or download the file directly from the Internet.		Value range	1 (True) - Check the BigFix Server first to download the file 0 (False) - Do not check the BigFix Server when to download		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
If this setting is set to 1 (True), then the BigFix Relay will first attempt to download the file from the BigFix Server.		the file Task available No ?		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
Note: This setting should stay at 0 (False) for the BigFix Server.				
_BESGather_Download_CheckInternetFlag When a file is requested from	Setting up a proxy connection Relay downloads patch from Internet directly instead of gathering from BigFix Server How to have Relays download content di-	Default value: BigFix Server: 1 (True) BigFix Relay: 0 (False)	Server, Relay	9.2 and later

Name/Description	References	Values		Component(s) affected	Version(s) applicable
the BigFix Relay, it can download the file directly from the Internet instead of from the BigFix Server. If this	rectly from the Internet	Setting type	Boolean		
		Value range	1 (True) - Download the file directly from the Internet 0 (False) - Do not download the file directly from the		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
setting is set to 1 (True), then the BigFix Relay will first attempt to connect to the BigFix Server (if the _BES-		Internet Task available No?		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
Gatherer_-Download_-Check-Parent-Flag is set to 1) then directly connect to the Internet if the download from the Big-				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
Fix Server fails. **Note:** This setting should stay at 1 (True) for the BigFix Server.				
_BESGather_Download_CacheLimitMB BigFix Gather		Default value: 1,024 Setting type: Numeric (MB)	Server, Relay	9.2 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
will cache downloaded files on the BigFix Server or BigFix Relay computer so that the file will not need to be downloaded		Value range: 1 - 4,294,967,296 Task available?: Yes		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
again if the action is reissued. This setting will specify how many MB to cache before overwriting the old files. When the				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
cache is full, the least recently used files will be replaced. This setting requires a restart of the BES Relay Service.				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
_BESGather_Download_-RetryMinutes When BigFix Gather fails to download a file from the Internet or its parent during an action,		Default value: 10 Setting type: Numeric (minutes) Value range: 0 - 4,294,967,295 Task available? No	Server, Relay	9.2 and later

<table>
<tr><th>Name/Description</th><th>References</th><th>Values</th><th>Component(s) affected</th><th>Version(s) applicable</th></tr>
<tr><td>it will wait for the specified amount of time then try again.</td><td></td><td></td><td></td><td></td></tr>
<tr><td>_BESGather_Download_RetryLimit
When BigFix Gather fails to download a file</td><td></td><td><table><tr><td>Default value</td><td>6</td></tr><tr><td>Setting type</td><td>Numeric (retry limit)</td></tr><tr><td>Value range</td><td>0 - 4,294,967,295</td></tr></table></td><td>Server, Relay</td><td>9.2 and later</td></tr>
</table>

Name/Description	References	Values	Component(s) affected	Version(s) applicable
from the Internet or its parent during an action, it will retry the specified number of times (each time it will dou-		Task available? No		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
ble the amount of time it waited to retry the last time).				
_BESGather_Download_Channel-Threshold BigFix Gather can simultaneously		Default value: 500,000 Setting type: Numeric (bytes) Value range: 1 - 100,000,000 (Set to 0 to disable the	Server, Relay	9.2 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
download two files at a time by using one "main channel" and one "thin channel". The main channel is used for all downloads, but		thin channel) Task available? No		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
if the main channel is currently downloading a large file, the thin channel can be used to download smaller files if the				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
download size is less than the specified threshold. If this setting is set high, BigFix Gather will use the thin channel to				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
download larger files, which could slow down actions because two large files may be downloading at the same time (each using half the				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
bandwidth) instead of one file after the other. If this setting is set low, the thin channel will be used for only very				

Name/Description	References	Values		Component(s) affected	Version(s) applicable
small file downloads.					
_BESGather_Download_InactivityTimeout When the BigFix Gather service is downloading a file from the Internet or		Default value	300	Server, Relay	9.2 and later
		Setting type	Numeric (seconds)		
		Value range	0 - 4,294,967,295 (Set to 0 to disable timeout mechanism)		
		Task available?	No		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
its parent, it sometimes will experience some inactivity because of network problems, disconnections, etc. When there				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
is inactivity, BigFix Gather will wait for the specified number of seconds to receive more data before				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
aborting the download. **Note:** This setting is deprecated starting from BigFix Version 9.0.				
_BESGather_Download_-Timeout-Seconds		Default value: 30	Server, Relay	9.5.7 and later

Name/Description	References	Values		Component(s) affected	Version(s) applicable
With this value, you specify the maximum amount of time since the last activity on the connection. The BigFix Gath-		Setting type	Numeric (seconds)		
		Value range	0 - 4,294,967,295		
		Task available	No		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
er service waits for the specified number of seconds before aborting the download operation. After the download				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
connection is established, if the time since the last activity is greater than the value set in this parameter, the				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
download operation is aborted.				
Gather-Service-Forward-Get_User-Agent-Override This string determines the User-Agent: string in the HTTP headers		Default value: None Setting type: String Value range: N/A Task available ?: No Component restart required ?: Yes	Server, Relay, Agent	9.2 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
that the BES Gather Service uses. This can be used to help work strict proxy configurations.				

Top (on page 406)

HTTPS

Name/Description	References	Values	Component(s) affected	Version(s) applicable
_BESRelay_Download_UseHttps The `_BESRelay_Download_UseHttps` setting controls how protocols are handled for server downloads, allow-	Managing Downloads *(on page 1099)*	Default value: 0 Setting type: Numeric Value range: • 0 - Uses the protocol defined in the URL. • 1 - Forces the use of HTTPS if the	Relay, Server **Note:** This setting is supported also on the BigFix Relay starting from Version 11.0.6.	11.0.5 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
ing you to enforce secure HTTPS connections when needed.			URL specifies HTTP. • 2 - Attempts to download using HTTPS first. If HTTPS fails, it uses HTTP for URLs starting with http://.	

Name/Description	References	Values	Component(s) affected	Version(s) applicable
		Task available? No Component restart required? Yes		
_BESGather_Use_Https You can use the HTTPS protocol to get license updates or gather	Customizing HTTPS for Gathering *(on page 127)*	Default value 2 Setting type Numeric Value range • 0 - The server uses the protocol de-	Server	9.5 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
external sites directly on a BigFix server or in an air-gapped environment. To enable the HTTPS protocol,			fined in the URL. • 1 - The server tries to gather all sites using the HTTPS protocol only. • 2 - The server first	

Name/Description	References	Values	Component(s) affected	Version(s) applicable
set the client keyword `_-BES-Gather_-Use_-Https` to 1. After enabling HTTPS, you can create or download a package of cer-			tries to gather all sites using the HTTPS protocol. If the server fails to gather a site using HTTPS, it will try to	

Name/Description	References	Values	Component(s) affected	Version(s) applicable
tificates that you want to trust. The BigFix server validates the certificates during its gathering process.		gather again using the HTTP protocol. Task available ? No Component restart required ? Yes		
_BESGather_CACert	Customizing HTTPS for Gathering *(on page 127)*	Default None	Server	9.5 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
Path of the downloaded set of trusted certificates.		value Setting type: String (path)		

Top (on page 406)

Deployment Encoding

Name/Description	References	Values	Component(s) affected	Version(s) applicable
_BESClient_DeploymentEncoding_IANAName This setting is mandato-	Login BigFix Console failed with error - 'The application's codepage (932) is not same as server's codepage (1252)' BigFix client backward compatibility	Default value: Empty Setting type: String Value range: Standard IANA	Client, Server	All

Name/Description	References	Values	Component(s) affected	Version(s) applicable
ry on each BigFix V9.5 client in environments where the version of BigFix installed on the server is earlier than	BigFix support for Unicode characters	values Platform: All Component restart required ?: Yes		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
V9.5. It specifies the deployment encoding to use when communicating with the infrastructure. The value for				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
this setting defaults to "windows-1252" on Mac clients. Other available values are: windows-874, windows-932/Shift_JIS, windows-936/GBK, windows-949/windows-949-2000, windows-950/Big5,				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
windows-1250, windows-1251, windows-1252, windows-1253, windows-1254, windows-1255, windows-1256, windows-1257, windows-1258. **Note:** This setting is ignored on the client side				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
beginning in version 9.5.	• Mac and Android use UTF-8 and can not be changed by setting this option. • Unix default is de-			

Name/Description	References	Values	Component(s) affected	Version(s) applicable
	termined by the client locale environment variables (ex: LC_ALL or LC_CTYPE). • Windows default is determined by the			

Name/Description	References	Values	Component(s) affected	Version(s) applicable
 language for non-Unicode programs selected in the Windows control panel. On the server side, this setting is re-				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
quired on Linux and a specific warning is logged if the setting is missing. It's used as default value for _-BES-				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
Client_Fxf_Encoding_IANAName.				

Top (on page 406)

Logging

Name/Description	References	Values	Component(s) affected	Version(s) applicable
_BES_Client_Log_TimestampsDetail This configuration setting will allow		Default value: 0 (Disabled) Setting type: Boolean Value range: (0) - Do not log date/timestamps on	Client	11.0.3 and later

Name/Description	References	Values		Component(s) affected	Version(s) applicable
you to specify the desired Agent log file format. By default, the setting is disabled. Setting the value to 1, al-			each line (1) - Log date/ time- stamps on each line		
		Task available ?	No		
		Component restart required ?	No		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
lows you to generate an Agent log file with date/time-stamps information in front of each line that is recorded.				
_BESClient_EMsg_Detail	Data Collection: BigFix Client	Default: 0 (none)	Client	9.2 and later

Name/Description	References	Values		Component(s) affected	Version(s) applicable
This configuration setting will enable the BigFix Client debug log that will give information about the BigFix Client		value			
		Setting type	Numeric (detail level)		
		Value range	0 - 10,000 • 0 - none • 1 - critical error • 10 - debug • 10,000 - verbose		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
activity. The higher the level of detail, the greater the detail of the messages that are put into the file and/		Task available? Yes		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
or NT event log. **Note:** Setting this value too high can cause the log file to grow extremely quickly filling up				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
disk space.				
_BESClient_EMsg_File Full path to file to store activity messages. If empty string and the computer is not Win9x, ac-	Data Collection: BigFix Client	Setting type: String (full path) Task available?: Yes	Client	9.2 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
tivity messages are added to the NT event log. If you change the name or the path of this file, to avoid character display prob-				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
lems, ensure to use names that have only standard ASCII characters and not extended ASCII characters. **Note:** The _-BES-				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
 Client_EMsg_Detail setting must be greater than 0 to use this option.				
_BESClient_EMsg_MaxSize Defines the maximum size of	Data Collection: BigFix Client	Default value: 52,428,800 (50 MB) Setting type: Numeric (bytes)	Client	9.2 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
the single client debug log file. **Note:** Starting from Version 10 Patch 1, when the file reaches the maximum size, the log		Value range: 0 - 104,857,600(100 MB) Platform: All		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
 file is rotated. This means that the whole disk usage for the logs will be eleven times the value of this setting.				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
_BESClient_Log_MaxSize Size of daily log file. When the log becomes this big, it is renamed to 'date'.back (unless this file already exists)	Data Collection: BigFix Client	Default value: 512,000 Setting type: Numeric (bytes) Value range: 0 - 4,294,967,295 Task available? No	Client	9.2 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
and then the log for the day is restarted. Result is that for any particular day, you have the first part of the day, and				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
the last part of the day.				
_BESClient_Log_Days Number of days to save Client log files.	Data Collection: BigFix Client	Default value: 10 Setting type: Numeric (days) Value range: 1 - 366 Task available? No	Client	9.2 and later
_BESRelay_Log_Verbose A non-zero	Enabling debug/verbose logging for the BES Root Server and BES Relay services *(on page 1228)*	Default value: 0 (disabled) Setting type: Boolean	Relay	

Name/Description	References	Values		Component(s) affected	Version(s) applicable
value will enable verbose logging on the BigFix relay for troubleshooting purposes. This setting increases the in-		Value range	(True) - Enable Verbose Logs 0 (False) - Disable Verbose Logs		
		Task available	No ?		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
formation written to the existing logfile.txt and should not be left on during normal operation especially				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
on the root server where its performances can be compromised.				
_BESClient_Linux-Patch_enable_debug_log This configuration setting will:		Default value: 0 (disabled) Setting type: Boolean Value range: • 0 ◦ Disable DEBUG mode for	Client	9.2 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
• Enable/disable DEBUG mode for deployment logs (EDR_Deployment-Results.txt), which generates on BigFix client (/var/		• 1	deployment logs while patching. ◦ EDR deployment files are deleted after patching. ◦ Enable DEBUG mode for deployment	

Name/Description	References	Values	Component(s) affected	Version(s) applicable
opt/BES-Client/EDRDeployData) while patching. • Generate/delete EDR deployment files like EDR_-Yum-config, EDR_-Repomd-Request, EDR_-Repomd-		Task available No ?	logs while patching. ◦ EDR deployment files are retained even after patching is completed.	

Name/Description	References	Values	Component(s) affected	Version(s) applicable
Mapping, EDR_Repo-Refresh, EDR_Metadata-Request, and EDR_Metadata-Mapping on BigFix client after patching.				

Auditing

<table>
<tr><th>Name/Description</th><th>References</th><th>Values</th><th>Component(s) affected</th><th>Version(s) applicable</th></tr>
<tr><td>_BESRootServer_Audit_Verbosity
It controls whether SSL connections are logged in the server audit log. Specify _BESRootServer_Audit_Verbosity = all to log the all information about both failed and successful SSL connections. Specify _BESRootServer_Audit_Verbosity = null or remove the option from the settings to disable the logging of SSL connections.</td><td></td><td><table><tr><td>Default value</td><td>None</td></tr><tr><td>Setting type</td><td>String</td></tr><tr><td>Value range</td><td>all, null</td></tr><tr><td>Task available ?</td><td>No</td></tr></table></td><td>Server components</td><td>All</td></tr>
</table>

Name/Description	References	Values	Component(s) affected	Version(s) applicable
You can specify this setting either in the besclient.config file or on the Console by accessing the Computers domain, right-clicking the Client where you want to log SSL information and selecting Edit Settings.				
_Audit_Logging_Log-MaxSize It sets the maximum size (in Bytes) for the audit log file of the BigFix Server. When the	Configuring logs *(on page 1154)*	Default value: 100 MB (104857600 Bytes) Setting type: Numeric (bytes) Value range: 0-4294967296 (4 GB)	BigFix Server	All

Name/Description	References	Values		Component(s) affected	Version(s) applicable
size reaches its maximum value, the log file is renamed and a new file is created. The log file format is server_audit.datetime.log. To remove the old audit log files, use the BigFix Administration tool Audit Cleaner.		Task available ?	No		
_BESAdminAudit_Logging_LogDirectoryPath It sets the audit log directory path of the BigFix Administration Tool. The directory may	Configuring logs (on page 1154)	Default value	• On Windows: ◦ For each user: `C:\Users`	BigFix Administration Tool	10 Patch 8

Name/Description	References	Values	Component(s) affected	Version(s) applicable
not exist, but it must be creatable. **Note:** If this client setting is not used, on Windows the audit log files may not be deleted from the Audit Cleaner **Removes old au-**			`\<USER-NAME>\App-Data\Local\BigFix` ◦ When the BigFix Administration Tool is invoked by a Fixlet or run	

Name/Description	References	Values		Component(s) affected	Version(s) applicable
dit log files function. For more details, refer to Configuring logs *(on page 1154)*.			by the Local System user: `C:\Windows\System32\config\systemprofile\AppData\Local\BigFix` • On Linux: `/var/log`		

Name/Description	References	Values		Component(s) affected	Version(s) applicable
		Setting type	String		
		Value range	all, null		
		Task available ?	No		
_BESAdminAudit_Logging_LogMaxSize It sets the maximum size (in Bytes) for the audit log file of the BigFix Administration Tool. When the size reaches its maximum value, the log file is renamed and a new file is created. The log file format is besadmin_-	Configuring logs (on page 1154)	Default value	100 MB (104857600 Bytes)	BigFix Administration Tool	10 Patch 8
		Setting type	Numeric (bytes)		
		Value range	0-4294967296 (4 GB)		
		Task available ?	No		

Name/Description	References	Values		Component(s) affected	Version(s) applicable
audit.datetime.log. To remove the old audit log files, use the BigFix Administration tool Audit Cleaner.					
_BESServerAudit_Logging_LogDirectoryPath It sets the audit log directory path of the BigFix Server. The directory may not exist, but it must be creatable. The name of the audit log file is server_audit.log.	Configuring logs (on page 1154)	Default value	• On Windows: `C:\Program Files (x86)\BigFix Enterprise\BES Server` • On Lin-	BigFix Server	11 Patch 4

Name/Description	References	Values		Component(s) affected	Version(s) applicable
			ux: `/var/opt/BESServer`		
		Setting type	String		
		Task available ?	No		
		Component restart required ?	Yes		

Top (on page 406)

CPU Usage

Name/Description	References	Values		Component(s) affected	Version(s) applicable
_BESClient_Resource_WorkIdle	Virtualized environ-	Default value	10	Client	All

Name/Description	References	Values		Component(s) affected	Version(s) applicable
The BigFix Client works (evaluate relevance) for a designated amount of time then sleeps for a designated amount of time. This setting controls how many milliseconds to work before going to sleep in each cycle. If this number is high in comparison to the _BESClient_Re-	ments and virtual machines (on page 1225) Configuring Client CPU Utilization (on page 1132)	Setting type	Numeric (milliseconds)		
		Value range	1 - 500		
		Task available ?	Yes		
		Platform	All		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
source_-Sleepl-dle set-ting, then the Big-Fix Client will evalu-ate Fixlet relevance faster, but the CPU usage is higher.				
_BESClient_Re-source_WorkNor-mal The Big-Fix Client controls the CPU work limit in normal evaluation mode. It represents the num-ber of mil-liseconds	How should I config-ure the Bigfix Client to run on an virtu-al serv-er that hosts multiple images?	Default value: 50 Setting type: Numer-ic (mil-lisec-onds) Value range: 1 - 500 Plat-form: All Com-ponent restart: No	Client	All

Name/Description	References	Values	Component(s) affected	Version(s) applicable
to work to balance CPU when in normal mode. By default the BigFix Client works for 50 milliseconds, then sleeps for 50 milliseconds (_BESClient_Resource_SleepNormal) when in normal mode.		re-quired ?		
_BESClient_Resource_SleepIdle The BigFix Client works (evaluate	Configuring Client CPU Utilization *(on page 1132)*	Default value: 480 Setting type: Numeric (milliseconds)	Client	All

Name/Description	References	Values		Component(s) affected	Version(s) applicable
relevance) for a designated amount of time then go to sleep for a designated amount of time. This setting controls how many milliseconds to sleep after working in each cycle. If this number is high in comparison to the _BESClient_Resource_WorkIdle setting, then		Value range	1 - 500		
		Platform	All		
		Task available ?	Yes		

Name/Description	References	Values		Component(s) affected	Version(s) applicable
the BigFix Client takes longer to evaluate Fixlet relevance, but the CPU usage is lower.					
_BESClient_Resource_SleepNormal The BigFix Client controls the CPU sleep interval in normal evaluation mode (as opposed to 'Idle' mode). It represents the number of millisec-	How should I configure the Bigfix Client to run on an virtual server that hosts multiple images?	Default value	50	Client	All
		Setting type	Numeric (milliseconds)		
		Value range	1 - 3,600,000 (1 hour)		
		Task available ?	No		
		Platform	All		
		Component	No		

Name/Description	References	Values		Component(s) affected	Version(s) applicable
onds to sleep after evaluating for _BESClient_Resource_WorkNormal milliseconds. The BigFix Client only sleeps for up to 1000 milliseconds to balance the CPU usage.		restart required ?			
_BESClient_Resource_QuietEnable Quiet mode causes the BigFix Client to not actively process Fixlets,	Virtualized environments and virtual machines *(on page 1225)*	Default value Setting type Value range	0 (false) Boolean 1 (True) - enable quiet mode 0 (False) - disable	Client	All

<table>
<tr><th>Name/Description</th><th>Refer-ences</th><th>Values</th><th>Compo-nent(s) affected</th><th>Version(s) applicable</th></tr>
<tr><td>send in reports, respond to com-mands, etc. It will basical-ly sit idle until the wakeup time is enabled. A BigFix Client can be put in quiet mode for a maxi-mum of 20 hours a day.

Note: If you put a Big-Fix</td><td></td><td><table><tr><td></td><td>quiet mode</td></tr><tr><td>Plat-form</td><td>All</td></tr><tr><td>Task avail-able ?</td><td>No</td></tr></table></td><td></td><td></td></tr>
</table>

Name/Description	References	Values	Component(s) affected	Version(s) applicable
Client in quiet mode, it will not send any reports to the BigFix Server or respond to any commands so use caution				

<table>
<tr><th>Name/Description</th><th>Refer-ences</th><th>Values</th><th>Compo-nent(s) affected</th><th>Version(s) applicable</th></tr>
<tr><td> when en-abling this set-ting.</td><td></td><td></td><td></td><td></td></tr>
<tr><td>_BESClient_Re-source_QuietSe-conds
If quiet mode is enabled, you can specify the number of seconds the Big-Fix Client should stay quiet. This num-ber must be greater than 0 and less or equal to 72000 (20 hours) in</td><td>Virtual-ized en-viron-ments and vir-tual ma-chines (on page 1225)</td><td><table><tr><td>Default value</td><td>0</td></tr><tr><td>Setting type</td><td>Numer-ic (sec-onds)</td></tr><tr><td>Value range</td><td>0 - 72,000(20 Hours)</td></tr><tr><td>Plat-form</td><td>All</td></tr><tr><td>Task avail-able ?</td><td>No</td></tr></table></td><td>Client</td><td>All</td></tr>
</table>

Name/Description	References	Values	Component(s) affected	Version(s) applicable
order for the BigFix Client to enter quiet mode. In quiet mode, the BigFix Client checks once a minute to determine if the quiet time has elapsed. However, the BigFix Client will quickly respond to service shutdown requests at any time.				
_BESClient_Resource_QuietStartTime	Virtualized environ-	Default value: None	Client	All

Name/Description	References	Values		Component(s) affected	Version(s) applicable
If quiet mode is enabled, you can specify a time for the BigFix Client to enter quiet mode. Set the time as a string in the form hh:mm where hh is number between 0 and 23, and mm is a number between 0 and 59. This is the local time for which the BigFix Client will quiet down to	ments and virtual machines *(on page 1225)*	Setting type	String		
		Value range	hh:mm		
		Platform	All		
		Task available ?	No		

<table>
<tr><th>Name/Description</th><th>References</th><th>Values</th><th>Component(s) affected</th><th>Version(s) applicable</th></tr>
<tr><td>no activity. If the value is not properly formatted, the client will not enter quiet mode.</td><td></td><td></td><td></td><td></td></tr>
<tr><td>_BESClient_Resource_InterruptSeconds
When a BigFix Client receives a UDP message but is too busy processing a relevance clause, this configuration setting determines the length</td><td></td><td><table><tr><td>Default value</td><td>60</td></tr><tr><td>Setting type</td><td>Numeric (seconds)</td></tr><tr><td>Value range</td><td>1 - 1200</td></tr><tr><td>Platform</td><td>All</td></tr><tr><td>Task available ?</td><td>Yes</td></tr></table></td><td>Client</td><td>All</td></tr>
</table>

<table>
<tr><th>Name/Description</th><th>References</th><th>Values</th><th>Component(s) affected</th><th>Version(s) applicable</th></tr>
<tr><td>of time the BigFix Client will wait before it stops evaluating the relevance clause to deal with the UDP message.</td><td></td><td></td><td></td><td></td></tr>
<tr><td>_BESClient_Resource_StartupNormalSpeed
It sets the new client to do work (evaluate relevance) until it gets through a complete pass of all the content. In this way a new client</td><td></td><td><table><tr><td>Default value</td><td>0</td></tr><tr><td>Setting type</td><td>Boolean</td></tr><tr><td>Value range</td><td>0 or 1</td></tr><tr><td>Task available ?</td><td>No</td></tr><tr><td>Platform</td><td>All</td></tr><tr><td>Component restart</td><td>Yes</td></tr></table></td><td>Client</td><td>All</td></tr>
</table>

Name/Description	References	Values		Component(s) affected	Version(s) applicable
can subscribe to sites and complete a full first pass quickly.		re-quired ?			
_BESClient_Resource_PowerSaveDeepSleepEnable It sets the client in deep sleep mode for a designated amount of time. The client ignores UDP messages, queuing them except for send client alert request message.		Default value	0	Client	All
		Setting type	Boolean		
		Value range	0 or 1		
		Task available ?	No		
		Platform	All		
		Component restart required ?	No		

<table>
<tr><th>Name/Description</th><th>References</th><th>Values</th><th>Component(s) affected</th><th>Version(s) applicable</th></tr>
<tr><td>When setting to 1, the client switches to the deep sleep mode.</td><td></td><td></td><td></td><td></td></tr>
<tr><td>_BESClient_Resource_AccelerateForPendingMessage
This setting affects how _BESClient_Resource_PowerSaveEnable behaves when it receives UDP messages. When set to 1, the client</td><td></td><td><table><tr><td>Default value</td><td>0</td></tr><tr><td>Setting type</td><td>Boolean</td></tr><tr><td>Value range</td><td>0 or 1</td></tr><tr><td>Task available ?</td><td>No</td></tr><tr><td>Platform</td><td>All</td></tr><tr><td>Component restart required ?</td><td>No</td></tr></table></td><td>Client</td><td>All</td></tr>
</table>

Name/Description	References	Values	Component(s) affected	Version(s) applicable
avoids sleeping when it detects pending commands. That is, the client wakes up immediately if there are pending commands. When setting to 0, the client always waits for timer expiration. **Note:** This setting does				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
not affect the _-BES-Client_-Resource_-Power-Save-Deep-Sleep-Enable setting in any manner.				
_BESClient_Resource_PowerSaveEnable It sets the client		Default value: 0 Setting type: Boolean	Client	All

Name/Description	References	Values		Component(s) affected	Version(s) applicable
in sleep mode for a designated amount of time. The client continues to process the UDP messages based on the value set for _BESClient_Resource_-AccelerateFor-Pending-Message. When set to 1, the client switches to the simple sleep mode. This option is		Value range	0 or 1		
		Task available ?	No		
		Platform	All		
		Component restart required ?	No		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
useful for tablets and laptops.				

There are six values of `BESClient_Resource_PowerSaveTimeout`*X* (where *X* ranges from 0 to 5) that control how many minutes the client sleeps; they will be applied based on both Network and Power Supply conditions.

Network	Power	Applied timeout	Default value
Connected	AC or Battery - High	`BESClient_Resource_PowerSaveTimeout0`	10
Connected	Battery - Medium	`BESClient_Resource_PowerSaveTimeout1`	20
Connected	Battery - Low	`BESClient_Resource_PowerSaveTimeout2`	60
Not connected	AC or Battery - High	`BESClient_Resource_PowerSaveTimeout3`	720
Not connected	Battery - Medium	`BESClient_Resource_PowerSaveTimeout4`	1,440

Name/Description	References	Values	Component(s) affected	Version(s) applicable

Network	Power	Applied timeout	Default value
Not connected	Battery - Low	`BESClient_Resource_PowerSaveTimeout5`	2,880

In cases where the network is "Connected", if there is at least one network interface with internet connection, the battery power is evaluated as follows:

OS	Battery - High	Battery - Medium	Battery - Low
Windows / OS X	BatteryLifePercent > 66%	33% <= BatteryLifePercent <= 66%	BatteryLifePercent < 33%
Unix / Android	BatteryLifePercent = 100%	100% <= BatteryLifePercent <= 20%	BatteryLifePercent < 20%

Note:

- For enterprise servers (which usually do not have a battery), only `_BESClient_Resource_PowerSaveTimeout0` and `_BESClient_Resource_PowerSaveTimeout3` are used.
- These parameters take effect only if `_BESClient_Resource_PowerSaveEnable` or `_BESClient_Resource_PowerSaveDeepSleepEnable` is set to 1.

Name/Description	References	Values		Component(s) affected	Version(s) applicable
_BESClient_Resource_PowerSaveTimeout0 This setting controls how many minutes the client sleeps when the network is present at high power.	Virtualized environments and virtual machines *(on page 1225)*	Default value	10	Client	All
		Setting type	Numeric (minutes)		
		Value range	0 - 2,147,483,647		
		Task available ?	No		
		Platform	All		
		Component restart required ?	No		
_BESClient_Resource_PowerSaveTimeout1 This setting controls how many minutes the client	Virtualized environments and virtual machines *(on page 1225)*	Default value	20	Client	All
		Setting type	Numeric (minutes)		
		Value range	0 - 2,147,483,647		

Name/Description	References	Values		Component(s) affected	Version(s) applicable
sleeps when the network is present at medium power.		Task available ?	No		
		Platform	All		
		Component restart required ?	No		
_BESClient_Resource_PowerSaveTimeout2 This setting controls how many minutes the client sleeps when the network is present at low power.	Virtualized environments and virtual machines *(on page 1225)*	Default value	60	Client	All
		Setting type	Numeric (minutes)		
		Value range	0 - 2,147,483,647		
		Task available ?	No		
		Platform	All		
		Component restart	No		

Name/Description	References	Values		Component(s) affected	Version(s) applicable
		re-quired ?			
_BESClient_Resource_PowerSaveTimeout3 This setting controls how many minutes the client sleeps when the network is not present at high power.	Virtualized environments and virtual machines *(on page 1225)*	Default value Setting type Value range Task available ? Platform Component restart required ?	720 Numeric (minutes) 0 - 2,147,483,647 No All No	Client	All
_BESClient_Resource_PowerSaveTimeout4 This setting controls how	Virtualized environments and virtual machines	Default value Setting type	1440 Numeric (minutes)	Client	All

Name/Description	References	Values		Component(s) affected	Version(s) applicable
many minutes the client sleeps when the network is not present at medium power.	*(on page 1225)*	Value range	0 - 2,147,483,647		
		Task available ?	No		
		Platform	All		
		Component restart required ?	No		
_BESClient_Resource_PowerSaveTimeout5 This setting controls how many minutes the client sleeps when the network is not present at low power.	Virtualized environments and virtual machines *(on page 1225)*	Default value	2,880	Client	All
		Setting type	Numeric (minutes)		
		Value range	0 - 2,147,483,647		
		Task available ?	No		
		Platform	All		

Name/Description	References	Values		Component(s) affected	Version(s) applicable
		Component restart required ?	No		
_BESClient_Query_-WorkTime This setting is used, together with **_BESClient_-Query_-SleepTime**, to configure the QnA CPU throttling. It determines how long the QnA processing a query runs before sleeping.	Query *(on page 807)*	Default value	10	Client	9.5 and later
		Setting type	Numeric (milliseconds)		
		Value range	1 - 500		
		Platform	All		
		Component restart required ?	No		

Name/Description	References	Values		Component(s) affected	Version(s) applicable
_BESClient_Query_SleepTime This setting is used, together with **_BESClient_Query_WorkTime**, to configure the QnA CPU throttling. It determines how long the QnA processing a query sleeps before continuing to run.	Query (on page 807) Run WebUI Query request with REST API	Default value	480	Client	9.5 and later
		Setting type	Numeric (milliseconds)		
		Value range	1 - 3,600,000		
		Component restart required ?	No		
		Platform	All		
_BESClient_Resource_WorkFastHashVerify		Default value	20 milliseconds	Client	10 Patch 2

Name/Description	References	Values		Component(s) affected	Version(s) applicable
The BigFix client controls the CPU work limit, in fast evaluation mode. This setting shows the milliseconds of the CPU work.		Setting type	Numeric		
		Value range	1 millisecond - 500 milliseconds		
		Component restart required ?	No		
_BESClient_Resource_SleepFastHashVerify The BigFix client controls the CPU sleep interval, in fast evaluation mode. This setting shows the mil-		Default value	60 milliseconds	Client	10 Patch 2
		Setting type	Numeric		
		Value range	1 millisecond - 1 hour (1000*60*60 milliseconds)		
		Component	No		

Name/Description	References	Values		Component(s) affected	Version(s) applicable
liseconds of CPU sleep, after evaluating for the _BESClient_Resource_WorkFastHashVerify value in milliseconds.		restart required ?			
_BESClient_Resource_ScaleWorkTimeWithCPUCores This setting enables CPU scaling on multi-core systems. By setting the value to 1 (enabled), the system gath-		Default value Setting type Value range Task Available ? Component restart	0 (Disabled) Boolean 0 (Disabled) - 1 (Enabled) No Yes	Client	11 Patch 7

Name/Description	References	Values	Component(s) affected	Version(s) applicable
ers the number of CPU cores, and if it is greater than 1, the system will perform adjustment of work and sleep variables as required.		required ?		

Top (on page 406)

Proxy agent

Name/Description	References	Values	Component(s) affected	Version(s) applicable
_ProxyAgent_ManagementRights_MinRefreshInterval		Default value: 0 (disabled)	Proxy agent	From 9.5.16

<table>
<tr><th>Name/Description</th><th>References</th><th>Values</th><th>Component(s) affected</th><th>Version(s) applicable</th></tr>
<tr><td>When enabled, it controls the minimum frequency at which the Proxy Agent evaluates the Management Rights actions. By default it is disabled. If enabled, the devices managed by the Proxy Agent will report faster on the BigFix console but changes against the operators management rights will not be evaluated immediately.</td><td></td><td><table>
<tr><td>Setting type</td><td>Numeric (Hours)</td></tr>
<tr><td>Value range</td><td>Value range 1 - 8,760 (1 hour - 365 days)</td></tr>
<tr><td>Task Available</td><td>No</td></tr>
<tr><td>Component restart required ?</td><td>Yes</td></tr>
</table></td><td></td><td></td></tr>
</table>

Proxy configuration

Client

Name/Description	References	Values		Component(s) affected	Version(s) applicable
_BESClient_Comm_-ProxyServer Sets the hostname that is used to reach the proxy.	Setting up a proxy connection on a client	Default value	None	Client	All
		Setting type	String		
		Value range	N/A		
		Platform	All		
		Component restart required ?	No		
_BESClient_Comm_-ProxyPort Sets the port that is used to communicate with the proxy.	Setting up a proxy connection on a client	Default value	None	Client	All
		Setting type	Numeric		
		Value range	0 - 65,535		
		Platform	All		

Name/Description	References	Values		Component(s) affected	Version(s) applicable
		Component restart required ?	No		
_BESClient_Comm_-ProxyUser Sets the user name that is used to authenticate with the proxy if the proxy requires authentication.	Setting up a proxy connection on a client	Default value	None	Client	All
		Setting type	String		
		Value range	N/A		
		Platform	All		
		Component restart required ?	No		
		Mandatory ?	No (depending on the authentication method)		

Name/Description	References	Values		Component(s) affected	Version(s) applicable
_BESClient_Comm_-ProxyPass Sets the password that is used to authenticate with the proxy if the proxy requires authentication.	Setting up a proxy connection on a client	Default value	None	Client	All
		Setting type	String		
		Value range	N/A		
		Platform	All		
		Component restart required ?	No		
		Mandatory ?	No (depending on the authentication method)		
_BESClient_Comm_-ProxyManualTryDirect Specifies whether direct connections can be used.	Setting up a proxy connec-	Default value	0	Client	All
		Setting type	Numeric		

Name/Description	References	Values		Component(s) affected	Version(s) applicable
This setting applies if the connection to the proxy uses the hostname or IP Address and port number that are specified in `_BESClient_Comm_ProxyServer` and `_BESClient_Comm_ProxyPort`. These values are available:	tion on a client	Value range	0 - 2 **0** Do not try direct connection. **1** Try direct connection if a proxy connection cannot		

Name/Description	References	Values		Component(s) affected	Version(s) applicable
			be established. **2** Try direct connection first.		
		Platform	All		
		Component restart required ?	No		
		Mandatory ?	No		
_BESClient_Comm_-ProxyAutoDetect Specifies whether the	Setting up a proxy con-	Default value	1	Client	All
		Setting type	Boolean		

Name/Description	References	Values		Component(s) affected	Version(s) applicable
system uses the proxy configuration settings that are specified for Internet Explorer. **Important:** Ensure that at least one user is logged in to the client to be able to get the Internet Explorer config-	nection on a client	Value range	0 - 1 **0** Use the values that are specified in `_-BES-Client_-Comm_-Proxy-Server` and `_-BES-`		

Name/Description	References	Values		Component(s) affected	Version(s) applicable
uration settings.			`Client_Comm_ProxyPort`. **1** Use the Internet Explorer configuration settings.		
		Component	No		

<table>
<tr><th>Name/Description</th><th>References</th><th>Values</th><th>Component(s) affected</th><th>Version(s) applicable</th></tr>
<tr><td></td><td></td><td><table><tr><td>restart required ?</td><td></td></tr><tr><td>Platform</td><td>Windows</td></tr><tr><td>Mandatory ?</td><td>No</td></tr></table></td><td></td><td></td></tr>
<tr><td>_BESClient_Comm_-ProxyAutoDetectTryDirect
Specifies whether direct connections can be used when the system uses the proxy configuration settings that are specified for Internet Explorer. This setting is valid only if <code>_-BESClient_-Comm_Proxy-</code></td><td>Setting up a proxy connection on a client</td><td><table><tr><td>Default value</td><td>1</td></tr><tr><td>Setting type</td><td>Numeric</td></tr><tr><td>Value range</td><td>0 - 2
0
Do not try direct connection.
1
Try di-</td></tr></table></td><td>Client</td><td>All</td></tr>
</table>

Name/Description	References	Values		Component(s) affected	Version(s) applicable
`AutoDetect = 1`.			rect con-nec-tion if a proxy con-nec-tion can-not be es-tab-lished. **2** Try di-rect con-nec-tion first.		
		Platform	Windows		

Name/Description	References	Values		Component(s) affected	Version(s) applicable
		Component restart required ?	No		
		Mandatory ?	No		
AutoProxyRawProxyList Specifies a blank space delimited list of proxies to try to connect to. **Note:** This setting is saved in the registry under	Setting up a proxy connection on a client	Default value	None	Client	All
		Setting type	String		
		Value range	NA		
		Platform	Windows		
		Component restart required ?	No		
		Mandatory ?	No		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
 the following key `HKLM\Software\BigFix\Enterprise Client\GlobalOptions` and not user-configured. The information provided here is only for trou-				

<table>
<tr><th>Name/Description</th><th>References</th><th>Values</th><th>Component(s) affected</th><th>Version(s) applicable</th></tr>
<tr><td>bleshooting purposes.</td><td></td><td></td><td></td><td></td></tr>
<tr><td>AutoProxyRawBypassList
Specifies a blank space delimited list of URLs to contact directly without passing through the proxy. You can use the "*" as a wildcard.
Note: This setting is saved in the reg-</td><td>Setting up a proxy connection on a client</td><td><table><tr><td>Default value</td><td>None</td></tr><tr><td>Setting type</td><td>String</td></tr><tr><td>Value range</td><td>NA</td></tr><tr><td>Platform</td><td>Windows</td></tr><tr><td>Component restart required ?</td><td>No</td></tr><tr><td>Mandatory ?</td><td>No</td></tr></table></td><td>Client</td><td>All</td></tr>
</table>

Name/Description	References	Values	Component(s) affected	Version(s) applicable
istry under the following key `HKLM\Software\BigFix\Enterprise Client\GlobalOptions` and not user-configured. The information provided here is on-				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
ly for troubleshooting purposes.				

Top (on page 406)

Server/Relay

Name/Description	References	Values	Component(s) affected	Version(s) applicable
Enterprise Server-ClientRegister_Proxy-Server A BigFix relay can be set up to use a proxy server when downloading files. This	Setting up a proxy connection on a relay	Default value: None Setting type: String Value range: N/A	Server, Relay	9.2 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
could be useful if there is a proxy between the BigFix relay and the BigFix server or if the BigFix relay is connecting directly to the internet for downloads and must use a proxy. This setting controls the hostname used to reach the proxy.	Configuration of Microsoft Azure Cloud Plug-in with proxy BigFix server cannot gather external sites (error: 17NotASignedMessage) Software Distri-	Task available ? No		

<table>
<tr><th>Name/Description</th><th>References</th><th>Values</th><th>Component(s) affected</th><th>Version(s) applicable</th></tr>
<tr><td></td><td>bution task fails with error "Unexpected HTTP Response 502"</td><td></td><td></td><td></td></tr>
<tr><td>_Enterprise Server_-ClientRegister_Proxy-Port
A BigFix relay can be set up to use a proxy server when downloading files. This could be useful if there is a proxy between the BigFix relay and the</td><td>Setting up a proxy connection on a relay
Configuration of Microsoft Azure</td><td><table><tr><td>Default value</td><td>None</td></tr><tr><td>Setting type</td><td>Numeric</td></tr><tr><td>Value range</td><td>0 - 65,535</td></tr><tr><td>Task available ?</td><td>No</td></tr></table></td><td>Server, Relay</td><td>9.2 and later</td></tr>
</table>

Name/Description	References	Values	Component(s) affected	Version(s) applicable
BigFix server or if the BigFix relay is connecting directly to the internet for downloads and must use a proxy. This setting controls the port used by the proxy server.	Cloud Plugin with proxy BigFix server cannot gather external sites (error: 17NotASignedMessage)			
Enterprise Server-ClientRegister_Proxy-User A BigFix relay can be set up to use a proxy server when downloading files. This	Setting up a proxy connection on a relay	Default value: None Setting type: String (username) Value range: N/A	Server, Relay	9.2 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
could be useful if there is a proxy between the BigFix relay and the BigFix server or if the BigFix relay is connecting directly to the internet for downloads and must use a proxy. This setting controls the username used to authenticate with the proxy if the proxy requires authentication.	Configuration of Microsoft Azure Cloud Plug-in with proxy BigFix server cannot gather external sites (error: 17NotASignedMessage)	Task available ? No		
Enterprise Server-ClientRegister_Proxy-Pass	Setting up a	Default value None	Server, Relay	9.2 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
A BigFix relay can be set up to use a proxy server when downloading files. This could be useful if there is a proxy between the BigFix relay and the BigFix server or if the BigFix relay is connecting directly to the internet for downloads and must use a proxy. This setting controls the password used to authenticate with the proxy if the proxy re-	proxy connection on a relay Configuration of Microsoft Azure Cloud Plug-in with proxy Best practices to consider when defining a proxy con-	Setting type: String Value range: N/A Task available ?: No		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
quires authentication. Warning: The password will be stored in plain text within the registry.	nection BigFix server cannot gather external sites (error: 17NotASigned-Message)			
Enterprise Server-ClientRegister_Proxy-SecureTunnel If set, it defines whether or not the proxy is enforced to attempt tunneling. By default the proxy does	Best practices to consider when defining a proxy con-	Default value: false Setting type: Boolean Value range: 0 | 1 Mandatory ?: No	Relay, Server	9.2 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
not attempt tunneling.	nection			
Enterprise Server-ClientRegister_Proxy-AuthMethodsAllowed Restricts the set of authentication methods that can be used. You can specify more than one value separated by a comma. For information about restrictions affecting the supported authentication methods when using FIPS, see Setting a proxy con-	Setting up a proxy connection on a relay	Default value: None (Any) Setting type: String Value range: basic | digest | negotiate | nltm Mandatory ?: No	Relay, Server	9.2 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
nection on the server.				
Enterprise Server-ClientRegister_ProxyUseForDownstream-Comm If set to 1, this setting indicates that all downstream communications in your Endpoint Manager environment pass through the proxy.	Setting up a proxy connection on a relay	Default value: 0 Setting type: Boolean Value range: 0 (false) | 1 (true) Mandatory ?: No	Relay, Server	9.2 and later
Enterprise Server-ClientRegister_Proxy-ExceptionList Specifies the computers, for example the parent relay, domains and	Setting up a proxy connection on a relay	Default value: localhost, 127.0.0.1 (internal communications are not diverted to-	Relay, Server	9.2 and later

<table>
<tr><th>Name/Description</th><th>References</th><th>Values</th><th>Component(s) affected</th><th>Version(s) applicable</th></tr>
<tr><td>subnetworks that must be reached by the relay without passing through the proxy. Use the following format:
<pre>"localhost, 127.0.0.1, hostname1, hostname2, IP_Addr_A, IP_Addr_B,domain_Z, domain_Y, ..."</pre>By default internal com-</td><td>Configuration of Microsoft Azure Cloud Plug-in with proxy</td><td><table><tr><td></td><td>wards the proxy)</td></tr><tr><td>Setting type</td><td>String</td></tr><tr><td>Value range</td><td>N/A</td></tr><tr><td>Mandatory ?</td><td>No</td></tr></table></td><td></td><td></td></tr>
</table>

Name/Description	References	Values	Component(s) affected	Version(s) applicable
munications are not diverted towards the proxy. To maintain this behavior, ensure that you include localhost, 127.0.0.1 in the list of exceptions when specifying a value for this setting. **Note:** Ensure that you read Setting up a proxy connection to				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
learn more about using the proxy exception list on a relay thru the samples.				

Top (on page 406)

Registration

Name/Description	References	Values	Component(s) affected	Version(s) applicable
_BESClient_Register_StopOnFlushError To prevent that duplicated computers are generated on the Console when the BESClient file system goes temporarily in read-only mode.		Default value: 0 (false) Setting type: Boolean Value range: 0 (false) - 1 (true)	Client	9.5.8 and later
_BESClient_Register_IntervalSeconds The BigFix Client will contact the ClientRegister CGI on the BigFix Server periodically to report its IP address (this is important	What is the meaning of the message "skipping relay select. Regis-	Default value: 21,600 (6 hours) Setting type: Numeric (seconds) Value range: 600 - 82,800 (23 Hours) Task available ?: No	Client	All

Name/Description	References	Values	Component(s) affected	Version(s) applicable
if the BigFix Client changes IP addresses through DHCP), get the BigFix Server time, and get the BigFix Server version. If this setting is set too high, the BigFix Client may not receive UDP "pings" that tell it to gather the latest sites when new actions or new Fixlet content is available. If this setting is set too low, the BigFix Client will flood the BigFix Server	tering with current relay" in BES Client Debug log?			

<table>
<tr><th>Name/Description</th><th>References</th><th>Values</th><th>Component(s) affected</th><th>Version(s) applicable</th></tr>
<tr><td>with registration requests.</td><td></td><td></td><td></td><td></td></tr>
<tr><td>_BESClient_SecureRegistration
To set the value for the password needed to perform a manual registration to an authenticating relay. This setting is useful when a new client needs to be registered to an existing authenticating relay, and the authenticating relay uses a password to perform the key exchange. This setting is</td><td>Client could not register with the BigFix server successfully
Reasons for large group of BigFix Clients failing to register</td><td><table><tr><td>Default value</td><td>None</td></tr><tr><td>Setting type</td><td>String</td></tr><tr><td>Value range</td><td>Only ASCII characters</td></tr><tr><td>Platform</td><td>All</td></tr></table></td><td>Client</td><td>9.5.7 and later</td></tr>
</table>

Name/Description	References	Values	Component(s) affected	Version(s) applicable
read at client startup time only. In case of failure, the client tries to register again to an existing authenticating relay and follows the same behavior of the relay selection.	and appear offline BigFix Clients not choosing their specified BigFix Relay			
_BESRelay_Selection_-AutoSelectableRelay A value of "0" makes the BigFix relay unavailable for auto selection after the next actionsite propagation. A value of "1"	How can I control client relay selection for a particular	Default value: 1 (enabled) Setting type: Boolean Value range: 1 (enabled) - Enable Auto Selection	Relay	All

Name/Description	References	Values		Component(s) affected	Version(s) applicable
allows the BigFix relay to be used by BigFix clients for auto selection.	Relay or DSA Server?		0 (disabled) - Disable Auto Selection		
		Task available ?	No		
_BESClient_RegistrationManager_Register-With51Relays This configuration setting enables a BigFix 6.0 Client to gather from a BigFix 5.1 Relay for troubleshooting purposes (TEM 6.0+).	How can I control client relay selection for a particular Relay or DSA Server?	Default value	0 (disabled)	Client	All
		Setting type	Boolean		
		Value range	1 (enabled) 0 (disabled)		
		Task available ?	No		
Enterprise Server-ClientRegister_Client-	Proxy Configu-	Default value	1,440 (1 day)	Relay, Server	All

Name/Description	References	Values		Component(s) affected	Version(s) applicable
RegistrationExpirationPeriod This setting controls how quickly BigFix relays mark BigFix clients as expired. BigFix clients have this length of time to register again before their license is made available for use by another BigFix client. This setting could be used to increase the rate at which licenses are transferred from offline Big-	ration (Server/Relay) *(on page 663)*	Setting type	Numeric (minutes)		
		Value range	1 - 4,294,967,295		
		Task available ?	No		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
Fix clients to active BigFix clients when the total number of clients gets above the license seat count. **Note:** This setting also impacts the send/refresh notify action towards the BigFix clients.				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
 The send/refresh notify action is sent only if the time elapsed since the last registration is not bigger than the value of this setting.				

Name/Description	References	Values		Component(s) affected	Version(s) applicable
Enterprise Server-ClientRegister_MaxRelayPathLength This setting is used to control the maximum length a chain of BigFix relays can extend. The setting prevents circular chains of BigFix relays from forming by automatic BigFix relay selection. This setting could be used to stop long chains of BigFix relays from forming or create a maximum depth a BigFix		Default value	32	Relay, Server	All
		Setting type	Numeric		
		Value range	1 - 4,294,967,295		
		Task available ?	No		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
relay can be from the BigFix server.				

Top (on page 406)

Relay selection

Name/Description	References	Values		Component(s) affected	Version(s) applicable
_BESClient_RelaySelect_UseMastheadRootServer By default, a BigFix Client installed on the same computer as a BigFix Server connects to it via the loopback address. This behavior is	Configuring HA and DR on Windows	Default value	0	Client	11.0.4 and later
		Setting type	Boolean		
		Value range	0 (false) - 1 (true)		
		Task available ?	No		
		Component restart required ?	Yes		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
based on the assumption that the local BigFix Server is always the right one to contact. However, in a failover cluster environment, that assumption is wrong, because there are multiple BigFix Servers and the local one might not be the currently active one. If you turn on this setting, the BigFix Client will drop that assumption				

<table>
<tr><th>Name/Description</th><th>References</th><th>Values</th><th>Component(s) affected</th><th>Version(s) applicable</th></tr>
<tr><td>and contact the BigFix Server using the hostname or IP written in the masthead fields.
This setting does not exist by default. Changing it requires a Client restart to become effective.</td><td></td><td></td><td></td><td></td></tr>
<tr><td>_BESClient_RelaySelect_AlwaysOnIPListChange
If a client is temporarily on a subnetwork from where it cannot register to the desired relay,</td><td></td><td><table><tr><td>Default value</td><td>0 (false)</td></tr><tr><td>Setting type</td><td>Boolean</td></tr><tr><td>Value range</td><td>0 (false) - 1 (true)</td></tr></table></td><td>Client</td><td>9.2.8 and later</td></tr>
</table>

Name/Description	References	Values	Component(s) affected	Version(s) applicable
the client registers with the root server. By default an auto relay selection is not triggered even if, later on, the client IP situation changes and the client can communicate again with the desired relay. This means that the client will continue to be registered with the root server until a new explicit relay selection occurs. You can assign the val-				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
ue **1** to this configuration setting to change the default behavior and trigger automatically a new relay selection when the IPList addresses changes after the registration even if the IP used for the current registration is still available.				
_BESClient_RelaySelect_IntervalSeconds The BigFix Relay selection algorithm will run this often. If	What is the meaning of the message "skip-	Default value: 21,600 (6 hours) Setting type: Numeric (number of seconds)	Client, Relay	All

Name/Description	References	Values		Component(s) affected	Version(s) applicable
a closer BigFix Relay is found, the closer BigFix Relay will be used. **Note:** If the BigFix Relay selection is failing, this interval is ignored.	ping relay select. Registering with current relay" in BES Client Debug log?	Value range	600 - 31,536,000 (1 year)		
		Task available ?	Yes		
_BESClient_RelaySelect_MinRetryIntervalSeconds		Default value	60	Client	All
		Setting type	Numeric (number		

Name/Description	References	Values		Component(s) affected	Version(s) applicable
If the BigFix Relay selection algorithm fails (no BigFix Relays are found), the BigFix Client will try again after this many seconds. The BigFix Client will double this value on each successive retry that fails to locate a BigFix Relay.			of seconds)		
		Value range	0 - 31,536,000 (1 year)		
		Task available ?	Yes		
_BESClient_RelaySelect_MaxRetryIntervalSeconds After failing to find a BigFix Relay, the BigFix Client		Default value	7,200 (2 Hours)	Client	All
		Setting type	Numeric (Seconds)		

Name/Description	References	Values		Component(s) affected	Version(s) applicable
will continue to try to find a BigFix Relay. Each time it fails, the BigFix Client will double the time it spends until this maximum is exceeded. Then the BigFix Client will try with this maximum retry interval until it successfully selects a BigFix Relay.		Value range	0 - 31,536,000 (1 year)		
		Task available ?	Yes		
_BESClient_RelaySelect_MaximumTTLToPing The maximum number	Automatic relay selection in a high	Default value	20	Client, Relay	All
		Setting type	Numeric		

Name/Description	References	Values		Component(s) affected	Version(s) applicable
of network hops (Time To Live) a BigFix Client will search over before considering its automatic relay selection attempt a failure. A BigFix Client will be able to find only BigFix Relays that are less than the maximum number of network hops away from the BigFix Client. This setting can be used to limit the ICMP traffic generated by	latency and/or high hop network is failing	Value range	1 - 255		
		Task available ?	Yes		

<table>
<tr><th>Name/Description</th><th>References</th><th>Values</th><th>Component(s) affected</th><th>Version(s) applicable</th></tr>
<tr><td>the automatic relay selection algorithm.</td><td></td><td></td><td></td><td></td></tr>
<tr><td>_BESClient_RelaySelect_PingDelayMicroSeconds
During the automatic relay selection algorithm, the BigFix Client will wait this long between each iteration. On each iteration the BigFix Client sends ICMP packets with a TTL greater than the last iteration. This setting can be used to slow down</td><td>Automatic relay selection in a high latency and/or high hop network is failing</td><td><table><tr><td>Default value</td><td>200</td></tr><tr><td>Setting type</td><td>Numeric (microseconds)</td></tr><tr><td>Value range</td><td>0 - 1,000,000 (1 second)</td></tr><tr><td>Task available ?</td><td>No</td></tr></table></td><td>Client</td><td>All</td></tr>
</table>

<table>
<tr><th>Name/Description</th><th>References</th><th>Values</th><th>Component(s) affected</th><th>Version(s) applicable</th></tr>
<tr><td>the automatic relay selection algorithm and reduce the network traffic generated.</td><td></td><td></td><td></td><td></td></tr>
<tr><td>_BESClient_RelaySelect_ResistFailureIntervalSeconds
This setting controls the time interval that the BigFix Client will ignore communications errors before performing BigFix Relay selection. Once a BigFix Relay has been selected and the BigFix Client has</td><td>Configuring relay failover (on page 116)</td><td><table><tr><td>Default value</td><td>600 (10 minutes)</td></tr><tr><td>Setting type</td><td>Numeric (seconds)</td></tr><tr><td>Value range</td><td>0 - 21,600 (6 hours)</td></tr><tr><td>Task available ?</td><td>Yes</td></tr></table></td><td>Client</td><td>All</td></tr>
</table>

Name/Description	References	Values	Component(s) affected	Version(s) applicable
successfully registered, it will ignore errors when it attempts to post its results to the BigFix Relay or BigFix Server for this long before deciding to choose another BigFix Relay.				
_BESClient_RelaySelect_FailoverRelay This setting determines what the BigFix Client will do in the event that no BigFix Relays respond to TTL pings up to the max-	What manual Relay selection options do I have for my clients and Relays?	Default value: None Setting type: String (URL of the failover relay) Task available ? : No	Client	All

Name/Description	References	Values	Component(s) affected	Version(s) applicable
imum configured distance. In this event, the BigFix Client will attempt to register with the defined failover BigFix Relay before trying the BigFix Server. This setting was first introduced in BigFix 5.1. **Note:** This format is the same as for the _-_Relay-				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
Server1 and __RelayServer2 settings. For example: `http://servername:port/bfmir`				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
 `ro r /d ow nl oa ds /` .				
_BESClient_RelaySelect_FailoverRelayList This setting contains a list of failover relays to choose from when no relay listed as primary, secondary or specified in the tertiary list responded to pings. This setting,		Default value: None Setting type: String Value range: A semicolon-separated list Task available ?: No Platform: All Component restart required ?: No	Client	All

Name/Description	References	Values	Component(s) affected	Version(s) applicable
first introduced in BigFix 9.0, is a semi-colon delimited list of relays to try. For automatic relay selection, see Relay Affiliation. If specified, this setting overrides _BESClient_RelaySelect_FailoverRelay. (Example: relay1.company.com;192.168.123.32;relay2.company.com)				
__RelaySelect_Automatic		Default value: 0 (disabled)	Client	All

Name/Description	References	Values		Component(s) affected	Version(s) applicable
Use this setting to specify if the client is configured for automatic parent relay selection. Setting "__RelaySelect_Automatic = 1" means that the client uses the automatic parent relay selection. **Note:** Clients that are configured for automatic par-		Setting type	Boolean		
		Value range	1 (enabled), 0 (disabled)		
		Task available ?	Yes		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
ent relay selection cannot communicate through a proxy with their parent relay because they must be able to ping				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
 the relay.				
_BESClient_RelaySelect_TertiaryRelayList When using manual relay selection, this setting is a way to specify a list of failover relays to choose from when the primary and secondary relays are not reachable. This setting is a semi-colon delimited list of relays to try. Manual selection goes in this order,	What manual Relay selection options do I have for my clients and Relays?	Default value: None Setting type: String Value range: A semi-colon-separated list Task available ?: No	Client	All

Name/Description	References	Values	Component(s) affected	Version(s) applicable
primary/secondary/tertiary list/failover/root. For automatic relay selection, see Relay Affiliation. (Example: **relay1.company.com;192.168.123.32;relay2.company.com**)				

Top (on page 406)

Relay management

Name/Description	References	Values	Component(s) affected	Version(s) applicable
_BESClient_Relay_NameOverride	Why are my BigFix Clients	Default value: None	Relay	All

Name/Description	References	Values		Component(s) affected	Version(s) applicable
If the BigFix Relay is using a DNS name that is not accessible by all BigFix Clients, an override can be used. If this value is set on the BigFix Relay computer, it will return this setting for the dns name inspector, which will be reported to the BigFix Server and used by other BigFix Clients to locate the BigFix Relay. You can also use this value	not choosing their specified BigFix Relay?	Setting type	String (override name)		
		Value range	hostname OR ipaddress (do not include port number)		
		Task available ?	Yes		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
if you want your BigFix Clients to locate their relays by IP address, rather than DNS name lookup. For example, if the BigFix Relay is reported at relay1:52311, you can use this setting to make the BigFix Relay be accessible at relay1.company-.com:52311 or even 192.168.100.123:52311. **Note:** This value				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
 needs to be set on only the one BigFix Relay computer and not on all the BigFix Client computers.				
_BESClient_Relay_-Chain_Days The relay chain information is	Viewing the relay chain	Default value: 10 Setting type: Numeric (days)	Client	9.5.13 and later

Name/Description	References	Values		Component(s) affected	Version(s) applicable
stored in `.txt` files within a new client folder named `RelayChain` located in the BES Global directory. Use this setting to specify the number of days for which the files should be retained in the folder.	on the client	Value range	0 - 366		

Top (on page 406)

Affiliation

Name/Description	References	Values		Component(s) affected	Version(s) applicable
_BESRelay_Register_-Affiliation_AdvertisementList		Default value	None	Relay	All

Name/Description	References	Values		Component(s) affected	Version(s) applicable
This BigFix Client setting should also be set to a semi-colon (;) delimited list of relay affiliation groups. For example: AsiaPacific;DMZ;* **Note:** The labels, defined in the client setting `_-BESRelay_-Register_-`		Setting type	String (A semi-colon-separated list)		
		Task available ?	No		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
`Affiliation_-Advertisement-List` and delimited by semi-colon (;), must not be bigger than 64 characters.				

Name/Description	References	Values		Component(s) affected	Version(s) applicable
_BESClient_Register_-Affiliation_SeekList This BigFix Client setting should be set to a semi-colon (;) delimited list of relay affiliation groups. For example: AsiaPacific;Americas;DMZ		Default value	None	Client	All
		Setting type	String (A semi-colon-separated list)		
		Task available ?	No		

Top (on page 406)

Child Number Limit

Name/Description	References	Values		Component(s) affected	Version(s) applicable
Enterprise Server-ClientRegister_Max-ChildCount Maximum number of	Logics about automatic relay	Default value	0 (unlimited)	Server, Relay	All
		Setting type	Numeric		

Name/Description	References	Values		Component(s) affected	Version(s) applicable
clients the BigFix relay will allow to register. Once this limit is reached, connections will be rejected. This limit also includes the registered BigFix clients that are not running but that are in the registration list. The not running BigFix clients are removed from the registration list after 24 hours. **Note:** The set-	selection	Value range	0 - 4,294,967,295		
		Task available ?	No		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
 ting should be applied before the clients are registered.				
Enterprise Server-ClientRegister_Max-ChildRelayCount Maximum allowed number of BigFix clients that are BigFix relays registered to the parent. This limit also includes the registered BigFix re-	Logics about automatic relay selection	Default value: 0 (unlimited) Setting type: Numeric Value range: 0 - 4,294,967,295 Task available ?: No	Server, Relay	All

Name/Description	References	Values	Component(s) affected	Version(s) applicable
lays that are not running but that are in the registration list. The not running BigFix relays are removed from the registration list after 24 hours. This limit is calculated independently of the MaxChildCount setting. **Note:** The setting should be applied be-				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
 fore the re-lays are regis-tered.				

Top (on page 406)

Client notification

Command polling

Name/Description	References	Values	Component(s) affected	Version(s) applicable
_BESClient_Comm_CommandPollEnable When this configuration setting	Enabling client polling Changing the gather interval for a BigFix Client via the command polling client settings	Default value: 0 Setting type: Boolean Value range: 1 (True) 0 (False)	Client	9.2 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
is enabled, computers will poll their parent relay for UDP commands they may have missed. If you set it to '1', the BigFix		Task available? Yes		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
Client will check with its BigFix Relay for any new gatherers or refreshes when the number of seconds specified in "_-				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
BES-Client_-Comm_-CommandPoll-Interval-Seconds elapses.999999"				
_BESClient_Comm_CommandPollIntervalSeconds When 'BES-Client_-Comm_-Com-	Changing the gather interval for a BigFix Client via the command polling client settings	Default value: 900 Setting type: Numeric (seconds) Value range: 60 - 31,536,000	Client	9.2 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
mand-Poll-Enable = 1', this configuration setting determines how often the BigFix Client will check with its Big-		(1 Year) Task available? Yes		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
Fix Relay for any gatherers or refreshes. Avoid setting this to be less than 900 seconds due to performance implications				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
cations.				

Top (on page 406)

UDP Notification

Name/Description	References	Values	Component(s) affected	Version(s) applicable
_Enterprise Server_ClientRegister_BatchCount This setting controls the number of UDP pings the BigFix re-		Default value: 100 Setting type: String Value range: N/A Task available?: No Component restart required?: Yes	Server, Client, Relay	9.2 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
lay will send before delaying for a period of time. The length of the delay is controlled by _-Enterprise Serv-				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
er_-Client-Register-Batch-Delay. This setting could be used to limit the rate at which a BigFix relay sends out UDP				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
pings if this network traffic is harmful in some way.				
_Enterprise Server_Client-Register_-BatchDelay This setting controls how long the BigFix		Default value: 1,000 Setting type: Numeric (milliseconds) Value range: 1 - 4,294,967,295	Server, Client, Relay	9.2 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
relay will wait between sending out a batch of UDP pings to BigFix clients. This setting could be used to limit the rate at		Task available ? No Component restart required ? Yes		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
which a BigFix relay sends out UDP pings if this network traffic is harmful in some way.				
_Enterprise Server_ClientRegister_DisableChildUDPMessages		Default value: 0 Setting type: Boolean	Server, Client	9.2 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
A client setting that you can set on the BigFix server or a BigFix relay to control the UDP notifications		Value range: 0 - 1 Task available?: No Component restart required?: Yes		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
sent to the BigFix clients children. If set to 1, no UDP notifications are sent from that machine to the attached Big-				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
Fix Clients. In this case the BigFix Client subscribed to that parent relay must periodically poll the parent to look				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
for notifications and downloads. See also **_BESClient_Comm_CommandPoll_Enable** and **_BESClient_Com-**				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
m_-Command-Poll-Interval-Seconds client settings. This setting does not affect the behavior of the BigFix Client in-				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
stalled on the relay or on the server which continues to receive UDP messages.				
_BESClient_Comm_ListenEnable When set to 0 the		Default value: 1 (True) Setting type: Boolean	Client	9.2 and later

Name/Description	References	Values		Component(s) affected	Version(s) applicable
BigFix Client will not listen for UDP messages. If the BigFix Client does not receive UDP messages it will not receive gather		Value range	1 (True) - Listen for UDP 0 (False) - Ignore UDP		
		Task available	No ?		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
pings sent from the BigFix Server notifying the BigFix Client of new actions and content.				

Top (on page 406)

Persistent TCP connections

For detailed documentation on how to manage persistent TCP connections, see Persistent connections (on page 368).

Name/Description	References	Values	Component(s) affected	Version(s) applicable
_BESClient_PersistentConnection_Enabled Controls the ability to create a persistent connection on the client. Set the value for this	Persistent connections *(on page 368)*	Default value: 0 (False) Setting type: Boolean Value range: 0 - 1 Task available? No	Client	9.5.11 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
setting to 1 (True) if you want to enable the process for creating a persistent connection on the client. The TCP per-				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
sistent connection is established between a client and its parent relay.				
_BESRelay_PersistentConnection_Enabled Controls the ability	Persistent connections *(on page 368)*	Default value: 0 (False) Setting type: Boolean	Relay	9.5.11 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
to create a persistent connection on the relay. Set the value for this setting to 1 (True) if you want to en-		Value range: 0 - 1 Task available: No		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
able the process for creating a persistent connection on the relay. The TCP persistent connection is established				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
between a client and its parent relay.				
_BESRelay_PersistentConnection_NumberPerSubnet It defines the number of simultaneous per-	Persistent connections *(on page 368)*	Default value: 3 Setting type: Numeric Component restart required ?: Yes	Relay	9.5.11 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
sistent connections that the relay can establish for each subnet. From this value depends the fault-tolerance capa-				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
bility of the relay if connection failures occur in the subnet which affect the clients.				
_BESRelay_PersistentConnection_MaxNumber	Persistent connections *(on page 368)*	Default value • 100 (up to Patch 3)	Relay	9.5.11 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
It defines the maximum overall number of simultaneous persistent connections that the relay can estab-		• 1000 (from Patch 4) Setting type: Numeric Component restart required ? Yes		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
lish. This setting prevents that a relay keeps too many connections open.				
_BESRelay_PersistentConnection_KeepAliveSeconds It determines how of-	Persistent connections *(on page 368)*	Default value: 600 Setting type: Numeric (seconds) Component: Yes	Relay	9.5.11 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
ten the re- lay per- forms the health check of the es- tab- lished per- sis- tent con- nec- tions.		restart required ?		
_BESClient_PersistentConnection_SubnetCidrOverride It over-	Persistent connections *(on page 368)*	Default value: NA Setting type: valid CIDR format	Client	9.5.11 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
rides the subnet value detected by the client and has only effect in the context of the persistent connection.		(For example 192.168.1.0/24) Component restart required ? No **Note:** The setting takes effect after the following client registration.		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
This setting is useful within more complex networks where the value of the subnet detected by the client might prevent				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
persistent connections from working properly. The setting is effective only on clients having **_BES-Client_-Per-**				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
sistent-Connection_-Enabled set to 1.				
_BESRelay_PersistentConnection_-RetryIntervalSeconds When a persistent connection fails, it defines		Default value: 180 Setting type: Numeric (seconds) Component restart required ?: Yes	Relay	11.0.3 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
the amount of time (specified in seconds) a child relay will wait before attempting to reestablish a new con-				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
nection.				
_BESRelay_PersistentConnection_RetryLimit When a persistent connection fails, it defines the number of times a child relay will		Default value: 4 Setting type: Numeric Component restart required ?: Yes	Relay	11.0.3 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
attempt to re-establish a new connection.				
_BESRelay_PersistentConnection_OpenParent Controls the ability to create a persis-		Default value: 0 (Disabled) Setting type: Boolean Component restart required ?: Yes	Relay	11.0.3 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
tent connection with a parent relay. Set the value for this setting to 1 (True), if you want to enable the process for				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
creating a persistent connection on the relay. The TCP persistent connection is established between a child				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
relay and its parent relay.				

Top (on page 406)

Communication

Name/Description	References	Values	Component(s) affected	Version(s) applicable
_BESClient_Comm_SkipInternetActiveTest Before the BigFix Client tries to gather it will check the Windows API connection test to determine if an internet connection is available. This		Default value: 0 (False) Setting type: Boolean Value range: 1 (True) - ignore connection test 0 (False) - perform	Client	9.2 and later

<table>
<tr><th>Name/Description</th><th>References</th><th>Values</th><th>Component(s) affected</th><th>Version(s) applicable</th></tr>
<tr><td>setting will allow the BigFix Client to ignore the results of the connection test and attempt to gather anyway.</td><td></td><td><table><tr><td></td><td>connection test</td></tr><tr><td>Task available ?</td><td>No</td></tr></table></td><td></td><td></td></tr>
<tr><td>_BESClient_Comm_IPCommunicationsMode
This setting determines the network topology preference. Available values for this settings are: "Ipv4ThenIpv6", "Ipv6ThenIpv4", "OnlyIpv4". This setting was first added in BigFix 9.0.
Note: PeerNest</td><td></td><td><table><tr><td>Default value</td><td>Ipv4ThenIpv6</td></tr><tr><td>Setting type</td><td>String</td></tr><tr><td>Value range</td><td>Ipv4ThenIpv6, Ipv6ThenIpv4, OnlyIpv4</td></tr><tr><td>Task available ?</td><td>No</td></tr><tr><td>Component restart required ?</td><td>No</td></tr></table></td><td>Client</td><td>9.2 and later</td></tr>
</table>

Name/Description	References	Values	Component(s) affected	Version(s) applicable
is not affected by the address family order and it uses IPv6 if available, otherwise it uses IPv4. If you want to force it on IPv4, then you must use the "Only-Ipv4"				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
setting value.				

Top (on page 406)

Authentication

Name/Description	References	Values	Component(s) affected	Version(s) applicable
_BESRelay_Comm_KeyExchangePassword If an agent does not have a certificate and can on-	Manual key exchange *(on page 1212)*	Default value: None Setting type: String Component restart required ?: No	Relay	9.2 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
ly reach an authenticating relay on the network, connected through the internet, you can enable the key exchange				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
with an authenticating relay. The client includes the password in its key exchange with the authenticating relay,				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
which verifies it before forwarding the key exchange to its parent.				
_BESRelay_Comm_Authenticating Set to 1 to configure	Authenticating relays (on page 1210) Setting up internet relays Manually deploying a new BigFix Virtual Relay instance	Default value: 0 Setting type: Boolean Value range: 0 (disabled)	Relay	9.2 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
an authenticating relay or use the related task in the BES Support site. To configure an open relay, set		1 (enabled) Component restart required ? Yes		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
the parameter to `0` or use the related task in the BES Support site.				
_BESRelay_Comm_ClientCertUpdatePassword If an agent		Default value: None Setting type: String Component: No	Relay	10 Patch 7

Name/Description	References	Values	Component(s) affected	Version(s) applicable
has an expired short-term certificate and can only reach an authenticating relay on the network, connected through		nent restart required ?		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
the internet, you can manually update the certificate with an authenticating relay. The client includes the				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
password in its update certificate request with the authenticating relay, which verifies it against this setting val-				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
ue before forwarding the certificate update request to its parent.				

Top (on page 406)

Download

Name/Description	References	Values		Component(s) affected	Version(s) applicable
_BESClient_Download_-SizeLimitMB	Managing Down-	Default value	max unsigned	Client	9.5.9 and later

Name/Description	References	Values		Component(s) affected	Version(s) applicable
Use this setting to specify the size in MB that must not be exceeded by the total size of the downloads associated to an action. If the total size of the downloads associated with the action exceeds the specified value, the action is prevented from running. This check is done by the target agent for each action requiring the download of one or more than one files. Condition `download <url>`:	loads *(on page 1099)*		int32 (4,294,967,295)		
		Setting type	Numeric (MB)		
		Value range	0 - 4,294,967,295		
		Task available ?	No		
		Component restart required ?	No		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
Not being predictable the size of the download, it is considered by default the maximum value (4,294,967,295). Therefore, if the connected Relay is not a preferred Relay in the download condition mentioned above, the action is prevented from running. Condition `download now <url>`: Does not use the Relay Hierarchy, the check will not be executed. It is the same condition as when the set-				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
ting is not defined. In case of a dynamic download, for each action, the check is done on the total size of the files downloaded in the prefetch passes. When the threshold is reached, the download stops.				
_BESClient_Download_-UntrustedSites It is a boolean setting. When it is set to 0 (default value), the downloads from untrusted sites are not allowed. When it is set to 1, the downloads from un-	Customizing HTTPS for downloads *(on page 129)*	Default value: 0 Setting type: Boolean Value range: 0-1 Restart Required: No	Client	10 Patch 8

Name/Description	References	Values	Component(s) affected	Version(s) applicable
trusted sites are allowed. **Note:** This client setting will be taken into account only if `_BES-Client_Download_-Direct` is set to 1.				
_BESClient_Download_-CaCertPath It is a string setting where you can set the full path of a custom CA bundle.	Customizing HTTPS for downloads *(on*	Default value: None Setting type: String Value range: None	Client	10 Patch 8

Name/Description	References	Values		Component(s) affected	Version(s) applicable
It overrides the default one. **Note:** This client setting will be taken into account only if `_BES-Client_Download_-Direct` is set to 1. Deprecated from BigFix Version 10.0.9	*page 129)*	Restart Required	No		
_BESClient_Download_-Direct Normally, BigFix Clients will con-	Managing Downloads *(on*	Default value Setting type	0 (False) Boolean	Client	All

Name/Description	References	Values		Component(s) affected	Version(s) applicable
tact their BigFix Relay to receive downloads from the internet. This setting forces the BigFix Client to download files directly from the internet using the URL specified. This setting takes precedence over other download settings.	*page 1099)*	Value range	1 (True) - download from the Internet; 0 (False) - download normally		
		Task available ?	No		
_BESClient_Download_-DirectOnFail When the BigFix Client fails to download a file from the BigFix Server or BigFix Relay, it can attempt to download the file di-	Managing Downloads *(on page 1099)*	Default value	0 (False)	Client	All
		Setting type	Boolean		
		Value range	1 (True) - download from the Internet on failure 0		

Name/Description	References	Values		Component(s) affected	Version(s) applicable
rectly from the Internet. **Note:** If there is a proxy in your network that blocks unauthenticated access to the Internet, turning this setting to 1 (True) may not work.			(False) - do not download from Internet on failure		
		Task available ?	No		
_BESClient_Download_-Direct_Domainlist This setting is a semi-colon (;) delimited list	Managing Downloads *(on*	Default value	None	Client	10 Patch 1
		Setting type	String		

Name/Description	References	Values		Component(s) affected	Version(s) applicable
of domains for which direct download is desired. You can use the "*" as a wildcard. (Example: my.company.com;192.168.123.*;microsoft.com;*.adobe.com)	*page 1099)*	Value range	A semicolon-separated list		
		Task available ?	No		
		Component restart required ?	No		
_BESClient_Download_DirectRecovery When the BigFix Client fails to download a file from the Internet, it can attempt to download the file from the BigFix Server or BigFix Relay.	Managing Downloads *(on page 1099)*	Default value	0 (False)	Client	10 Patch 1
		Setting type	Boolean		
		Value range	0 (False) - Do not download from the BigFix Server or Big-		

Name/Description	References	Values		Component(s) affected	Version(s) applicable
			Fix Relay on failure. 1 (True) - Download from the BigFix Server or BigFix Relay on failure.		
		Task available ?	No		
		Component restart required ?	No		
_BESClient_Download_-ResetOnRelaySwitch If set to 1, this setting enables the download in-	Managing Downloads *(on*	Default value	0 (false)	Client	10 Patch 7
		Setting type	Boolean		

Name/Description	References	Values		Component(s) affected	Version(s) applicable
terruption on the Relay switch. It means that when the BigFix Client registers to a new Relay, the download in progress from the old Relay will be stopped and is restarted from the new Relay.	*page 1099)*	Value range	0,1		
		Restart Required	No		
_BESClient_Download_-Direct_SubnetList This setting is a semi-colon (;) delimited list of subnets in CIDR format and allows you to specify the list of subnets that allow the Direct Download. The setting accepts only subnets specified	Managing Downloads *(on page 1099)*	Default value	None	Client	10 Patch 7
		Setting type	String		
		Value range	A semi-colon-separated list		
		Restart Required	No		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
in CIDR format, for example: `192.1.77.0/25;192.1.0.0/16` If the Client connected to the Relay is part of one of this subnets, the direct download is allowed.				
_BESClient_Download_MinimumDiskFreeMB This setting stops both stages of downloading (normal stage and precaching stage) if the free space of the disk on which the client stores downloads is less than the value of this setting.	Managing Downloads *(on page 1099)*	Default value: 20 Setting type: Numeric (in MB) Value range: 0 - 4,294,967,295 Task available ?: No	Client	All

Name/Description	References	Values	Component(s) affected	Version(s) applicable
_BESClient_Download_-NormalStageDiskLimit-MB This setting stops normal stage downloading if the client is already using this much normal stage download disk space. Actions marked for normal downloads will report constrained if the total space used for downloads exceeds this limit. **Note:** Normal stage downloads may ex-	Managing Downloads (on page 1099) Job-Disk-Limited message in the BigFix Client log	Default value: 2048 Setting type: Numeric (in MB) Value range: 0 - 4,294,967,295 Task available ?: No	Client	All

Name/Description	References	Values	Component(s) affected	Version(s) applicable
 ceed this limit by borrowing some space from the pre-cache stage space if it is not full.				
_BESClient_Download_-PreCacheStageContinue-WhenDiskLimited This setting indicates whether files that are pre-cached for a group action can be removed on the Client system to allow the Agent to continue with action download	Enabling data pre-cache *(on page 1106)*	Default value: 0 (Disabled) Setting type: Boolean Value range: 0 (Disabled) or 1 (Enabled) Task available ? : No	Client	9.5.10 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
requests for subsequent sub actions in the group. Files that are removed from the pre-cache area on the Client have to be downloaded again at run time from the parent relay, if required by a sub action.				
_BESClient_Download_-PreCacheStageDiskLimitMB This setting stops pre-cache stage downloading if the client is already using this much pre-cache stage download disk space. Actions marked for	Enabling data pre-cache (on page 1106) Job-Disk-Limited message	Default value: 250 Setting type: Numeric (in MB) Value range: 0 - 4,294,967,295 Task available ?: No	Client	All

Name/Description	References	Values	Component(s) affected	Version(s) applicable
pre-caching will report constrained if the total space used for downloads exceeds this limit.	in the BigFix Client log			
_BESClient_Download_-PreCacheStageEnabled This setting determines if pre-cache download functionality is enabled. Set this to zero (0) to disable pre-caching of downloads.		Default value: 1 (Enabled) Setting type: Boolean Value range: 1 (Enabled), 0 (Disabled) Task available ? : No	Client	All
_BESClient_Download_-CheckAvailabilitySeconds This setting specifies the amount of time, in seconds, af-	Controlling UDP traffic from BigFix Server	Default value: 600 Setting type: Numeric (seconds)	Client	All

Name/Description	References	Values		Component(s) affected	Version(s) applicable
ter which the client will retry the download of a file from the Relay. It configures the client to poll for downloads, it can be useful in case the UDP is not working.	and Relays	Value range	60-3600		
		Task available ?	No		
_BESClient_Download_RetryMinutes When the BigFix Client fails to download a file from the BigFix Server or a BigFix Relay during an action, it will wait for the specified amount of time then try again. After each failed attempt, it will double the pre-	BigFix Troubleshooting Downloads	Default value	1	Client	All
		Setting type	Numeric (minutes)		
		Value range	0 -10 (minutes)		
		Task available ?	No		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
vious retry interval used until is reached an upper bound limit of 480 minutes (8 hours) hard-coded. **Note:** The RetryMinutes value means: another download attempt does not start for a period of x minutes; when the attempt actually occurs				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
might vary depending on the client activities and load.				
_BESClient_Download_-RetryLimit When the BigFix Client fails to download a file from the BigFix Server or a BigFix Relay during an action, it will retry the specified number of times. The retry interval is managed by the _BESClient_Download_-RetryMinutes setting.	BigFix Troubleshooting Downloads	Default value: 6 Setting type: Numeric (retry limit) Value range: 0 - 100 Task available ?: No	Client	All

Name/Description	References	Values		Component(s) affected	Version(s) applicable
_BESClient_Download_-ChannelThreshold The BigFix Client can simultaneously download two files at a time by using one "main channel" and one "thin channel" when the BigFix Client is attached to the BES Root Server. The main channel is used for all downloads, but if the main channel is currently downloading a large file, the thin channel can be used to download smaller files if the download size is less than		Default value	500,000	Client	All
		Setting type	Numeric (bytes)		
		Value range	1 - 100,000,000 (Set to 0 to disable the thin channel)		
		Task available ?	No		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
the specified threshold. If this setting is set high, the BigFix Client will use the thin channel to download larger files, which could slow down actions because two large files may be downloading at the same time (each using half the bandwidth) instead of one file after the other. If this setting is set low, the thin channel will be used for only very small file downloads.				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
_BESClient_Download_InactivityTimeout When the BigFix Client is downloading a file from the BigFix Server or BigFix Relay, it sometimes will experience some inactivity because of network problems, disconnections, etc. When there is inactivity, the BigFix Client will wait for the specified number of seconds to receive more data before aborting the download.	BigFix Troubleshooting Downloads	Default value: 300 (5 minutes) Setting type: Numeric (seconds) Value range: 0 - 4,294,967,295 Task available ? : No	Client	All
_BESClient_Download_DownloadsCacheLimitMB		Default value: 20	Client	All

Name/Description	References	Values		Component(s) affected	Version(s) applicable
This configuration setting sets the BigFix client download cache limit. The BigFix client keeps all files that were cached in the same day, regardless of the download cache limit. If there is only one file which is larger than the configured download cache size, the BigFix client keeps this file, regardless of the age or the download cache limit.		Setting type Value range Task available ?	Numeric (in MB) 0 - 4,294,967,295 No		
_BESClient_Download_UtilitiesCacheLimitMB This configuration setting sets		Default value Setting type	10 Numeric (in MB)	Client	All

Name/Description	References	Values		Component(s) affected	Version(s) applicable
the Utility cache limit.		Value range	0 - 4,294,967,295		
		Task available ?	No		
_BESRelay_Downloads_-OlderThanInDays This setting specifies which rows listing the downloaded files must be deleted by the cleanup task in the Mirror Server/Config/DownloadState.db database on the Relay and on the Root server. The rows representing files older than the value specified in this setting	Download Status Report Cleanup	Default value	60	Root server, Relay	All
		Setting type	Numeric (days)		
		Value range	0 - 2,147,483,647		
		Task available ?	No		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
are deleted from that database. **Note:** The information about the downloaded files is visible at: `http://server_host:52311/cgi-bin/bfenterprise/BESMirrorRequest.exe`.				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
_BESRelay_Download_-UntrustedSites It is a boolean setting. When it is set to 0 (default value), the downloads from untrusted sites are not allowed. When it is set to 1, the downloads from untrusted sites are allowed.	Customizing HTTPS for downloads *(on page 129)*	Default value: 0 Setting type: Boolean Value range: 0-1 Restart Required: No	Server/Relay	10 Patch 8
_BESRelay_Download_-CaCertPath It is a string setting where you can set the full path of a custom CA bundle. It overrides the default one. Deprecated from BigFix Version 10.0.9	Customizing HTTPS for downloads *(on page 129)*	Default value: None Setting type: String Value range: None Restart Required: No	Server/Relay	10 Patch 8

Name/Description	References	Values		Component(s) affected	Version(s) applicable
_BESClient_Download_-FastHashVerify When this setting is enabled, the client increases the CPU consumption during the hash verification of the file. If the pre-cache option is selected, this acceleration will not be activated. The values used for the CPU increase are set in _BESClient_-Resource_Work-FastHashVerify and _BESClient_-Resource_Sleep-FastHashVerify.		Default value	0 (False)	Client	10 Patch 2
		Setting type	Boolean		
		Value range	0 (False) - Use default settings for CPU consumption. 1 (True) - Switch to fast verify scenario.		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
By default, the setting is disabled.				
_BESRelay_Download_-CaCertDirectory It is a string setting where you can set the full path of a folder where the user stores its custom CA bundles. All the `.crt` and `.pem` files contained in this folder are added to the default certificates.	Customizing HTTPS for downloads *(on page 129)*	Default value: None Setting type: String Value range: None Restart Required: No	Server/Relay	10 Patch 9
_BESClient_Download_-CaCertDirectory It is a string setting where you can set the full path of a folder where the user stores its custom CA bundles.	Customizing HTTPS for downloads *(on*	Default value: None Setting type: String Value range: None	Client	10 Patch 9

Name/Description	References	Values		Component(s) affected	Version(s) applicable
All the `.crt` and `.pem` files contained in this folder are added to the default certificates. **Note:** This client setting will be taken into account only if `_BESClient_Download_Direct` is set to 1.	*page 129)*	Restart Required	No		
_BESRelay_Download_IgnoreDefaultCaCerts It is a boolean setting with two	Customizing HTTPS	Default value	0	Server/Relay	10 Patch 9

Name/Description	References	Values		Component(s) affected	Version(s) applicable
possible values: 0 and 1. When set to 1, it tells the component to ignore the default CA bundle file, contained in BES Support. The default value is 0.	for downloads *(on page 129)*	Setting type	Boolean		
		Value range	0-1		
		Restart Required	No		
_BESClient_Download_-IgnoreDefaultCaCerts It is a boolean setting with two possible values: 0 and 1. When set to 1, it tells the component to ignore the default CA bundle file, contained in BES Support. The default value is 0. **Note:** This	Customizing HTTPS for downloads *(on page 129)*	Default value	0	Client	10 Patch 9
		Setting type	Boolean		
		Value range	0-1		
		Restart Required	No		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
client setting will be taken into account only if `_BES-Client_Download_-Direct` is set to 1.				
_BESRelay_Download_-ActiveDownloadsMax-SizeMB This setting is available on the Relay. If the value is > 0, it represents the maximum content size, specified in MB, that the ActiveDownloads folder can con-	Managing Downloads *(on page 1099)*	Default value: None Setting type: Numeric (MB) Value range: 0-4,294,967,295 Restart Required: Yes	Relay	Version 11

Name/Description	References	Values	Component(s) affected	Version(s) applicable
tain. If the content that the Relay is going to download exceeds the ActiveDownloads free space, according to this setting, the Relay does not download the file and the Client reports the "No Space In Active Downloads" action status. The download automatically resumes if the ActiveDownloads free space increases or the setting is modified. **Note:** Mini-				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
mum valid value is 256 MB. Values lower than 256 MB will de defaulted to 256.				
_BESRelay_Download_-SuspendExpiredAction This setting is available on the Relay. If the value is set to 1, the downloads for the expired actions are suspended. If more than one action is required to download the same file, the Relay suspends the down-	Managing Downloads *(on page 1099)*	Default value: 0 Setting type: Boolean Value range: 0-1 Restart Required: Yes	Relay	11 Patch 5

Name/Description	References	Values	Component(s) affected	Version(s) applicable
load only if all these actions expire. If a new action requires to download a suspended file, the Relay resumes the suspended download, avoiding to restart from the beginning.				

Top (on page 406)

Reports management

Name/Description	References	Values	Component(s) affected	Version(s) applicable
_BESClient_Report_-MinimumInterval When the BigFix Clients are performing an action, they will		Default value: 60 Setting type: Numeric (in seconds)	Client	All

Name/Description	References	Values		Component(s) affected	Version(s) applicable
send reports of the status of the action (Evaluating, Waiting, Running, Fixed, etc.). This setting controls the minimum number seconds between sending reports. If this number is too high, the BigFix Clients will take a long time update their action status. If this number is too low, it will put extra load on the BigFix Server to process the extra ac-		Value range	0 - 86,400		
		Task available ?	No		

<table>
<tr><th>Name/Description</th><th>References</th><th>Values</th><th>Component(s) affected</th><th>Version(s) applicable</th></tr>
<tr><td>tion status reports.</td><td></td><td></td><td></td><td></td></tr>
<tr><td>_BESRelay_PostResults_ResultTimeLimit
The BigFix relay accumulates reports from the BigFix clients and then compresses them and sends them to the parent BigFix relay or BigFix server after waiting a specified amount of time. This setting sets the amount of time to wait before sending the reports. The</td><td>BigFix Client report interval vs BigFix Relay post</td><td><table><tr><td>Default value</td><td>3</td></tr><tr><td>Setting type</td><td>Numeric (seconds)</td></tr><tr><td>Value range</td><td>1 - 4,294,967,295</td></tr><tr><td>Task available ?</td><td>No</td></tr></table></td><td>Relay</td><td>All</td></tr>
</table>

Name/Description	References	Values	Component(s) affected	Version(s) applicable
longer the wait, the more latency before the BigFix server receives the reports from the BigFix clients, but the greater the efficiency of the system (lower network traffic).				
_BESRelay_PostResults_RawCarbonCopyPath If specified, this setting defines the path on the relay where a copy of the BigFix reports must be saved for		Default value: None Setting type: String (path)	Relay	All

Name/Description	References	Values	Component(s) affected	Version(s) applicable
troubleshooting purposes. By default no copy is saved.				
_BESRelay_PostResults_QueryRawCarbonCopyPath If specified, this setting defines the path on the relay where a copy of the BigFix Query reports must be saved for troubleshooting purposes. By default no copy is saved.		Default value: None Setting type: String (path)	Relay	9.5.2 and later
_BESRelay_PostResults_ResultSizeLimit This is the maximum size in bytes		Default value: 1024*1024	Relay	All

<table>
<tr><th>Name/Description</th><th>References</th><th>Values</th><th>Component(s) affected</th><th>Version(s) applicable</th></tr>
<tr><td>of inputs to an archive that a relay submits to its parent.
No more files will be added to the archive once this limit has been exceeded (although the last file added cab push the size over the limit).

Note:

• The archive of uncompressed and un-</td><td></td><td><table><tr><td>Setting type</td><td>Numeric (size in bytes)</td></tr><tr><td>Component restart required ?</td><td>Yes</td></tr></table></td><td></td><td></td></tr>
</table>

Name/Description	References	Values	Component(s) affected	Version(s) applicable
 encrypted reports are themselves compressed, so the output of the archiving process may be smaller than ResultSizeLim-				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
 it even when the limit gets hit. • The Relay does *not* immediately start posting the next report after run-				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
ning into this limit, so this setting can serve as a throughput limiter on the relay.				

Top (on page 406)

Query

Name/Description	References	Values	Component(s) affected	Version(s) applicable
_BESClient_Query_MOMaxQueryTime This setting determines how long the QnA can process a query submitted by a Master Operator before stopping, discarding the request and responding to the agent that the request could not be completed because the maximum processing time has elapsed.	Getting client information by using BigFix Query (on page 203) BigFix Query	Default value: 60 Setting type: Numeric (seconds) Value range: 1 - 3,600	Client	9.5.2 and later
_BESClient_Query_NMOMaxQueryTime	Getting client	Default value: 10	Client	9.5.2 and later

Name/Description	References	Values		Component(s) affected	Version(s) applicable
This setting determines how long the QnA can process a query submitted by a Non Master Operator before stopping, discarding the request and responding to the agent that the request could not be completed because the maximum processing time has elapsed.	information by using BigFix Query (on page 203) BigFix Query	Setting type	Numeric (seconds)		
		Value range	1 - 3,600		
_BESClient_Query_Idle-Timeout This setting determines how long the	Getting client information by	Default value	600	Client	9.5.2 and later
		Setting type	Numeric (seconds)		

Name/Description	References	Values		Component(s) affected	Version(s) applicable
QnA will wait for new reports to be processed before stopping. The QnA will start automatically when a new query to process is received by the agent.	using BigFix Query *(on page 203)*	Value range	60 - 3,600		
_BESRelay_Query_RemovalTask This setting determines how often the BigFix Query requests cleanup task runs to free up space in the cache on the relay.	Getting client information by using BigFix Query *(on page 203)*	Default value Setting type	10 Numeric (minutes)	Relay	9.5.2 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
_BESRelay_Query_MinTime This setting determines the criteria used by the cleanup task for removing BigFix Query requests. When the cleanup task runs, it removes the requests whose age is older or equal to the value specified in the setting.	Getting client information by using BigFix Query *(on page 203)*	Default value: 60 Setting type: Numeric (minutes)	Relay	9.5.2 and later
_BESRelay_Query_MemoryLimit This setting specifies the maximum size of the cache that can be used on	Getting client information by using BigFix	Default value: 100 Setting type: Numeric (MB)	Relay	9.5.2 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
the relay to store the BigFix Query requests. When the specified threshold is exceeded, the older entries are removed until the threshold is satisfied again.	Query *(on page 203)*			
_BESRelay_Query_ResultsMemoryLimit This setting specifies the maximum size of the cache that can be used on the relay to store the BigFix Query results. When the specified threshold ex-	Getting client information by using BigFix Query *(on page 203)*	Default value: 100 Setting type: Numeric (MB)	Relay	9.5.2 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
ceeds the older entries, regardless of their age, are removed until the threshold is satisfied again.				

Top (on page 406)

Application Usage Collection

Name/Description	References	Values	Component(s) affected	Version(s) applicable
_BESClient_UsageManager_EnableAppUsage If you set this configuration setting to 1, you enable the application usage tracking on		Default value: 0 (false) Setting type: Boolean Value range: 0 or 1 Platform: All	Client	9.2 and later

Name/Description	References	Values		Component(s) affected	Version(s) applicable
the Client system. By doing so you allow application usage tracking inspectors to report the time intervals when the instances of the applications are running.		Component restart required ?	No		
_BESClient_UsageManager_EnableAppUsageSummaryApps In this configuration setting you can specify the list of names of the applications that the agent should collect usage summary information about. Use the		Default value Setting type Value range Platform Component restart required ?	Null String Null All No	Client	9.2 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
':' character as separator, for example: "+:app1:app2" means to track app1 and app2 "-:app1:app2" means to track all applications but app1 and app2 This setting requires the **_BESClient_UsageManager_EnableAppUsageSummary** to be set to 1.				
_BESClient_UsageManager_EnableAppUsageSummary Set this configuration set-	Application usage summary	Default value: 0 (false) Setting type: Boolean	Client	9.2 and later

Name/Description	References	Values		Component(s) affected	Version(s) applicable
ting to 1 to enable collecting application usage summary data on the Client system.		Value range	0 or 1		
		Platform	All		
		Component restart required ?	No		
_BESClient_UsageManager_EnableAppUsageSummaryPath This setting is used to determine whether or not the Client must collect and save the full path information of the executable of an application. You can retrieve this information by running the appli-		Default value	0 (false)	Client	9.5.5 and later
		Setting type	Boolean		
		Value range	0 or 1		
		Platform	All platforms except AIX		
		Component restart required ?	No		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
cation usage summary inspector on the Client system. When this setting is turned on (value 1), the Agent collects and saves the executable full path data in the UsageData stat files. When the setting is turned off (value 0), the saved full path data is deleted and the used disk space is released. If **_BESClient_UsageManager_EnableAppUsage-**				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
SummaryPath is set to 0, the application usage summary inspector returns "nonexistent object" errors. This setting requires the **_BESClient_UsageManager_EnableAppUsageSummary** to be set to 1.				

Top (on page 406)

Bandwidth throttling

Set on the BigFix Server or on the BigFix Relays to control how to download files during the gathering phase.

Name/Description	References	Values		Component(s) affected	Version(s) applicable
_BESGather_Download_LimitBytesPerSecond BigFix Gather can "throttle" the download speed by limiting the amount of bytes it downloads per second. This setting is especially effective if network bandwidth is a concern (such as with a dial-up connection). When this setting is used on the BigFix Relay, the BigFix Relay will limit its download speed when	Bandwidth throttling *(on page 1090)*	Default value	0 (disabled)	Server, Relay	All
		Setting type	Numeric (bytes)		
		Value range	0 - 4,294,967,295 (Set to 0 to disable download throttling and use the maximum possible bandwidth)		
		Component restart required ?	Yes		
		Task available ?	Yes		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
downloading from the BigFix Server or parent BigFix Relay. This setting throttles all downloading traffic.				
_BESGather_Download_ThrottleGroup The parent considers this Relay to be part of the group that is specified here.	Bandwidth throttling (on page 1090) Static throttling (on page 1097)	Default value: An empty string Setting type: String Platform: Windows	Relay	All
_BESClient_Download_ThrottleGroup The parent considers this client to be	Bandwidth throttling (on	Default value: An empty string Setting type: String	Client	All

<table>
<tr><th>Name/Description</th><th>References</th><th>Values</th><th>Component(s) affected</th><th>Version(s) applicable</th></tr>
<tr><td>part of the group that is specified here.</td><td>page 1090)
Static throttling (on page 1097)</td><td></td><td></td><td></td></tr>
<tr><td>_BESRelay_HTTPServer_ThrottleKBPS
'Global' throttling on outgoing data to BigFix clients from the BigFix relay/server (0 means no limit). If its value is 1000 and there are 100 clients downloading simultaneously, the BigFix relay/server will send data to each</td><td>Bandwidth throttling (on page 1090)</td><td><table><tr><td>Default value</td><td>0 (disabled)</td></tr><tr><td>Setting type</td><td>Numeric (KBPS)</td></tr><tr><td>Value range</td><td>0 - 4,294,967,295</td></tr><tr><td>Task available ?</td><td>Yes</td></tr><tr><td>Component restart required ?</td><td>Yes</td></tr></table></td><td>Server, Relay</td><td>All</td></tr>
</table>

Name/Description	References	Values	Component(s) affected	Version(s) applicable
BigFix client at 10KBPS (for a total of 1000KBPS). Throttling is implemented by sending a chunk of data at full speed, and then waiting chunk-size/KBPS seconds before sending the next chunk so if all requests are smaller than the chunk size, then there may be no throttling at all. Uploaded data to other BigFix relays/server is not throttled				

<table>
<tr><th>Name/Description</th><th>References</th><th>Values</th><th>Component(s) affected</th><th>Version(s) applicable</th></tr>
<tr><td>by this setting (PostResults data).</td><td></td><td></td><td></td><td></td></tr>
<tr><td>_BESClient_Download_LimitBytesPerSecond
The BigFix Client can "throttle" the download speed by limiting the amount of bytes it downloads per second. This setting is especially effective if network bandwidth is a concern (such as with a dial-up connection). If the BigFix Server or BigFix Relay is al-</td><td>Bandwidth throttling (on page 1090)</td><td><table><tr><td>Default value</td><td>0 (disabled)</td></tr><tr><td>Setting type</td><td>Numeric (bytes/second)</td></tr><tr><td>Value range</td><td>0 - 4,294,967,295 (Set to 0 to disable download throttling)</td></tr><tr><td>Task available ?</td><td>Yes</td></tr><tr><td>Component restart required ?</td><td>No</td></tr></table></td><td>Client</td><td>All</td></tr>
</table>

<table>
<tr><th>Name/Description</th><th>References</th><th>Values</th><th>Component(s) affected</th><th>Version(s) applicable</th></tr>
<tr><td>so set to use throttling, the lower of the two values will be used as the download speed.</td><td></td><td></td><td></td><td></td></tr>
<tr><td>_WebReports_-HTTPServer_Throttle-KBPS
Server-side static throttling settings control the total amount of download traffic that a server will send out to clients using static throttling. This setting denotes the amount of bandwidth allocated to any given write</td><td>Bandwidth throttling (on page 1090)</td><td><table><tr><td>Default value</td><td>0 (disabled)</td></tr><tr><td>Setting type</td><td>Numeric (KBPS)</td></tr><tr><td>Value range</td><td>0 - 4,294,967,295</td></tr><tr><td>Task available ?</td><td>Yes</td></tr><tr><td>Component restart required ?</td><td>Yes</td></tr></table></td><td>Web Reports</td><td>All</td></tr>
</table>

Name/Description	References	Values	Component(s) affected	Version(s) applicable
connection divided by the number of active write connections.				

Top (on page 406)

Airgap Mode

Name/Description	References	Values	Component(s) affected	Version(s) applicable
_BESServer_Airgap-Mode_Enabled With this setting, you specify if your server will access the Internet for updating the license and gathering the sites or not. This setting does not have an impact on		Default value: 0 Setting type: Boolean Value range: 0 (disabled) and 1 (enabled) Component restart required ?: Yes	Server	9.5.11 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
the actions containing prefetch or download instructions. Set to 1 to prevent your server from accessing the Internet for updating its license and gathering new sites. In this case, the log files will not display any errors. Set to 0 to allow your server to connect to the Internet to verify if some license updates are available and if new sites				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
versions can be gathered. In this case, if the server does not have Internet connectivity, the log files will display some errors.				

Top (on page 406)

HTTP server

Name/Description	References	Values	Component(s) affected	Version(s) applicable
_BESRelay_HTTPServer_DisableMetrics Enable or disable the metrics exposure on the HTTP Server of the BigFix Relay.		Default value: 0 Setting type: Boolean Value range: 0 - 1	Relay	11 Patch 4 and later

Name/Description	References	Values		Component(s) affected	Version(s) applicable
When the Relay authentication is enabled, then, the access to the metrics exposure is disabled if an invalid certificate is provided, even if the setting is set to 0 (default value).		Task available ?	No		
_BESRelay_HTTPServer_MaxConnections The maximum number of connections the Relay or RootServer HTTP Server can handle. It can be used as throttling parameter.		Default value Setting type Value range Task available ?	2,048 Numeric (number of threads) 1 - 65,536 No	Relay, Root Server	All

Name/Description	References	Values		Component(s) affected	Version(s) applicable
_HTTPServer_Referrer_CheckEnabled Enable or disable the check on an optional header of the HTTP protocol. It identifies the address of the web page (that is the URI or IRI) that linked to the resource being requested. If the referrer check is enabled, BigFix checks if the referrer does not exist or, if it exists, if it contains the hostname that originated the request. If nei-	Assumptions and requirements *(on page 99)*	Default value	0 (disabled)	Relay, Root server, Web Reports	9.5.2 and later
		Value range	0 (disabled) 1 (enabled)		
		Setting type	Boolean		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
ther of these two conditions is satisfied, the requested is rejected due to an "Invalid Referrer". If SAML is enabled, BigFix automatically does not check the referrer when running Web Reports because, otherwise, the requests coming from the SAML backend would be rejected. For fresh BigFix installations, this setting is set to 1 at				

<table>
<tr><th>Name/Description</th><th>References</th><th>Values</th><th>Component(s) affected</th><th>Version(s) applicable</th></tr>
<tr><td>installation time.</td><td></td><td></td><td></td><td></td></tr>
<tr><td>_HTTPServer_Host-Header_CheckEnabled
Enable or disable the check on the host header of the HTTP protocol. When enabled, the host header is compared against the value of <code>_WebReports_HTTPServer_HostName</code> or <code>localhost</code> or the server IP. This helps prevent any possible attempts to manipulate the mapping between an IP</td><td></td><td><table><tr><td>Default value</td><td>0 (disabled)</td></tr><tr><td>Value range</td><td>0 (disabled)
1 (enabled)</td></tr><tr><td>Setting type</td><td>Boolean</td></tr></table></td><td>Relay, Root server, Web Reports</td><td>9.5.15 and later</td></tr>
</table>

Name/Description	References	Values	Component(s) affected	Version(s) applicable
address and a DNS host name (FQDN) in WebReports. When SAML is enabled, BigFix does not automatically check the referrer while running WebReports; otherwise, the requests coming from the SAML backend are rejected.				

Top (on page 406)

HTTPS

Name/Description	References	Values		Component(s) affected	Version(s) applicable
WebReports-HTTPServer_UseSSLFlag If enabled, this setting enforces the use of HTTPS with 443 as the default port number when building the host URL.	Customizing HTTPS on Web Reports	Default value	0 (disabled)	Web Reports	All
		Setting type	Boolean		
		Value range	0 (disabled) and 1 (enabled)		
		Component restart required ?	Yes		
WebReports-HTTPServer_SSLCertificateFilePath This setting indicates the Web Reports SSL certificate file location.	Customizing HTTPS on Web Reports	Default value	Empty	Web Reports	All
		Setting type	String		
		Task available ?	No		
WebReports-HTTPServer_SSLPrivateKeyFilePath	Customizing HTTPS	Default value	Empty	Web Reports	All

Name/Description	References	Values		Component(s) affected	Version(s) applicable
This setting indicates the Web Reports SSL private key file location.	on Web Reports	Setting type	String		
		Task available ?	No		
WebReports-HTTPServer_PortNumber This setting indicates the Web Reports HTTP port.	Customizing HTTPS on Web Reports	Value range	The value is 443 if _WebReports_-HTTPServer_UseSSLFlag is enabled, or 80 otherwise.	Web Reports	All
		Setting type	Numeric		
WebReports-HTTPRedirect_Enabled This setting indicates whether HTTP redirect is en-	Customizing HTTPS on Web	Default value	1 (enabled)	Web Reports	All
		Value range	0 (disabled) 1 (enabled)		

Name/Description	References	Values		Component(s) affected	Version(s) applicable
abled for Web Reports.	Reports	Setting type	Boolean		
		Component restart required ?	Yes		
WebReports-HTTPRedirect_Port-Number This setting indicates the HTTP port number for Web Reports.	Customizing HTTPS on Web Reports	Default value	80	Web Reports	All
		Setting type	Numeric		
WebReports-HTTPServer_Require-TLS13 This setting enables or disables the use of TLS 1.3.	Customizing HTTPS on Web Reports	Default value	0 (disabled)	Web Reports	All
		Value range	0 (disabled) 1 (enabled)		
		Setting type	Boolean		

Name/Description	References	Values		Component(s) affected	Version(s) applicable
		Component restart required ?	Yes		
_BESRelay_HTTPServer_UseSSLFlag	Customizing HTTPS on REST API *(on page 148)*	Default value	0 (disabled)	Relay, Root server	All
		Setting type	Boolean		
		Component restart required ?	Yes		
_BESRelay_HTTPServer_SSLCertificateFilePath This setting indicates the path to the SSL certificate file. The .pem file might contain	Customizing HTTPS on REST API *(on page 148)*	Default value	• Windows: `\\cert-file.pem` • Linux: `/var/opt/`	Relay, Root server	All

Name/Description	References	Values		Component(s) affected	Version(s) applicable
both the certificate and private key for the Relay or Root Server, or only the certificate. The file is not included in Relay Diagnostics for security reasons.			`BESRelay/cert-file.pem`		
		Setting type	String		
		Task available ?	No		
_BESRelay_HTTPServer_SSLPrivateKeyFilePath This setting indicates the path of the Relay or Root server SSL private key file.	Customizing HTTPS on REST API *(on page 148)*	Default value	• Windows: `\\private.key` • Linux: `/var/opt/BESRelay/private.key`	Relay, Root server	All

Name/Description	References	Values		Component(s) affected	Version(s) applicable
		Setting type	String		
		Task available ?	No		
_BESRelay_HTTPServer_RequireTLS13	Customizing HTTPS on REST API *(on page 148)*	Default value	0 (disabled)	Relay, Root server	All
		Setting type	Boolean		
		Component restart required ?	Yes		

Top (on page 406)

Logging

Name/Description	References	Values		Component(s) affected	Version(s) applicable
_BESRelay_HTTPServer_LogFilePath The path to the log file.	Get Computer	Default value	• **Server** ◦ `<BES Relay Installation Path>/BESRelay.log` (Windows) ◦ `/var/log/BESRelay.log` (Linux) **Relay** ◦ `<BES Relay Install-`	Relay, Root server	All

<table>
<tr><th>Name/Description</th><th>References</th><th>Values</th><th>Component(s) affected</th><th>Version(s) applicable</th></tr>
<tr><td></td><td></td><td><table>
<tr><td></td><td>tion Path>/log-file.txt (Windows) ◦ /var/log/BESRelay.log (Linux)</td></tr>
<tr><td>Setting type</td><td>String</td></tr>
<tr><td>Component restart required ?</td><td>Yes</td></tr>
</table></td><td></td><td></td></tr>
<tr><td>_BESRelay_HTTPServer_LogFileSizeLimit
This setting is used to control the maximum size,</td><td>Relay log rotates automatically even</td><td><table>
<tr><td>Default value</td><td>52,428,800 (50MB)</td></tr>
<tr><td>Value range</td><td>0 - 4,294,967,295</td></tr>
</table></td><td>Relay, Root server</td><td>All</td></tr>
</table>

Name/Description	References	Values		Component(s) affected	Version(s) applicable
in bytes, of the log file. A maximum of 10 rotated copies of the log file are maintained in addition to the active log file (for example, on Relays, the active log file, logfile.txt and rotated copies, logfile.txt_0, logfile.txt_1, ..., logfile.txt_9). The active log file is filled in with logged data until it reaches the specified maximum size. Its content then overwrites	without setting '_BESRelay_HTTPServer_LogFileSizeLimit' Enabling debug/verbose logging for the BES Root Server and BES Relay	Setting type	Numeric (bytes)		
		Task available ?	No		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
the content of the oldest log file. The active log file is then emptied ready to continue to collect new logged data. For example, if you want your log to rotate across 10 files, each one being at most 100 KB, you must set "LogFileSizeLimit" to 100000.	services			
_BESRelay_HTTPServer_LogFileRotationLimit This setting is used to define the maximum amount of files that the		Default value: 10 Value range: 0 - 4,294,967,295 Setting type: Numeric	Relay, Root server	9.5.7 and later

Name/Description	References	Values		Component(s) affected	Version(s) applicable
server or the relay should write.		Task available ?	No		
		Component restart required?	Yes		
_BESRelay_HTTPServer_HttpLogDirectoryPath If set a connection log will be written to this path. A new log file will be created each day with the days date as the name of the file. Connection logs can get quite big, gigabytes per day, so this setting should be		Default value	Empty	Relay, Root server	All
		Setting type	String (path)		
		Value range	N/A		
		Task available ?	No		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
used for only troubleshooting purposes.				
_BESRelay_HTTPServer_HttpLogExpirationDays Sets the number of days to keep connection logs for. After this many days the older connection logs will be deleted.		Default value: 0 Setting type: Numeric (days) Value range: 0 - 4,294,967,295 Task available ? : No	Relay, Root server	All

Top (on page 406)

Relay diagnostics

You can allow users to access relay diagnostics information by enabling the access to the relay diagnostics page at http://<computer_name>:52311/RelayDiagnostics or http://<computer_name>:52311/rd, where <computer_name> is the address of the workstation where the server or the relay that you want to check is installed. When enabled, the relay diagnostics page shows which settings are enabled and contains information about buffer directory size, site gathering, clients registration and downloads.

Name/Description	References	Values		Component(s) affected	Version(s) applicable
_BESRelay_Diagnostics_PlugIn_Disable Use this setting to disable the relay diagnostics plug-in. If you disable the plug-in, you will no longer be able to reach the relay diagnostics page. The default value assigned to this setting is 0 (enabled).	Relay and Server diagnostics *(on page 1222)*	Default value	0 (enabled)	Relay, Root server	11 Patch 4 and later
		Setting type	Boolean		
		Value range	0 (enabled) 1 (disabled)		
		Component restart required ?	Yes		
_BESRelay_Diagnostics_Enable Use this setting to enable or disable the relay diagnostics page. If you specify	Relay and Server diagnostics (on	Default value	0 (disabled)	Relay, Root server	9.2.12 and later 9.5.6 and later
		Setting type	Boolean		
		Value range	0 (disabled)		

Name/Description	References	Values		Component(s) affected	Version(s) applicable
a value in _-BESRelay_-Diagnostic_-Password, the user is requested to enter the specified password to access that page. After setting the _BESRelay_Diagnostics_Password, the URL to the relay diagnostics page must use the https protocol instead of http, otherwise the browser will show the "403 forbidden" error and the re-	page 1222) Enabling (debug level) verbose logging for the BES Root Server and BES Relay services		1 (enabled)		
		Component restart required ?	No		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
lay diagnostics page will not display. Starting from V9.5.6, the default value assigned to this setting is 0 (disabled) for fresh installations and when upgrading a version in which this setting was not defined. If you upgrade an version in which a value for this setting was specified, that value is not changed.				

Name/Description	References	Values		Component(s) affected	Version(s) applicable
_BESRelay_Diagnostics_Password Use this setting to specify a password for accessing the relay diagnostics page. This setting is taken into account only if _BESRelay_Diagnostics_Enable=1. After setting the _BESRelay_Diagnostics_Password, the URL to the relay diagnostics page must use the https protocol instead of http,	Relay and Server diagnostics *(on page 1222)*	Default value	None	Relay, Root server	9.2.12 and later 9.5.6 and later
		Setting type	String		
		Component restart required ?	Yes		

<table>
<tr><th>Name/Description</th><th>References</th><th>Values</th><th>Component(s) affected</th><th>Version(s) applicable</th></tr>
<tr><td>otherwise the browser will show the "403 forbidden" error and the relay diagnostics page will not display.</td><td></td><td></td><td></td><td></td></tr>
<tr><td>_BESRelay_Diagnostics_MaxLoginAttempts
Use this setting to specify after how many failed attempts the user must be prevented from accessing the relay diagnostic page. When this threshold is reached, the relay diagnostic page</td><td>Relay and Server diagnostics (on page 1222)</td><td><table><tr><td>Default value</td><td>5</td></tr><tr><td>Setting type</td><td>Numeric</td></tr><tr><td>Value range</td><td>0 - 4,294,967,295</td></tr><tr><td>Task available ?</td><td>No</td></tr><tr><td>Component restart required ?</td><td>Yes</td></tr></table></td><td>Relay, Root server</td><td>9.5.10 and later</td></tr>
</table>

<table>
<tr><th>Name/Description</th><th>References</th><th>Values</th><th>Component(s) affected</th><th>Version(s) applicable</th></tr>
<tr><td>is locked for the number on minutes specified in _BESRelay_-Diagnostics_-LockoutDurationMinutes

Note: For any changes to this value to take effect, restart the BESRelay service.</td><td></td><td></td><td></td><td></td></tr>
<tr><td>_BESRelay_Diagnostics_LockoutDurationMinutes
Use this setting to specify for how many minutes the relay diagnostic page must remain locked after the max-</td><td>Relay and Server diagnostics (on page 1222)</td><td><table><tr><td>Default value</td><td>1</td></tr><tr><td>Setting type</td><td>Numeric (minutes)</td></tr><tr><td>Task available ?</td><td>No</td></tr><tr><td>Component</td><td>Yes</td></tr></table></td><td>Relay, Root server</td><td>9.5.10 and later</td></tr>
</table>

Name/Description	References	Values	Component(s) affected	Version(s) applicable
imum number of failed access attempts is reached. **Note:** For any changes to this value to take effect, restart the BESRelay service.		restart required ?		

Top (on page 406)

User authentication

Name/Description	References	Values	Component(s) affected	Version(s) applicable
_BESDataServer_AuthenticationTimeoutMinutes This setting	How to configure BigFix to integrate with SAML 2.0 *(on page 103)*	Default value: 5 Setting type: Numeric	Server components	All

Name/Description	References	Values		Component(s) affected	Version(s) applicable
allows you to specify the duration of the SAML authentication session when logging in to the BigFix			(minutes)		
		Value range	1 - 4,294,967,295		
		Task available ?	No		
		Component restart required ?	No		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
Console. The value for this setting is specified in minutes. **Note:** Ensure that you set for this setting a value				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
which is bigger than the values you set for the BigFix Console advanced settings.				

Top (on page 406)

Explorer

Name/Description	References	Values	Component(s) affected	Version(s) applicable
BESExplorer-HTTPServ-	Administering the BigFix Explorer	Default 9383	Explorer	11.0.2 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
er_Port-Number This setting indicates the BigFix Explorer HTTP port.		value Setting type: Numeric		
BESExplorer-HTTPServer_HostName This setting allows you to spec-	Administering the BigFix Explorer	Default value: None Setting type: String	Explorer	11.0.2 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
ify the BigFix Explorer hostname, IP Address or FQDN. If not indicated, the computer name of the BigFix				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
Explorer host is used.				

Top (on page 406)

Web Reports

Name/Description	References	Values	Component(s) affected	Version(s) applicable
WebReports-HTTPServer_Host-Name This setting allows you to specify the Web	Updating the BigFix Console shortcut How can I change the URL link to Web Reports in the BigFix Console?	Default value: http://localhost:52311 Setting type: String Component restart required: Yes, of Web Reports ?	Web Reports	9.2.5 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
Reports URL to open the Web Reports interface from the BigFix Console.				
_WebReports_Properties_Blacklist This setting allows you		Default value: None (setting not existing) Setting type: String	Web Reports	9.5.9 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
to enumerate the set of properties that are blacklisted for Web Reports. Set the value for this setting from the Big-		Component restart required Yes, of Web Reports ?		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
Fix Console, by selecting the computer on which the Web Reports component is installed and using the **Edit Com-**				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
puter Settings dialog. Upon startup, the Web Reports component reads this setting, and while loading properties from				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
the server, it discards any property whose name matches with one of those specified. Use the symbol "|" as a sepa-				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
rator. For example: _-Web-Reports_-Properties_-Blacklist=InstalledApplications Windows|Services - Windows|OS Type				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
- Windows means that none of the values associated to these properties will be displayed in Web Reports.				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
This means also that the specified properties: - Cannot be selected in the Computer page. - Cannot be				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
included in a report. - If any of those were included in a saved report, as soon as the report is opened, the				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
Web Reports component notifies the user that one or more properties specified do not exist, and it does not in-				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
clude them in the report. **Note:** Do not specify basic properties, such as the built-in ones, among the blacklisted prop-				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
erties to avoid undesired behaviors.				

Top (on page 406)

Deployment Telemetry

Name/Description	References	Values	Component(s) affected	Version(s) applicable
_Enterprise Server_MetricsCollection_Period This setting indicates the		Default value: 7 Setting type: Numeric (number of days)	Server	9.5.13 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
number of days between two data collection and upload intervals. The 0 value means that the metrics collection		Value range: 0 - 60 Task available? No		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
is disabled.				
_Enterprise Server_MetricsCollection_-Schedule This setting allows you to specify one rule to decide when the task should run.		Setting type: ASCII (<Day><h:mm>) Task available? No	Server	9.5.13 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
Enter the value in the case-sensitive format <Day><hh:mm> where: <Day> can be Mon, Tue, Wed, Thu, Fri, Sat, or Sun. <hh:mm> is				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
in 24 hour clock format. For example, this value will schedule the collection of metrics every Sunday at 9am: _-En-				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
terprise Server_-Metrics-Collection_-Schedule=Sun09:00.				
_Enterprise Server_MetricsCollection_UploadURL This setting indicates the URL that should point		Default value: Empty string Setting type: String Task available?: No	Server	9.5.13 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
to the deployment metrics server. The default value for this setting is an empty string.				
_Enterprise Server_MetricsCollection_UploadTask-		Default value 15	Server	9.5.13 and later

Name/Description	References	Values		Component(s) affected	Version(s) applicable
CheckPeriod This setting indicates the time, specified in minutes, that the task will wait before checking if there is new da-		Setting type	Numeric (minutes)		
		Task available	No?		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
ta collected. The default value for this setting is 15.				
_Enterprise Server_MetricsCollection_UseRandomInterval If setting the value to		Default value: 1 Setting type: Numeric Task available? No	Server	9.5.13 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
1, it makes the upload occur anytime in the following 12 hours. Otherwise, it makes the upload occur when there is				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
new data collected. In any case, the upload runs if the last upload occurred **before** the last run of the collection.				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
The default value for this setting is 1.				

Top (on page 406)

UI History

Set on the BigFix Server to control the Client UI History.

Name/Description	References	Values	Component(s) affected	Version(s) applicable
_BESClient_Action-Manager_-History-ModeUser This is the history		Default value: UI Setting type: String Value range: None, UI, Non-	Client	All

Name/Description	References	Values		Component(s) affected	Version(s) applicable
mode displayed to a normal user of the UI. If this and the "Technician" mode are both set to "none", the Client will not record his-			Trivial, All		
		Platform	All		
		Component restart required ?	Yes		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
tory information. Potential history modes: • "None": The history option is not enabled, and no action his-				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
tory will be displayed. • "UI": The history option is enabled, but the user will only be able to see actions that dis-				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
played UI. • "Non-Trivial": The history option is enabled, but the user will not be able to see some actions that are marked as				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
"trivial" (this includes subscription and settings actions). • "All": The history option is enabled, and will display every action that				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
runs on the machine.				
_BESClient_ActionManager_HistoryModeTech This is the history mode displayed to a "Technician" user of the UI.		Default value: All Setting type: String Value range: None, UI, NonTrivial, All Platform: All Component restart required ?: Yes	Client	All

Name/Description	References	Values	Component(s) affected	Version(s) applicable
If this and the "User" mode are both set to "none", the Client will not record history information. Potential histo-				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
ry modes: • "None": The history option is not enabled, and no action history will be displayed. • "UI": The history				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
option is enabled, but the user will only be able to see actions that displayed UI. • "Non-Trivial": The history option				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
is enabled, but the user will not be able to see some actions that are marked as "trivial" (this includes subscription and settings				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
actions). • "All": The history option is enabled, and will display every action that runs on the machine.				
_BESClient_ActionManager_History-		Default: 30 Type: Numer-	Client	All

Name/Description	References	Values		Component(s) affected	Version(s) applicable
DisplayDaysUser This setting controls the maximum amount of history information the client will send to the UI, if history			ic (days)		
		Value range	0 - 4,294,967,295		
		Platform	All		
		Component restart required ?	Yes		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
ry is enabled.				
_BESClient_Action-Manager_-History-Display-DaysTech This setting controls the maximum amount of history information the		Default: 30 Type: Numeric (days) Value range: 0 - 4,294,967,295 Platform: All Component restart required ?: Yes	Client	All

Name/Description	References	Values	Component(s) affected	Version(s) applicable
client will send to the UI when the UI is operating in "Technician" mode, if history is enabled.				

Top (on page 406)

Relays in DMZ

Parent relay

Name/Description	References	Values	Component(s) affected	Version(s) applicable
_BESRelay_DMZ_ParentEnable If set it to 1 (enabled), the parent relay can establish a persistent TCP con-	Relays in DMZ *(on page 374)*	Default value: 0 Setting type: Boolean Value range: 0 - 1 Task available ?: Yes Component restart required ?: Yes	Relay	9.5.13 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
nection.				
_BESRelay_DMZ_ChildRelayHosts Used by the parent relay to know who will be the child relays in DMZ that need to use	Relays in DMZ *(on page 374)*	Default value: None Setting type: String Task available?: Yes Component restart required?: Yes	Relay	9.5.13 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
this new capability. By default, it is an empty string. To set a value, you can specify a comma-separa-				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
rated list of `<Hostname/IP Address>,<Host`				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
`name/IP Address>` **Note:** If this value remains empty, even if you have enabled				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
 the parent relay, the parent relay will not communicate with any child relays.				
_BESRelay_DMZ_ConnectionKeepAliveSeconds It de-	Relays in DMZ *(on page 374)*	Default value: 60 Setting type: Numeric	Relay	9.5.13 and later

Name/Description	References	Values		Component(s) affected	Version(s) applicable
fines the time (in seconds) that a ping message will be periodically sent between parent and children to ensure			(seconds)		
		Task available ?	No		
		Component restart required ?	No		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
that the socket is still open. By default, it is set to 60 seconds. It can be applied to both child and parent. You must				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
set the same value on the parent and on the child relay.				
_BESRelay_DMZ_OpenChildIntervalSeconds It defines the period that the	Relays in DMZ *(on page 374)*	Default value: 60 Setting type: Numeric (seconds) Task available? No	Relay	9.5.13 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
parent relay will wait before attempting a DMZ connection to the child. By default it is set to 60 seconds.		Component restart required ? No		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
_BESRelay_DMZ_-Cleanup-Seconds It defines the period of the task that will periodically release the resources allocat-	Relays in DMZ *(on page 374)*	Default value: 60 Setting type: Numeric (seconds) Task available ?: No Component restart required ?: No	Relay	9.5.13 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
ed to the DMZ connection. By default, it is set to 60 seconds. It can be applied to both child and parent.				

Child relay

Name/Description	References	Values	Component(s) affected	Version(s) applicable
_BESRelay_DMZ_ChildEnable If set it to 1 (enabled), the child relay can establish a persistent TCP connection.	Relays in DMZ *(on page 374)*	Default value: 0 Setting type: Boolean Value range: 0 - 1 Task available?: Yes Component restart required?: Yes	Relay	9.5.13 and later
_BESRelay_DMZ_	Relays in DMZ *(on page 374)*	Default: None	Relay	9.5.13 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
ParentRelayIPs Optionally used by the child DMZ relay to filter the parent relays allowed to start a DMZ communi-		value Setting type: String Task available ?: No Component restart required ?: No		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
cation with the child. By default, it is an empty string. To set a value, a comma separated list of IP ad-				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
dresses can be specified to define which parent relays are allowed to communicate with the DMZ relay.				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
_BESRelay_DMZ_-ConnectionKeepAliveSeconds It defines the time (in seconds) that a ping message will be periodically sent between	Relays in DMZ *(on page 374)*	Default value: 60 Setting type: Numeric (seconds) Task available?: No Component restart required?: No	Relay	9.5.13 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
parent and children to ensure that the socket is still open. By default, it is set to 60 seconds. It can be applied				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
to both child and parent. You must set the same value on the parent and on the child relay.				
_BESRelay_DMZ_Cleanup-Seconds It de-	Relays in DMZ *(on page 374)*	Default value: 60	Relay	9.5.13 and later

Name/Description	References	Values		Component(s) affected	Version(s) applicable
fines the period of the task that will periodically release the resources allocated to the DMZ connec-		Setting type	Numeric (seconds)		
		Task available?	No		
		Component restart required?	No		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
tion. By default, it is set to 60 seconds. It can be applied to both child and parent.				

Peer to peer mode

Name/Description	References	Values	Component(s) affected	Version(s) applicable
_BESClient_Peer-	Working with PeerNest *(on page 378)*	Default | 0	Client	9.5.11 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
Nest_Enabled If set it to 1 (enabled), the setting enables the peer to peer mode. The client will try to download files from peers be-		value Setting type: Boolean Value range: 0 - 1 Component restart required ?: Yes		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
fore fall-back-ing to the re-lay. **Note:** When en-abling this set-ting on a Big-Fix serv-er or re-lay, the set-ting will be				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
 ignored and its value will be set to false.				
_BESClient_PeerNest_IsPassive If set it to 1 (enabled), the client will only download	Working with PeerNest *(on page 378)*	Default value: 0 Setting type: Boolean Value range: 0 (disabled) -1 (enabled) Component restart: Yes	Client	9.5.11 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
from the other peers. It will not share content with the other peers.		re-quired ?		
_BESClient_PeerNest_Priority It defines the priority used by	Working with PeerNest *(on page 378)*	Default value: 100 Setting type: Numeric Component restart: No	Client	9.5.11 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
the peers in the same subnet to download from the relay. You can specify a value range between 1 and 1000. The client		required ?		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
having the higher priority in the subnet will likely be the client to download files from the relay. The same prior-				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
ity defines also the client that will share content with the other peers.				
_BESClient_PeerNest_ResponseTimeoutSeconds It defines the time (in	Working with PeerNest *(on page 378)*	Default value: 30 Setting type: Numeric (seconds) Component: No	Client	9.5.11 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
seconds) that the client waits for answers to the request it delivers in broadcast to the peers, for example a request		restart required ?		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
for the availability of a file. The default value is 30 seconds but you can increase it in case of a complex net-				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
works with many clients. You can specify a value range between 5 and 300 seconds.				
_BESClient_PeerNest_MaxActiveFileDownloads It defines the	Working with PeerNest *(on page 378)*	Default value: 5 Setting type: Numeric Com- po- No	Client	9.5.11 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
maximum number of allowed simultaneous download operations from the other peers.		nent restart required ?		
_BESClient_PeerNest_DownloadsCacheLimitMB	Working with PeerNest *(on page 378)*	Default value 2,048	Client	9.5.11 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
It defines the maximum amount of disk used by the directory containing the files available for the other peers. The		Setting type: Numeric (MB) Value range: 0 - 4,294,967,295 Component restart required ?: No		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
default value is 2048 MB.				
_BESClient_PeerNest_MinimumDiskFreeMB This setting stops the caching of files available for the other peers if the	Working with PeerNest *(on page 378)*	Default value: 20 Setting type: Numeric (MB) Value range: 0 - 4,294,967,295 Component restart required ? No	Client	9.5.11 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
free space of the disk on which the client stores the files is less than the value of this setting.				
_BESClient_PeerNest_MinimumCacheDays Any file	Working with PeerNest *(on page 378)*	Default value: 0 Setting type: Numeric	Client	9.5.11 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
older than the value specified for _BESClient_PeerNest_MinimumCacheDays might be removed to make space for new		ic (days) Component restart required ? No		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
files, driven by the used bytes limit and the free space limit.				
_BESClient_PeerNest_MaximumCacheDays Any file older than the value spec-	Working with PeerNest *(on page 378)*	Default value: 20 Setting type: Numeric (days) Component restart: No	Client	9.5.11 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
ified for _-BESClient_-PeerNest_-MaximumCacheDays is removed even if space is not needed.		required ?		
_BESClient_PeerNest_DebugOutPath	Working with PeerNest *(on page 378)*	Default value: None	Client	9.5.11 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
Enables the PeerNest log. The PeerNest log is written to a dedicated file in the specified directory. Enable the		Setting type: String Component restart required ?: No		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
setting as follows: _-BES-Client_-PeerNest_-Debug-OutPath=<full path of the PeerNest log file> **Note:** The full path has to point				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
to a file in an existing folder; the file will be created by the client if not already existing.				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
BESClient_PeerNest_MaxLogSize This setting specifies the maximum size in bytes that a log file can reach before rotation. When the	Working with PeerNest *(on page 378)*	Default value: 4,194,304 bytes (4 MB) Setting type: Numeric (bytes) Value range: 0 bytes (Minimum value) - 1,072,693,248 bytes (1 GB) (Maximum value) Component: No	Client	10 Patch 1 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
file reaches the maximum size, the log file is rotated. **Note:** This means that the whole disk usage for the logs will be		restart required ?		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
 eleven times the value of this setting				
_BESClient_PeerNest_UseNoSpaceDownload If the BigFix Client based on the PeerNest is elected to download	Working with PeerNest *(on page 378)*	Default value: 0 (False) Setting type: Boolean Value range: 0 (False) - Do not use this enhancement; 1 (True) -	Client	10 Patch 2 and later

Name/Description	References	Values		Component(s) affected	Version(s) applicable
a file which is bigger than the maximum size of its Peer cache, it does not warn the other Peers that it is downloading			Use this enhancement.		
		Component restart required ?	No		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
it so that the other Peers can independently carry out the download, without waiting for it to complete the down-				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
load. The peer client behavior is in passive mode.				
_BESClient_PeerNest_ExcludedSubnetList This setting allows you to disable the PeerNest fea-	Working with PeerNest *(on page 378)*	Default value: None Setting type: String Value range: A semicolon-separated list. Com- Yes po-	Client	10 Patch 4 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
ture on the specified subnets. The setting is a semi-colon (;) delimited list of subnets. If the client belongs to		nent restart required ?		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
one of these subnets, the PeerNest mechanism is disabled on this client. The conditions required to execute this check are:				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
• PeerNest was enabled • The _-BES-Client_-PeerNest_-Excluded-SubnetList setting value is filled in. (Example: 10.1.77.0/25;10.1.77.129/25)				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
_BESClient_PeerNest_DisableMulticast This setting allows you to enable the new PeerNest implementation on the BigFix Clients. Clients that have	Configuring PeerNest without the multicast function *(on page 391)*	Default value: 0 Setting type: Boolean Value range: 0 (disabled) -1 (enabled) Component restart required ?: Yes	Client	11 Patch 4 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
multicast disabled (this configuration parameter set to 1) will not send any multicast messages but they will				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
still be able to receive multicast messages from other Clients, if the network configuration allows to do so.				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
_BESRelay_PeerNest_FileDownloadedTimeoutSecs This setting defines the time (in seconds) that the Relay waits after a client completed a file	Configuring PeerNest without the multicast function *(on page 391)*	Default value: 300 Setting type: Numeric (seconds) Value range: 30-4294967295 Component restart required ? No	Relay	11 Patch 4 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
download before removing the client entry for that file from the database if the client does not send the "File Available" report.				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
This is useful to prevent a client request that has won the eligibility phase from blocking other clients that want to				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
download the file (if, for instance, the client that had requested the file in the first place gets migrated or otherwise los-				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
es contact with the relay).				
_BESRelay_PeerNest_FileNewTimeoutSecs This setting defines the time (in seconds) that the Relay waits after	Configuring PeerNest without the multicast function *(on page 391)*	Default value: 300 Setting type: Numeric (seconds) Value range: 30-4294967295 Component restart required ? No	Relay	11 Patch 4 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
a client has asked for a file before removing the client entry for that file from the database if the client does not start down-				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
loading the file. This is useful to prevent a client request that has won the eligibility phase from blocking other				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
clients that want to download the file (if, for instance, the client that had requested the file in the first place gets migrated or				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
otherwise loses contact with the relay).				
_BESRelay_PeerNest_Eligibility-Timeout-Seconds This setting value represents the timeout for	Configuring PeerNest without the multicast function *(on page 391)*	Default value: 30 Setting type: Numeric (seconds) Value range: 0-4294967295 Component restart: No	Relay	11 Patch 4 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
the eligibility phase. Upon expiration, the Relay will send a PeerNest specific UDP ping to the winner client.		required ?		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
_BESClient_PeerNest_Statistics-Enabled This setting enables the BigFix Client to collect data on the PeerNest activity, send it to the	PeerNest statistics: Exporting a JSON file with aggregated data *(on page 394)*	Default value: 1 (True) Setting type: Boolean Value range: 0 (False) - Do not collect PeerNest statistics; 1 (True) - Collect PeerNest statistics Component: Yes	Client	11 Patch 6 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
BigFix Server, and store it locally for a configurable number of days. It requires the `_BES_Client_Peer_Nest_Enablement restart required ?`				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
`abled` setting to be enabled. This setting, if you enabled `_-BES-Client_-Peer-Nest_-Enable-Gather`, also collects the met-				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
rics related to the gather process.				
_BESClient_PeerNest_DataExpirationPeriod The BigFix Client periodically removes locally	PeerNest statistics: Exporting a JSON file with aggregated data *(on page 394)*	Default value: 7 days Setting type: Numeric (days) Value range: 1-any time period in days Component restart: Yes	Client	11 Patch 6 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
stored PeerNest activity data, removing old entries. This setting controls the number of days that must pass before data		required ?		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
is deleted.				
_Enterprise Server_PeerNest_Data-Expiration-Period The BigFix Server periodically removes old PeerNest data from the data-	PeerNest statistics: Exporting a JSON file with aggregated data *(on page 394)*	Default value: 14 days Setting type: Numeric (days) Value range: 1-any time period in days Component restart required ?: No	Server	11 Patch 6 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
base. This setting controls the number of days that must pass before data is deleted.				
_BESClient_PeerNest_EnableGather If set to 1	Working with PeerNest *(on page 378)*	Default value: 0 Setting type: Boolean	Client	11 Patch 6 and later

Name/Description	References	Values		Component(s) affected	Version(s) applicable
(enabled), this setting enables PeerNest site gather. The client will attempt to gather sites from peers on the same subnet before falling		Value range	0 (disabled) -1 (enabled)		
		Component restart required ?	Yes		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
back to the Relay. This setting is effective only when `_BES_Client_Peer_Nest_Enabled` is also set to 1.				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
Note: When enabling this setting on a BigFix server or relay, the setting will be ignored and its value will be set				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
 to false.				

Automatic Server Clean Up

This table describes the settings related to the service to clean up the files uploaded by the Archive Manager to the BigFix Server.

Name/Description	References	Values	Component(s) affected	Version(s) applicable
Enterprise Server-Archive-Manager-Cleanup_-Enable This setting, if enabled, deletes the files uploaded by the		Default value: 0 (disabled) Setting type: Boolean Value range: 0-1 Task available?: No Component restart: Yes	Server	11.0.2 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
Archive Manager to the BigFix Server.		required ?		
Enterprise Server-Archive-Manager-Cleanup_-Refresh-Hours This setting allows you to specify the		Default value: 24 (Hours) Setting type: Numeric (Hours) Task available ?: No Component restart required ?: Yes	Server	11.0.2 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
amount of hours after which the clean up service runs.				
Enterprise Server-Archive-Manager-Cleanup_-Older-ThanDays This setting specifies for how many		Default value: 7 (Days) Setting type: Numeric (Days) Task available?: No Component restart: Yes	Server	11.0.2 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
days you want to keep the uploaded files before they are deleted from the server.		required ?		

User base management

Name/Description	References	Values	Component(s) affected	Version(s) applicable
_BESRelay_LDAP_RefreshSeconds	User-based management *(on page 93)*	Default value 900	Server	11.0.7 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
The BigFix root server periodically refreshes LDAP group membership information. This setting controls the		Setting type: Numeric Component restart required ?: Yes		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
time interval between the refreshes.				
_BESClient_ManagedBy_UserPrincipalName This setting acts as the explicit identity for the	User-based management *(on page 93)*	Default value: None Setting type: String Component restart required ?: No	Client	11.0.7 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
managing user. This value takes absolute priority and serves as a reliable fallback when standard directory lookups				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
fail or when the automated name field is found to be empty. This setting is designed to explicitly replace the				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
`user principal name` property of the `idp user` object. If the`idp user` object does not exist the client will automat-				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
ically create a new one containing all standard default values. The only exceptions are the `user principal`				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
`name`, which will be set exactly to the value provided in this setting, and the `directory type`, which will be flagged				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
as `Manual`.If the `idp user-` object already exists all current properties of the object will remain strictly unchanged,				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
with the sole exception of the `user principal name`, which will be overwritten by the value specified in this setting.				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
_BESClient_ManagedBy_MemberOf This setting supports a semi-colon-separated list of group names (e.g., Group 1; Group 2). Spaces within individ-	User-based management *(on page 93)*	Default value: None Setting type: String Component restart required ?: No	Client	11.0.7 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
ual group names and surrounding the delimiter are fully valid. These values force the generation of temporary client-side				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
Identity Provider (IdP) group objects, completely overriding any directory data natively provided by the IdP.				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
This setting is used to forcefully define the group names to which an `idp user` belongs. Specifically, it overrides the `name`				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
property of the `idp groups` objects associated with an `idp user`. The same creation and override logic applies here.				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
The client parses the semicolon-separated list of groups. If the underlying group objects do not exist, the client gener-				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
ates them using default values, assigning the parsed values to the group `name` property and setting the `directory`				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
`type` to `Manual`. If they do exist, the group names are forced to match your setting while preserving the other properties				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
according to the same manual override rules.				

Top (on page 406)

External Repository Sites

Name/Description	References	Values	Component(s) affected	Version(s) applicable
_Enterprise Server_RepositorySite_-Enable This setting enables the	External Repository Sites *(on page 1191)*	Default value: 1 (enabled) Setting type: Boolean Value range: 0 (disabled)	Server	11.0.7 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
Root Server Repository Site scheduler. A value of "0" prevents the scheduler from being registered; any non-zero		- 1 (enabled) Component restart required ? Yes		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
value enables it.				
_Enterprise Server_RepositorySite_SyncPeriodMinutes This setting specifies the time interval, in minutes, used by the Root	External Repository Sites *(on page 1191)*	Default value: 1 Setting type: uint Value range: None Component restart required ?: Yes	Server	11.0.7 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
Server Repository Site scheduler to check configured Repository Sites for updates.				
_Enterprise Server_RepositorySite_-NumArchiveVersions	External Repository Sites *(on page 1191)*	Default value: 7 Setting type: uint	Server	11.0.7 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
This setting specifies the maximum number of archived Repository Site versions retained in the <BES Server storage		Value range: None Component restart required ?: Yes		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
directory>\Repository-Site-Gatherer\Sites\<Repository Site folder>\Archive directory after a successful propaga-				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
tion. A value of "0" retains all versions.				

Top (on page 406)

Miscellaneous

Name/Description	References	Values	Component(s) affected	Version(s) applicable
_BESClient_Report_Encryption This setting allows you to set up	Managing Client Encryption (on page 1073) Enabling encryption on Clients Encryption	Default value: optional Setting type: String Value range: **required** Causes	Client	All

Name/Description	References	Values	Component(s) affected	Version(s) applicable
your BigFix Client to encrypt all outgoing reports to protect data. You must have encryption enabled for your BigFix			the Client to always encrypt. If there is no encryption certificate available in the masthead or if the target computer	

Name/Description	References	Values		Component(s) affected	Version(s) applicable
Server before enabling it for your Clients.			er (Relay or Server) cannot accept encryption, the Client will not send reports.		
		optional	The Client encrypts if it		

Name/Description	References	Values		Component(s) affected	Version(s) applicable
			can, otherwise it sends its reports in clear-text.		
			none No encryption is done, even if an encryption certificate is		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
		present. This allows you to turn off encryption after you enable it.		
		Task available ? No		
		Component restart required ? No		
_Enterprise Server_MSEn-	Adding Identity Provider Oper-	Default 0	Server	11 Patch 3 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
traID_PageSize This setting was introduced to let BigFix users configure the Microsoft Entra page size for REST API responses.	ators (on page 84) Associating an Identity Provider Group (on page 91)	value Setting type: integer Value range: 0 - 999 Task available ?: No Component restart required ?: No		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
This property must be set on the machine where the BigFix Server is running. By default, it is set to 0, indi-				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
cating that no paging option is applied, and the Microsoft Entra default value of 100 items is returned for each re-				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
quest. Increasing the page size can enhance performance when dealing with a large number of Microsoft Entra users or groups.				

Name/Description	References	Values		Component(s) affected	Version(s) applicable
_BESClient_PowerShell_DisableExecPolicyBypass The Powershell scripts are executed using the `-ExecutionPolicy Bypass` option	Pre-Execution Action Script tab Post-Execution Action Script Tab Edit Actions Tab Action Script Tab	Default value	0 (False)	Client	10 Patch 4 and later
		Setting type	Boolean		
		Value range	0 (False) The Powershell scripts will run using the `-ExecutionPolicy Bypass` option		

Name/Description	References	Values		Component(s) affected	Version(s) applicable
to allow skipping the default execution policy for the current process only. When this setting is enabled (val-		- 1 (True) The Powershell scripts will run without the `-ExecutionPolicy Bypass` option			
		Task available?	No		
		Component restart	No		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
ue is set to 1), the script runs without the `-Execution-Policy Bypass` option and the Client uses the default		required ?		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
execution policy set in the operating system. The default value is 0.				
_BESRelay_Server-Based-Group_Re-	Creating Server Based Computer Groups	Default value: 1	Server	10 Patch 4 and later

Name/Description	References	Values		Component(s) affected	Version(s) applicable
freshMinutes Controls the frequency at which the BigFix Server refreshes all the Server Based Computer Groups that you have		Setting type	DWORD		
		Value range	Unrestricted		
		Task available?	No		
		Component restart required?	Yes		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
defined.				
BESRelay-Server-Based-Group_-Ignore-Agents-NotReporting-AfterHours Specifies the amount of hours after which the not reporting devices	Creating Server Based Computer Groups	Default value: 0 Setting type: Numeric (Hours) Value range: Unrestricted Task available?: No Component restart required?: Yes	Server	10 Patch 4 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
should not be taken into account by the BigFix Server for a membership in a Server Based Computer Group. When				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
setting the value to 0, you disable the feature.				
EnableLockHints A new server setting available on Windows under HKEY_-		Default value: 0 Setting type: Boolean Value range: 0 - 1 Task available?: No Component: Yes	Server	9.5.11 and later

Name/Description	References	Values	Component(s) affected	Version(s) applicable
LOCAL_-MACHINE\SOFTWARE\Wow6432Node\BigFix\Enterprise Server\Database If set to 1, it enables the BES Server to use lock hints in its inter-		nent restart required ?		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
nal queries to the SQL Server. This setting helps if the Server is involved in deadlocks, but it might cause a small performance				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
decrease.				
_BESServer_Database_-Password Configure to set the password while updating the database. Enclose the password text within		Default value: None Setting type: String Component restart required ?: Yes	Server	All

Name/Description	References	Values	Component(s) affected	Version(s) applicable
double quotation marks.				
_BESClient_Cryptography_FipsMode Each BigFix component works in a FIPS-mode determined by the masthead con-	Configuring FIPS 140-2 on the BigFix Server (on page 1069) Configuring FIPS 140-3 on the BigFix Server (on page 1072)	Default value: optional Setting type: String Task available?: No Component restart required?: Yes	All	All

Name/Description	References	Values	Component(s) affected	Version(s) applicable
figuration, and the component version. In addition, this client setting can be used to override the FIPS configu-				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
ration written in the masthead for the BES Client, the BES Console and the Web Reports components. When setting the value				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
to **none**, these components will work in non-FIPS mode, regardless of the FIPS configuration written in the masthead.				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
When setting the value to **required**, they will work in FIPS 140-2 mode up to version 11.0.6 and in FIPS 140-3 mode if at version				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
11.0.7 or later, regardless of the FIPS configuration written in the masthead. When setting the value to **option-**				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
al, these components will work in a FIPS mode that depends on the FIPS configuration written in the masthead, and on				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
their version.				
_BESClient_LicenseType_productname When you categorize the clients, you assign to the clients non-default metrics for each se-	Categorizing the clients	Default value: None Setting type: String Component restart required ?: No	Client	All

Name/Description	References	Values	Component(s) affected	Version(s) applicable
lected product, from the licensed ones by setting this configuration parameter.				
_BESClient_Comm_ForceNotActionsiteEvaluation Keeps eval-		Default value: 0 (False) Setting type: Boolean	Client	9.5.14 and later

Name/Description	References	Values		Component(s) affected	Version(s) applicable
uating custom and external sites when a new action-site version is received. By default, when a new version of		Value range	0 (False) Disables custom and external sites evaluation - 1 (True) Enables custom and external sites evaluation		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
action-site, mailbox site or operator site is received, the BigFix client limits the evaluation to these three site		Task available ? No Component restart required ? No		

Name/Description	References	Values	Component(s) affected	Version(s) applicable
types so to send quickly a report. If the BigFix client is subscribed to many operational sites, and the content of the				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
action-site and of the operational sites changes frequently, the evaluation of the custom and external sites might				

Name/Description	References	Values	Component(s) affected	Version(s) applicable
take a long time to complete. Use caution when enabling this setting because the report could be delayed.				

Top (on page 406)

Chapter 23. Additional configuration steps

These topics explain additional configuration steps that you can run in your environment.

Installing and configuring the email notification service

You can use the email notification service to send automatic email notifications to notify you about the completion status of Baselines that you run. To use this notification feature, you must first install the notification service by running a Task. After you have installed the service, you must configure it by running another Task.

The BigFix Server Plugin Service must be installed on the BigFix Server and the REST API master operator credentials must be configured using Fixlet 1294 on BES Support.

The installation and setup Tasks are available from the BES Support site. To help you ensure that you run the Tasks in the correct order, the installation and setup Tasks become relevant in the order in which you need to run them. So each Task will become relevant only when you need to run it.

Complete the following steps to install and set up the notification service.

1. To install the notification service, from the **Fixlets and Tasks** node of the navigation tree in BES Support, select one of the following Task, depending on whether you are installing the notification service on Windows or Linux:
    - Task 2238 `Install Latest Notification Service (SHA256)` to install the notification service on Windows operating system. When you run the Task, target the BigFix Server and enter the port number on which you want the notification service to listen. Enter a string in the **passphrase for cert generation** field (For secure communication, the Notification service requires SHA 256 certificate that is generated using the OpenSSL). This Task downloads and installs the notification service.
    - Task 2241 `Install Latest Notification Service (sha256 - RHEL)` to install the notification service on Linux. When you run the Task, target the BigFix Server and enter the port number on which you want the notification

service to listen. Enter a string in the **passphrase for cert generation** field (For secure communication, the Notification service requires SHA 256 certificate that is generated using the OpenSSL). This Task downloads and installs the notification service.

Note: To install the latest version of Notification Service 3.2.3, upgrade RHEL to 8.2 or later.

If the installation does not complete successfully, check to see if a Task has become relevant in the **Notification > Warnings** folder. If there is a Task that is relevant in the **Warnings** folder, run the relevant Task. After the Task has completed, run the installation Task again.

2. Activate analysis 2243 `Notification Service Settings`.
3. Run Task 2240 `Configure Settings for Notification Service` to configure the SMTP login settings for the email notification service. Complete the form, as follows, and then click **Take Action**:
    - **Notification Service Port**: you can change the value here to update the port number that the notification service listens on, as set in Task 2238 `Install Latest Notification Service` or Task 2241 `Install Latest Notification Service (RHEL)`.
    - **From Email Address**: you can change the value here to update the default `From` email address that is displayed in the `From` field of the notification email and the email address as configured in Task 2244 **Send an Email Notification**, which is the Task that you use to send the email notification.
    - **SMTP Method**: select an SMTP method for the notification service, either **Login** or **None**. If you select **None**, no SMTP authentication is used and the **User name** and **Password** fields are disabled. If you select **Login**, you must authenticate with a user name and password is required.
    - **SMTP Host**: enter the IP address, host name or fully qualified host name of the SMTP server. This is a required field.
    - **SMTP Port**: accept the default port number value of 25 or enter a different value for the port number of the SMTP server. This is a required field.

- **User name**: enter the user name for the email account. This is a required field if you selected **Login** as the SMTP method.
- **Password**: enter the password for the email account. This is a required field if you selected **Login** as the SMTP method.
- **Confirm Password**: confirm the password that you entered in the previous field. This is a required field if you selected **Login** as the SMTP method.

After you have completed this step, the notification service is set up and ready to use.

The notification service is set up and configured according to your settings.

Note: To uninstall the notification service, use `Task 2239 Uninstall Notification Service` to uninstall from Microsoft Windows platforms or `Task 2242 Uninstall Notification Service (RHEL)` to uninstall from Linux platforms.

Sending email notifications

You can use the email notification service to send automatic email notifications to notify you about the completion status of Baselines that you run.

Before you can send email notifications, you must set up and configure the notification service.

After you have installed and configured the notification service, you can send email notifications to notify you about the completion status of Baselines.

Complete the following steps to send email notifications.

1. To use the notification service, add a modified copy of Task 2245 `Sample Task: Send an Email Notification` as a component in a Baseline at the point at which you want the notification to be sent. For example, if you want an email notification to be sent when the Baseline has completed, add the Task as the last component in the Baseline. Copy and modify the Task as follows:

- Copy Task 2245 `Sample Task: Send an Email Notification using Static Action Script Content`.
- Modify the copy of the Task by removing all the relevances except for the relevance "True" and by changing the details in the action script to include your specific settings for the following keys. The keys included in the following example represent the minimum required keys to send a notification:

```
// 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
```

The complete list of keys are listed in the following table.

Table 6. Notification keys

Key	Description
`to`	The email addresses to whom the notification is be sent. The valid values are a comma (,) separated list of valid email addresses. This is a required key.
`from`	The email address that is displayed as the sender of the notification email. The email address does not have to exist, but the value that you specify must be in the format of a valid email address. This is a required key.
`subject`	The notification summary that appears as the subject of the email. Any text is a valid entry for this key.

Key	Description
	The text that you enter can include tokens that are replaced at run-time. This is a required field.
`body`	The main notification text content for the email. You can enter any text for this key, including tokens that are replaced at run-time. This is a required field.
`failure-trigger`	This defines the number of failures that cause the notification to be sent. When this key is specified, the notification is sent when the specified number of failures occurs. When this key is omitted, the notification is sent when it completes. Enter a whole number greater than zero. This is an optional key.
`scope`	Determines which action is examined for result statuses when determining if an action failed or completed successfully. When this key is omitted, the results of the action containing these notification comments are examined. The valid value is `parent`, which means the results of the parent Baseline are examined instead of the action containing the notification comments. This is an optional key.

   - Save the modified copy of the Task.
   - Add your modified copy to the Baseline for which you want to send the email notification.

   Review the Task description for complete information about how to modify the Task and include in a Baseline. Examples are included in the **Example** section below.
2. Run the Baseline. When the Baseline completes, an email notification is sent to the email addresses that you specified.

The following examples show how you can use the keys:

If you add the Task with the following notification comments as a component in a baseline, it sends an email when this Task (component) completes (for example, this may be useful if added to the middle of a baseline):

```
// 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
```

When a Task consisting of the following notification comments is added as a component in a Baseline, it sends an email when that overall Baseline completes:

```
// 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
```

You can add the previous two examples only to baselines that are statically targeted. Baselines targeted by group, property or name list do not enable a notification to be sent.

You can specify the following example in Baselines targeted statically or dynamically:

```
// 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
```

Configuring FillDB

The FillDB process runs on the BigFix Server system and is responsible for storing the information returned from the BigFix Agents into the BigFix database.

This information can be:

- Data, such as the value of retrieved properties or the result of the evaluation of the applicability relevance or the success criteria of Fixlets and Tasks.
- The information contained in a report returning the result of a BigFix Query.

This is how you can configure FillDB processing:

- Configuring the FillDB database performance *(on page 1035)*
- Increasing the size of the FillDB buffer directory *(on page 1036)*
- Enabling FillDB parallel processing *(on page 1038)*

Configuring the FillDB database performance

DatabaseBoostLevel parameter

The FillDB utility offers a parameter called `DatabaseBoostLevel` to optimize BigFix performance.

On **Windows** systems: The possible values for the `DatabaseBoostLevel` parameter are 1 for enabled and 0 for disabled. The default value is 1.

On **Linux** systems (fresh installations and upgrades): The `DatabaseBoostLevel` parameter value is always: Database Boost Level is ON with maxBatchSize = 1000.

The improvements to BigFix performance depend on the environment in which BigFix runs. To find the best performance configuration for your environment, tune the data insertion mechanism in the FillDB database as follows:

1. Enable the performance log.
   Set the following string value in the `[HKLM\Software\Wow6432Node\BigFix\Enterprise Server\FillDB]` registry:

   ```
   "PerformanceDataPath"[REG_SZ] = "[BigFix Server
    folder]\FillDB\FillDBperf.log"
   ```

2. Restart the FillDB service and monitor the performance log for a while, registering the database insertion rate in "rows per second" of the various tables.
3. Add the `DatabaseBoostLevel` DWord value to the registry key `HKLM\Software\Wow6432Node\BigFix\Enterprise Server\FillDB` and set it to 0.
4. Restart the FillDB service and monitor the performance log again for a while, registering the database insertion rate in "rows per second" of the various tables.
5. Compare the insertion rate before and after setting the new value of the `DatabaseBoostLevel` parameter, ensuring that you keep the same level of workload during the monitoring. The more rows that are processed per second, the better the performance. The key tables to monitor are: questionresults, fixletresults, actionresults and longquestionresults. The number of rows processed per table is an indicator of the importance that the table has in your environment.

Increasing the size of the FillDB buffer directory

The FillDB buffer directory temporarily stores reports from the clients before they are stored into the database.

By default the directory is full if it contains 3MB of files or if it has more than 10,000 files. The consequence is that the information is not sent to the BigFix server quickly, and it might be a severe problem.

You can configure the FillDB buffer directory and the maximum number of hold files by performing the following steps:

On Windows systems:

1. Add the following keys to the registry path `HKLM\Software\Wow6432Node\BigFix\Enterprise Server\PostResults`:

   BufferDirectoryMaxSize

   It defines the maximum size of the FillDB buffer directory, in bytes. The default value is 3MB.

   Note: Do not increase this value over 20MB without specific guidance from HCL Support.

   BufferDirectoryMaxCount

   It defines the maximum number of files allowed in the FillDB buffer directory. The default value is 10,000.
2. Restart the FillDB service.

Note: On Windows systems, if you are adding these keys to a BigFix relay and not to a BigFix server, the registry path where to add them is `HKLM\Software\BigFix\Enterprise Server\PostResults`.

On Linux systems:

1. Add the following lines to the `/var/opt/BESServer/besserver.config` file:

```
[Software\BigFix\Enterprise Server\PostResults]
BufferDirectoryMaxSize = <SIZE_IN_BYTES>

[Software\BigFix\Enterprise Server\PostResults]
BufferDirectoryMaxCount = <MAX_NUMBER_OF_FILES>
```

   where:

   BufferDirectoryMaxSize

   It defines the maximum size of the FillDB buffer directory, in bytes. The default value is 3MB.

BufferDirectoryMaxCount

It defines the maximum number of files allowed in the FillDB buffer directory. The default value is 10,000.

2. Restart the FillDB service.

Enabling FillDB parallel processing

On the BigFix Server, the FillDB process is responsible to run in a single thread the following activities.

- Read the buffer directory content.
- Parse the report, which includes:
  - Decrypting the encrypted reports.
  - Decompressing the compressed reports.
- Store the report data in the database.
- Replicate content from other DSA servers (optional).

An additional thread is responsible for performing the same type of processing for the reports returned by the BigFix Query processing.

The parallel processing is enabled by default during a fresh installation and upgrade according to the following rules:

- If the machine has 6 to 9 cores, the parallelism (**ParallelismEnabled**) is enabled for normal reports by configuring 3 parsing threads and 3 database update threads.
- If the machine has at least 10 cores, the parallelism (**ParallelismEnabled** and **ParallelismEnabledForQuery**) is enabled both for normal and for query reports by configuring for each of them 3 parsing threads and 3 database update threads for a total of 12 threads.
- If the machine has less than 6 cores, the **ParallelismEnabled** and the **ParallelismEnabledForQuery** settings have 0 as their default values, and then parallelism is disabled.

You can manually enable or disable the FillDB parallel processing by configuring the following settings on the BigFix Server:

- ParallelismEnabled
- ParallelismEnabledForQuery

Once you enabled the FillDB parallel processing, you can configure its behavior by specifying the following settings on the BigFix Server:

- NumberOfParsingThreads
- NumberOfDBUpdatingThreads
- MaxNumberOfReportsReadyForDB
- MinNumberOfReportsReadyForDB
- MaxNumberOfReportsInParsingQueue
- NumberOfParsingThreadsForQuery
- NumberOfDBUpdatingThreadsForQuery
- MaxNumberOfQueryReportsReadyForDB
- MinNumberOfQueryReportsReadyForDB
- MaxNumberOfQueryReportsInParsingQueue

For more information about these settings, see Configuring parallel FillDB *(on page 1040)*.
Run the following steps to activate changes on one or more of the settings specified above:

If the BigFix Server is installed on a Windows system:

1. Stop the BES FillDB service.
2. Update the values of the settings as appropriate in the Windows registry.
3. Start the BES FillDB service

If the BigFix Server is installed on a Linux system:

1. Stop besfilldb, for example `systemctl stop besfilldb`
2. Stop besserver, for example `systemctl stop besserver`
3. Update the values of the settings as appropriate in the `besserver.config` file.

4. Start besserver, for example `systemctl start besserver`
5. Start besfilldb, for example `systemctl start besfilldb`

Up to BigFix Version 11.0.1, for backward compatibility reasons, you can also run the command:

```
/etc/init.d/besfilldb stop|start
/etc/init.d/besserver stop|start
```

Configuring parallel FillDB

Name/Description	Values		Component(s) affected	Version(s) applicable	References
Parallelism-Enabled This setting allows to enable parallel processing	Default Value	Value depending from the configuration as reported in the referenced page	Server	9.5.5 and later	Enabling FillDB parallel processing *(on page 1038)*
	Setting Type	REG_DWORD on Windows, entry in besserv-			

Name/Description	Values		Component(s) affected	Version(s) applicable	References
for agents reports. For details on the occasions on which parallelism is enabled, see Enabling FillDB parallel process-		er.config on Linux			
	Value range	0 - 1			
	Task available	No			
	Requires restart	FillDB Restart on Windows systems and FillDB and BesRootServer restart on Linux			

Name/Description	Values		Component(s) affected	Version(s) applicable	References
ing *(on page 1038)*.					
NumberOfParsingThreads This setting defines the number of threads that are responsible for parsing	Default Value	3	Server	9.5.5 and later	Enabling FillDB parallel processing *(on page 1038)*
	Setting Type	REG_DWORD on Windows, entry in besserver.config on Linux			
	Value range	1 - 5			
	Task available	No			
	Requires restart	FillDB Restart on Windows systems and FillDB and Bes-			

Name/Description	Values		Component(s) affected	Version(s) applicable	References
the reports. It is used only if the parallelism is enabled.		RootServer restart on Linux			
NumberOfDBUpdatingThreads This setting defines the number of threads that	Default Value Setting Type Value range	3 REG_DWORD on Windows, entry in besserver.config on Linux 1 - 5	Server	9.5.5 and later	Enabling FillDB parallel processing *(on page 1038)*

Name/Description	Values		Component(s) affected	Version(s) applicable	References
store report data in the DB. It is used only if the parallelism is enabled.	Task available	No			
	Requires restart	FillDB Restart on Windows systems and FillDB and BesRootServer restart on Linux			
MaxNumberOfReports-ReadyForDB This setting represents the maximum	Default Value	1000 / NumberOfDBUpdatingThreads	Server	9.5.5 and later	Enabling FillDB parallel processing *(on page 1038)*
	Setting Type	REG_DWORD on Windows, entry in besserv-			

<table>
<tr><th>Name/Description</th><th colspan="2">Values</th><th>Component(s) affected</th><th>Version(s) applicable</th><th>References</th></tr>
<tr><td>number of reports that a DB updating thread can process in a single run. It is used only if the parallelism is enabled.</td><td colspan="2"><table>
<tr><td></td><td>er.config on Linux</td></tr>
<tr><td>Value range</td><td>500-5000 / NumberOfDBUpdatingThreadsStandard</td></tr>
<tr><td>Task available</td><td>No</td></tr>
<tr><td>Requires restart</td><td>FillDB Restart on Windows systems and FillDB and BesRootServer restart on Linux</td></tr>
</table></td><td></td><td></td><td></td></tr>
</table>

Name/Description	Values		Component(s) affected	Version(s) applicable	References
MinNumberOfReportsReadyForDB This setting represents the minimum number of reports that a DB updating thread can process in a single	Default Value	MaxNumberOfReportsReadyForDB / 2	Server	9.5.5 and later	Enabling FillDB parallel processing *(on page 1038)*
	Setting Type	REG_DWORD on Windows, entry in besserver.config on Linux			
	Value range	MaxNumberOfReportsReadyForDB / 1-3			
	Task available	No			
	Requires restart	FillDB Restart on Windows systems			

Name/Description	Values		Component(s) affected	Version(s) applicable	References
run. It is used only if the parallelism is enabled.		and FillDB and BesRootServer restart on Linux			
MaxNumberOfReportsInParsingQueue This setting represents the maximum number of reports	Default Value Setting Type Value range	10 * NumberOfParsingThreads REG_DWORD on Windows, entry in besserver.config on Linux (2-20) * Number-	Server	9.5.5 and later	Enabling FillDB parallel processing *(on page 1038)*

Name/Description	Values		Component(s) affected	Version(s) applicable	References
that can be in the queue waiting for a parsing thread to process them. It is used only if the parallelism is enabled.		OfParsingThreads			
	Task available	No			
	Requires restart	FillDB Restart on Windows systems and FillDB and BesRootServer restart on Linux			
ParallelismEnabledForQuery	Default Value	Value depending	Server	9.5.5 and	Enabling FillDB parallel processing *(on page 1038)*

Name/Description	Values		Component(s) affected	Version(s) applicable	References
This setting allows to enable parallel processing for BigFix query reports.		from the configuration as reported in the referenced page		later	
	Setting Type	REG_DWORD on Windows, entry in besserver.config on Linux			
	Value range	0 - 1			
	Task available	No			
	Requires restart	FillDB Restart on Windows systems and FillDB			

Name/Description	Values		Component(s) affected	Version(s) applicable	References
		and BesRootServer restart on Linux			
NumberOfParsingThreadsForQuery This setting defines the number of threads that are responsible for parsing	Default Value	3	Server	9.5.5 and later	Enabling FillDB parallel processing *(on page 1038)*
	Setting Type	REG_DWORD on Windows, entry in besserver.config on Linux			
	Value range	1 - 5			
	Task available	No			
	Requires restart	FillDB Restart on Windows systems and FillDB and Bes-			

Name/Description	Values		Component(s) affected	Version(s) applicable	References
the BigFix query reports. It is used only if the parallelism is enabled for BigFix query reports' processing.		RootServer restart on Linux			
NumberOfDBUpdatingThreadsForQuery	Default Value	3	Server	9.5.5 and later	Enabling FillDB parallel processing *(on page 1038)*

Name/Description	Values		Component(s) affected	Version(s) applicable	References
This setting defines the number of threads that store BigFix query report data in the DB. It is used only if the parallelism is enabled	Setting Type	REG_DWORD on Windows, entry in besserver.config on Linux			
	Value range	1 - 5			
	Task available	No			
	Requires restart	FillDB Restart on Windows systems and FillDB and BesRootServer restart on Linux			

Name/Description	Values		Component(s) affected	Version(s) applicable	References
for BigFix query reports' processing.					
MaxNumberOfQueryReportsReadyForDB This setting represents the maximum number of BigFix query	Default Value	1000 / NumberOfDBUpdatingThreadsForQuery	Server	9.5.5 and later	Enabling FillDB parallel processing *(on page 1038)*
	Setting Type	REG_DWORD on Windows, entry in besserver.config on Linux			
	Value range	(500-5000) / NumberOfDBUp-			

<table>
<tr><th>Name/Description</th><th>Values</th><th>Component(s) affected</th><th>Version(s) applicable</th><th>References</th></tr>
<tr><td>reports that a DB updating thread can process in a single run. It is used only if the parallelism is enabled for BigFix query re-</td><td><table><tr><td></td><td>datingThreadsForQuery</td></tr><tr><td>Task available</td><td>No</td></tr><tr><td>Requires restart</td><td>FillDB Restart on Windows systems and FillDB and BesRootServer restart on Linux</td></tr></table></td><td></td><td></td><td></td></tr>
</table>

Name/Description	Values		Component(s) affected	Version(s) applicable	References
ports' processing.					
MinNumberOfQueryReportsReadyForDB This setting represents the minimum number of BigFix query reports that a DB up-	Default Value	MaxNumberOfQueryReportsReadyForDB / 2	Server	9.5.5 and later	Enabling FillDB parallel processing *(on page 1038)*
	Setting Type	REG_DWORD on Windows, entry in besserver.config on Linux			
	Value range	MaxNumberOfQueryReportsReadyForDB / (3-1)			

Name/Description	Values		Component(s) affected	Version(s) applicable	References
dating thread can process in a single run. It is used only if the parallelism is enabled for BigFix query reports' processing.	Task available	No			
	Requires restart	FillDB Restart on Windows systems and FillDB and BesRootServer restart on Linux			

Name/Description	Values		Component(s) affected	Version(s) applicable	References
MaxNumberOfQueryReportsInParsingQueue This setting represents the maximum number of BigFix query reports that can be in the queue waiting	Default Value	10 * NumberOfParsingThreadsForQuery	Server	9.5.5 and later	Enabling FillDB parallel processing *(on page 1038)*
	Setting Type	REG_DWORD on Windows, entry in besserver.config on Linux			
	Value range	(2-20) * NumberOfParsingThreadsForQuery			
	Task available	No			
	Requires restart	FillDB Restart on Windows systems			

Name/Description	Values		Component(s) affected	Version(s) applicable	References
for a parsing thread to process them. It is used only if the parallelism is enabled for BigFix query reports' processing.		and FillDB and BesRootServer restart on Linux			

Configuring ODBC data sources

How to use Open DataBase Connectivity (ODBC) to set up and configure ODBC Data Sources with BigFix.

The BigFix components that use a database connect to it using Open DataBase Connectivity (OBDC).

To open an ODBC connection, you need:

- An ODBC driver, a piece of software allowing your application call the ODBC APIs of the database engine.
- An ODBC data source, the information identifying the target database and detailing the connection parameters.

There can be several ODBC drivers installed on the same computer and several ODBC data sources saved on it. Often, the data source also specifies what driver to use to connect to the target database.

Each ODBC data source is identified by a Data Source Name (DSN), not to be confused with a DNS (Domain Name System), which has a similar acronym but is an entirely different thing.

ODBC on Windows

On Windows, an ODBC data source can be 32-bit or 64-bit. There can be two sources with the same name if one is 32-bit and the other is 64-bit.

ODBC data sources with the same name but different bitness often refer to the same database and have the same connection parameters, except for the path of the ODBC driver they specify.

In fact, ODBC drivers often have a 32-bit dll for 32-bit applications and 64-bit dll for 64-bit applications.

ODBC data sources are saved in the registry:

- 32-bit ODBC data sources appear in HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\ODBC\ODBC.INI
- 64-bit ODBC data sources appear in HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBC.INI

You can edit them directly or configure them using these dedicated tools:

- Start > Windows Administrative Tools > ODBC Data Sources (32-bit)
- Start > Windows Administrative Tools > ODBC Data Sources (64-bit)

The BigFix Platform components use these ODBC data sources on Windows:

- enterprise_setup, used by the installers, especially during upgrades
- bes_bfenterprise, used by the BigFix Server to access its database
- LocalBESReportingServer, used by Web Reports to access its database

All the above ODBC data source names have a corresponding 32-bit source and a 64-bit source.

In total there are 6 BigFix data sources (3 pairs).

The BigFix Administration Tool (BESAdmin) usually relies on the same ODBC data source as the BigFix Server, but it may use a different data source depending on the command it runs.

The BigFix WebUI does not use ODBC data sources and saves its connection data internally.

If you want to change a BigFix ODBC data source, remember to:

- Update both the 32-bit and 64-bit versions of the data source.
- Ensure that the corresponding settings in other data sources remain consistent.

If you do not keep all data sources aligned, you may encounter database connection issues when using certain BigFix functions but not while using others.

For example, if you change bes_bfenterprise and forget to update the corresponding information in enterprise_setup, you might only find out during the BigFix pre-upgrade checks.

BigFix components might connect to databases hosted on SQL Server either:

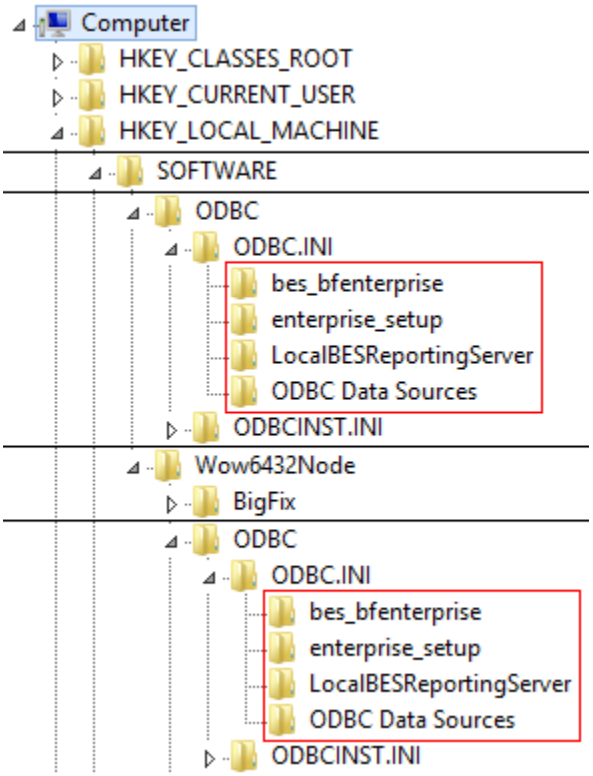
- Via Windows authentication, which relies on the Windows credentials of the user running the application process.
- Via SQL Server authentication, which relies on separate credentials, unrelated to Windows user credentials.

When a BigFix component uses Windows authentication, the user running its service is the same that will access the database. This is true for all components except for the BigFix WebUI.

When connecting to a local database, BigFix uses Windows authentication by default.

When connecting to a remote database, BigFix can be configured to use Windows authentication or SQL Server authentication.

This image shows the ODBC keys of a computer with both the BigFix Server and Web Reports installed.



Changes introduced in Version 11 Patch 6

Given a few differences in how the two drivers can be configured, any customization of the BigFix ODBC data sources done prior to upgrading to Version 11.0.6 might no longer work as expected after upgrading to Version 11.0.6. Therefore, if starting from a non-default configuration, after upgrading to Version 11.0.6, it is recommended to review and verify the consistency and effectiveness of the BigFix ODBC data source configurations.

ODBC examples for Windows

Example of ODBC configuration since Version 11 Patch 6

This configuration code shows the 64-bit and 32-bit ODBC keys of a computer with the BigFix Server and Web Reports using a local database and the Microsoft ODBC Driver 18 for the connection.

```
Windows Registry Editor Version 5.00

[HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\ODBC\ODBC.INI]

[HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\ODBC\ODBC.INI\bes_bfenterprise]
"Database"="BFEnterprise"
"Driver"="C:\\Windows\\SysWOW64\\msodbcsql18.dll"
"Registration"="System Data Source"
"Server"="(local)"
"Trusted_Connection"="Yes"
"Encrypt"="Yes"
"TrustServerCertificate"="Yes"

[HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\ODBC\ODBC.INI\enterprise_setup]
"Driver"="C:\\Windows\\SysWOW64\\msodbcsql18.dll"
"Registration"="System Data Source"
"Server"="(local)"
"Trusted_Connection"="Yes"
"Encrypt"="Yes"
```

```
"TrustServerCertificate"="Yes"

[HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\ODBC\ODBC.INI\LocalBESReportingSer
ver]
"Database"="BESReporting"
"Driver"="C:\\Windows\\SysWOW64\\msodbcsql18.dll"
"Registration"="System Data Source"
"Server"="(local)"
"Trusted_Connection"="Yes"
"Encrypt"="Yes"
"TrustServerCertificate"="Yes"

[HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\ODBC\ODBC.INI\ODBC Data Sources]
"enterprise_setup"="ODBC Driver 18 for SQL Server"
"bes_bfenterprise"="ODBC Driver 18 for SQL Server"
"LocalBESReportingServer"="ODBC Driver 18 for SQL Server"

[HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBC.INI]

[HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBC.INI\bes_bfenterprise]
"Database"="BFEnterprise"
"Driver"="C:\\Windows\\system32\\msodbcsql18.dll"
"Registration"="System Data Source"
"Server"="(local)"
"Trusted_Connection"="Yes"
"Encrypt"="Yes"
"TrustServerCertificate"="Yes"

[HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBC.INI\enterprise_setup]
"Driver"="C:\\Windows\\system32\\msodbcsql18.dll"
"Registration"="System Data Source"
"Server"="(local)"
```

```
"Trusted_Connection"="Yes"
"Encrypt"="Yes"
"TrustServerCertificate"="Yes"

[HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBC.INI\LocalBESReportingServer]
"Database"="BESReporting"
"Driver"="C:\\Windows\\system32\\msodbcsql18.dll"
"Registration"="System Data Source"
"Server"="(local)"
"Trusted_Connection"="Yes"
"Encrypt"="Yes"
"TrustServerCertificate"="Yes"

[HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBC.INI\ODBC Data Sources]
"enterprise_setup"="ODBC Driver 18 for SQL Server"
"bes_bfenterprise"="ODBC Driver 18 for SQL Server"
"LocalBESReportingServer"="ODBC Driver 18 for SQL Server"
```

The "Trusted_Connection" registry value is set to "Yes", which means it is using Windows authentication. If that value is missing, the connection defaults to SQL Server authentication.

Example of ODBC configuration since Version 10 Patch 8 and up to Version 11 Patch 5

This configuration code shows the 64-bit and 32-bit ODBC keys of a computer with the BigFix Server and Web Reports using a local database and the Microsoft ODBC Driver 17 for the connection.

```
Windows Registry Editor Version 5.00

[HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBC.INI]

[HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBC.INI\bes_bfenterprise]
"Database"="BFEnterprise"
"Driver"="C:\\Windows\\system32\\msodbcsql17.dll"
```

```
"Registration"="System Data Source"
"Server"="(local)"
"Trusted_Connection"="Yes"

[HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBC.INI\enterprise_setup]
"Driver"="C:\\Windows\\system32\\msodbcsql17.dll"
"Registration"="System Data Source"
"Server"="(local)"
"Trusted_Connection"="Yes"

[HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBC.INI\LocalBESReportingServer]
"Database"="BESReporting"
"Driver"="C:\\Windows\\system32\\msodbcsql17.dll"
"Registration"="System Data Source"
"Server"="(local)"
"Trusted_Connection"="Yes"

[HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBC.INI\ODBC Data Sources]
"enterprise_setup"="ODBC Driver 17 for SQL Server"
"bes_bfenterprise"="ODBC Driver 17 for SQL Server"
"LocalBESReportingServer"="ODBC Driver 17 for SQL Server"

[HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\ODBC\ODBC.INI]

[HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\ODBC\ODBC.INI\bes_bfenterprise]
"Database"="BFEnterprise"
"Driver"="C:\\Windows\\SysWOW64\\msodbcsql17.dll"
"Registration"="System Data Source"
"Server"="(local)"
"Trusted_Connection"="Yes"

[HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\ODBC\ODBC.INI\enterprise_setup]
```

```
"Driver"="C:\\Windows\\SysWOW64\\msodbcsql17.dll"
"Registration"="System Data Source"
"Server"="(local)"
"Trusted_Connection"="Yes"

[HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\ODBC\ODBC.INI\LocalBESReportingSer
ver]
"Database"="BESReporting"
"Driver"="C:\\Windows\\SysWOW64\\msodbcsql17.dll"
"Registration"="System Data Source"
"Server"="(local)"
"Trusted_Connection"="Yes"

[HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\ODBC\ODBC.INI\ODBC Data Sources]
"enterprise_setup"="ODBC Driver 17 for SQL Server"
"bes_bfenterprise"="ODBC Driver 17 for SQL Server"
"LocalBESReportingServer"="ODBC Driver 17 for SQL Server"
```

ODBC on Linux

On Linux, the concept of an ODBC data source is more vaguely defined and database connection settings are set by interacting with the DB2 Client and driver, usually via the command line.

Configuring the number of Web Reports results

To avoid excessive use of memory when displaying Web Reports results on `Explore Computers` reports, you can configure the `MaxReportResults` keyword. This keyword sets the maximum number of rows that can be displayed in a web report. Its default value is `1000000`.

The valid value range of the keyword is 1-4294967295.

When you get the message:

```
Unable to update data table: server aborted or there was an error
 processing your request
```

on the report page and you see the exception:

```
Too many results returned from computer report. Report execution has been
 aborted
```

in the log files, indicating that the number of lines to be displayed exceeds the default value, tune the keyword value by taking into account the type of report, the computer properties, and the resources of the system where Web Reports is running.

You can set this keyword both on Windows and Linux systems where Web Reports is running, by completing the following steps:

On Windows systems:

Run **regedit** and locate the path `HKEY_LOCAL_MACHINE\Software\Wow6432Node\BigFix\Enterprise Server\BESReports`.

Create the string value (reg_sz) `MaxReportResults` and set it to the identified value.

```
"MaxReportResults"[REG_SZ] = "1000000"
```

On Linux systems:

Add the `MaxReportResults` keyword to the `[Software\BigFix\Enterprise Server\BESReports]` section of the `beswebreports.config` file and set it to the identified value:

```
MaxReportResults = 1000000
```

Managing Daylight Saving Time (DST) for Web Reports

By default, Web Reports track scheduled activity next attempt times using UTC rather than the local time zone.

So, when the local time zone changes, for example from daylight saving time to non-daylight saving time, reports run an hour earlier or later, depending on the direction of the change.

Starting from BigFix Version 9.5 you can prevent to run scheduled reports an hour earlier or later because of the change due to the daylight saving time by setting the `AdjustScheduleForDST` keyword to 1.

You can set this keyword both on Windows and Linux systems where Web Reports is running, by completing the following steps:

On Windows systems:

Run **regedit** and locate the path `HKEY_LOCAL_MACHINE\Software\Wow6432Node\BigFix\Enterprise Server\BESReports`.

Create the string value `AdjustScheduleForDST` and set it to 1.

```
"AdjustScheduleForDST" = "1"
```

On Linux systems:

Add the `AdjustScheduleForDST` keyword to the `[Software\BigFix\Enterprise Server\BESReports]` section of the `beswebreports.config` file and set it to 1:

```
AdjustScheduleForDST = 1
```

Important: The Daylight Saving Time setting becomes effective when the first event after the change is triggered.

FIPS 140-2 cryptography in the BigFix environment

BigFix uses the BigFix Cryptographic Module to perform cryptographic functions throughout its environment.

For instance, every time an operator logs into the BigFix console, creates a new user, initiates an action, or subscribes to new content there are cryptographic operations performed by this module.

The BigFix Cryptographic Module uses OpenSSL FIPS Object Module 3.0.8 that has been certified by NIST as compliant with the FIPS (Federal Information Processing Standard) 140-2 standard.

Note: The OpenSSL FIPS Object Module version for Windows ARM 64 is 3.0.9.

Configuring FIPS 140-2 on the BigFix Server

You can configure the BigFix server to use FIPS 140-2.

To verify the appropriate setup and initialization of the module you must check the client log file by completing the following steps:

1. On the BigFix server, launch the BigFix Administration Tool by selecting **Start > All Programs > BigFix > BigFix Administration Tool**.
2. Browse to the location of your site license and click **OK**
3. Select the **Masthead Management** tab.
4. Click **Edit Masthead**.
5. Check **Require use of FIPS 140-2 compliant cryptography** to enable FIPS 140-2.
6. Click **OK**.
7. Enter the Administrator password to perform the action.
8. To ensure that the setting has been enabled check the client log file (default log path: `C:\Program Files\BigFix Enterprise\BES Client\__BESData\__Global\Logs\YYYYMMDD.log` for the following types of messages:
   - **FIPS 140-2 Enable log file message**

     ```
     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 Disabled log file message**

     ```
     At 14:58:28 -0700 -
     FIPS mode disabled by default.
     Unrestricted mode
     ```

You can enforce the FIPS mode, by setting the `_BESClient_Cryptography_FipsMode` value on the client.

Note: The client setting **_BESClient_Cryptography_FipsMode** overrides the FIPS setting specified in the masthead for the BES Client and the Web Reports components. When setting the value to **none**, the BES Client and the Web Reports components will not use the FIPS libraries. When setting the value to **required**, they will use the FIPS libraries.

In this way the client does not run in FIPS mode when the Cryptographic Module encounters an error at startup.

To force BigFix components to use only the FIPS validated Cryptographic library, complete the following steps:

1. Launch the BigFix Console.
2. From the **Computers** tab, right-click any listed computer and choose **Edit Computer Settings**.
3. Click **Add**.
4. In the Add Custom Settings dialog enter: `_BESClient_Cryptography_FipsMode` in the **Setting Name** and `required` in the **Setting Value**.
5. Click **OK**.
6. In the **Target** tab select `All computers`. When FIPS mode is enabled all cryptographic operations such as digital signatures, encryption and SHA1, SHA2 hashing are performed using the FIPS 140-2 Level 2 certified cryptographic module.
7. In the **Execution** tab of the dialog choose **Reapply this action whenever it becomes relevant again** and click **OK**

Note: Up to BigFix version 11.0.6, to enable FIPS 140-2 on the BigFix Linux server, see the **`-advRequireFIPScompliantCrypto`** option described in Editing the Masthead on Linux systems *(on page 1119)*.

Note: Starting from BigFix version 11.0.7, to enable FIPS 140-2 on the BigFix Linux server, see the **`-advRequireFIPS_140_2_Crypto`** option described in Editing the Masthead on Linux systems *(on page 1119)*.

Note:

- The most common error related to the FIPS mode startup occurs on AIX systems when there is not enough system entropy available for the Cryptographic Module.
- The FIPS Mode setting and the Message Level Encryption (MLE) setting are independent. You can set FIPS without setting the MLE and viceversa.

For more information on Message Level Encryption, see Message Level Encryption (MLE) Overview (on page 1078) and Message Level Encryption and DSA (on page 118).

FIPS 140-3 cryptography in the BigFix environment

BigFix uses the BigFix Cryptographic Module to perform cryptographic functions throughout its environment.

For instance, every time an operator logs into the BigFix console, creates a new user, initiates an action, or subscribes to new content there are cryptographic operations performed by this module.

Starting from BigFix Platform Version 11.0.7, the BigFix Cryptographic Module uses OpenSSL FIPS Object Module that has been certified by NIST as compliant with the FIPS (Federal Information Processing Standard) 140-3 standard.

Note: BigFix Platform pre-11 components will no longer work if you enable FIPS 140-3.

Note: If you enable FIPS 140-3, BigFix Platform 11 components up to 11.0.6, will run in FIPS 140-2 mode.

Note: If you enable FIPS 140-3, BigFix Server requires at minimum the usage of the TLS 1.2 protocol with the Extended Master Secret (EMS) extension to successfully connect to HTTPS servers.

Configuring FIPS 140-3 on the BigFix Server

Starting from BigFix Platform Version 11.0.7, you can configure the BigFix server to use FIPS 140-3.

On Windows

To enable FIPS 140-3, perform the following steps:

1. On the BigFix server, launch the BigFix Administration Tool by selecting **Start > All Programs > BigFix > BigFix Administration Tool**.
2. Browse to the location of your site license and click **OK**
3. Select the **Masthead Management** tab.
4. Click **Edit Masthead**.
5. Check **Require use of FIPS compliant cryptography** ; it will enable the two mutually exclusive FIPS alternatives, with FIPS 140-2 selected by default.
6. Select the **Use FIPS 140-3 compliant cryptography** option.
7. Click **OK**.
8. Restart the BigFix server processes.

To ensure that the FIPS 140-3 mode has been successfully enabled, verify if the following messages are present in the `C:\Program Files (x86)\BigFix Enterprise\BES Server\BESRelay.log` file of the BigFix Server:

```
OpenSSL Initialized (FIPS Mode)
FIPS mode: 140-3 Compliant Mode selected.
```

For more information about how to manage FIPS with BigFix Administration Tool on Windows, see Editing the Masthead on Windows systems (on page 1114).

On Linux

To enable FIPS 140-3, perform the following steps:

1. On the BigFix server, move to the `/opt/BESServer/bin` directory.
2. Run the following BigFix Administration Tool command:

```
BESAdmin.sh -editmasthead -advRequireFIPS_140_3_Crypto=true
 -sitePvkLocation=/opt/iemlic/license.pvk
```

3. Enter the license.pvk password.
4. Restart the BigFix server processes.

To ensure that the FIPS 140-3 mode has been successfully enabled, verify if the following messages are present in the `/var/log/BESRelay.log` file of the BigFix Server:

```
OpenSSL Initialized (FIPS Mode)
FIPS mode: 140-3 Compliant Mode selected.
```

For more information about how to manage FIPS with BigFix Administration Tool on Linux, see Editing the Masthead on Linux systems (on page 1119).

Managing Client Encryption

Server and relay-bound communications from clients can be encrypted to prevent unauthorized access to sensitive information. To enable it, you must generate a key and provide a value on clients for the setting named `_BESClient_Report_Encryption`.

By default, the value for this setting is set to **optional**. The value is set in the console and is described in Enabling encryption on Clientsthe BigFix Installation Guide.

On Windows servers, the key is generated from the **Encryption** tab of the BigFix Administration Tool:

1. Launch the BigFix Administration Tool by selecting **Start > Programs > BigFix > BigFix Administration Tool**.
2. Select the **Encryption** tab. At the top of the dialog is a statement of the current state.

Client encryption has four states: Disabled, Pending, Enabled, and Pending Rotation:

Disabled

This state indicates that no encryption certificate is included in your deployment masthead, which means that Clients cannot encrypt their reports even if they are told to do so. Click **Generate Key** to create an encryption certificate (and the corresponding private key, which can be used to decrypt reports at the receiving end). The state is set to **Pending** state.

Pending

In this state, an encryption certificate has been generated and is ready for deployment, but the private key has not yet been distributed to all necessary decrypting relays and servers. When you have manually distributed the private key, click the **Enable Encryption** button to embed the certificate in the masthead and send it out to all clients. The state is set to Enabled. Click **Cancel** to return to the Disabled state.

Enabled

In this state, an encryption certificate has been found in your deployment masthead, which means that you are able to turn on encryption (using the setting discussed previously) for any of the clients in your deployment. At any time, you can click **Generate new key** to create a new encryption certificate. This is useful if you have a key rotation policy or if your encryption key is ever compromised (see next section). Generating a new key returns the state to Pending (unless you choose to deploy immediately as described in the next section). You can also click **Disable** to move back to the Disabled state.

Pending Rotation

In this state, an encryption certificate is included in your deployment masthead, and a new certificate has been generated and is ready to replace the existing certificate.

On Linux servers, you can encrypt clients by completing the following steps as super user:

1. Generate the key:

```
./BESAdmin.sh -reportencryption -generatekey -privateKeySize=max
 -deploynow=yes
-sitePvkLocation=<path+license.pvk> -sitePvkPassword=<password>
```

2. Activate the key:

```
./BESAdmin.sh -reportencryption -enablekey
 -sitePvkLocation=<path+license.pvk>
-sitePvkPassword=<password>
```

To list all the available options run:

```
./BESAdmin.sh -reportencryption -h
```

Generating a new encryption key

If your private key is compromised or if you have a policy of rotating keys, you can generate a new key from the **BigFix Administration Tool**.

1. Launch the BigFix Administration Tool by selecting **Start > Programs > BigFix > BigFix Administration Tool**.
2. Select the **Encryption** tab.
3. Click the **Generate key** button. The Create Encryption Credentials dialog opens.



4. From this dialog, select the key size. The default is 2048, which is adequate for most purposes. Check the box to use this key immediately. However, if you have established relays that use encryption, leave this box unchecked until you can distribute the new key to those relays.
5. Click **OK** to distribute this new key to your clients. You must provide your Site Administration Private Key to propagate the action. A final dialog asks for confirmation. For more information about encryption key sizes and server requirements, see the knowledge-base article on server requirements at the BigFix support site.

Creating top-level decrypting relays

When an action is deployed, thousands of clients might report back in a short time frame, typically to a relay. If you have chosen to encrypt these reports, the relay bundles the reports together and passes them to the server, which must then split up and decrypt each one of them.

With many thousands of clients, this can impose a significant computational burden on the server.

To improve performance, you can lighten the load on your server by allowing your top-level relays to do the bulk of the decryption. If you have over 50,000 clients, you might be able to substantially reduce the load on your server by moving decryption down into the relay chain. If the relay has its own decryption key, it can first decrypt the client messages into plain text and then bundle thousands of them into a single archive. This can then be compressed, encrypted, and passed to the server. At that point, the server can perform a single decryption on the entire archive, noticeably reducing its overhead.

To spread the decryption tasks, distribute your encryption keys to your top-level relays. For normal server-level encryption, HCL creates an encryption key for you and places it in the program folder:

On Windows systems:

```
%PROGRAM FILES%\BigFix Enterprise\BES Server\Encryption Keys
```

On Linux systems:

```
var/opt/BES Server/Encryption Keys
```

To allocate the load to your top-level relays, place the encryption key in the equivalent relay directory:

On Windows systems:

```
%PROGRAM FILES%\BigFix Enterprise\BES Relay\Encryption Keys
```

On Linux systems:

```
var/opt/BES Relay/Encryption Keys
```

These top-level relays decrypt all the documents received, bundle them together, and then re-sign them with a single signature. You can put as many keys as you want in the folder and the relay attempts to use each of them when it gets an encrypted client report. clients encrypt against the key found in the masthead file, which should be the last key created. However, it is possible that a client transmits a report with an older version of the masthead (and thus a different encryption key) if it has not gathered the latest actionsite for any reason.

When you use top-level encryption, consider the following best practices:

- You must manually transfer the key file from the server to the relay every time you create a new encryption key.
- During the transfer process, it is important not to expose your private key file. This means that you must not move the key over the internet because anyone listening might be able make a copy of your private key file. Instead, physically transfer the key from one computer to another, for example, with a USB key.
- During the encryption key creation process, you have the option to create the private key file, but not propagate it out in the masthead. This step gives you time to transfer the new key file to the relays before clients start posting encryption messages with that key.

Message Level Encryption (MLE) Overview

Message Level Encryption (MLE) allows your Clients to encrypt upstream data using a combination of an RSA public/private key-pair and an AES session key.

The RSA key-pair can be of 2048- or 4096-bit key length, with longer keys offering additional security, but requiring more processing power for decryption at the server. The AES session key uses the maximum FIPS-recommended length of 256 bits. You can configure your Relays to reduce the load on the Server by decrypting and repackaging the Client data before relaying it.

The RSA public key encrypts the session key and adds it to the AES-encrypted report. At the BigFix Server (or a decrypting Relay) the corresponding RSA private key is used to decrypt the AES session key, which is then used to decrypt the Client report.

There are three levels of report encryption:

Required

Clients require encryption of reports and uploads. The client does not report or upload files if it cannot find an encryption certificate or if its parent relay does not support receipt of encrypted documents.

Optional

Clients prefer, but do not require encryption of reports and uploads. If encryption cannot be performed, reports and uploads are done in clear-text.

None

Clients do not encrypt, even if an encryption certificate is present.

For more information about how to set encryption on Clients, see Enabling encryption on Clientsthe Installation Guide.

Requirements

To enable MLE, the BigFix Server will require additional CPU resources to process the encrypted client. Server hardware recommendations (for CPU) are as follows:

Deployment Size	CPU
250	2-3 GHz - 2 Cores
1,000	2-3 GHz - 2-4 Cores
10,000	2-3 GHz - 4 Cores
50,000	2-3 GHz - 4-8 Cores
100,000	2-3 GHz - 8-16 Cores
200,000	2-3+ GHz - 16 Cores

If your deployment is over 50,000 seats, or you are using an encryption key strength of 2048 or 4096 bits, BigFix highly recommends also configuring one or more decrypting top level Relays (with 2-4 CPU cores each) to help distribute the additional processing load.

Enabling Message Level Encryption

Windows Server:

1. Managing Client Encryption *(on page 1073)*.
2. Generating a new encryption key *(on page 1075)*

   Note: If you plan on leveraging top-level relays to decrypt incoming client data, make sure to uncheck "begin encrypting with this key" before clicking **OK**.
3. Deploy the Task in the Support Site called 'BigFix Client Setting: Encrypted Reports' (Task ID 543 in BigFix Support) to enable encryption on Clients, and select one of the encryption level options.

   OR by adding a custom setting.



Note: When disabling Message Level Encryption on Windows, the encryption goes into a Pending state. To definitely disable it, click the same function again.

Linux Server:

To enable Message Level Encryption on Linux, run the following steps as super user:

1. Generate the key:

```
./BESAdmin.sh -reportencryption -generatekey -privateKeySize=max
 -deploynow=no -outkeypath=<path>
-sitePvkLocation=<path+license.pvk> -sitePvkPassword=<password>
```

2. Activate the key:

```
./BESAdmin.sh -reportencryption -enablekey -deploynow=yes
 -sitePvkLocation=<path+license.pvk>
- sitePvkPassword=<password>
```



Note: Step 2 is not necessary when running step 1 using the option -deploynow=yes

Available options can be retrieved from command help:

```
./BESAdmin.sh -reportencryption -h
```



Note: When disabling Message Level Encryption on Linux by running the command

```
./BESAdmin.sh -reportencryption -disable
 -sitePvkLocation=/tmp/license.pvk
```

the encryption goes into a Pending state. To definitely disable it, run the same command again.

Enabling MLE in a DSA Server Setup

To enable a MLE in a DSA Server Setup:

1. You will need to transfer the encryption key to the DSA BES Server. For more information, see Message Level Encryption and DSA *(on page 118)*. Transferring of the encryption key file must be done securely (for example, with a USB key). Should

the key be exposed or compromised, a new encryption key can be generated using BigFix Admin.

2. Once the encryption key has been copied to the DSA BES Server, execute BESAdmin on the DSA Server: select the **Encryption** tab, and click **Deploy key**.
3. Click **OK**.

Enabling Decrypting Relays (Optional)

You can enable a Relay to decrypt data and pass the decrypted data to the Server. This is a useful way to offload CPU load from the main Server to a relay, but it complicates the MLE setup (which is otherwise very simple). Additionally, if you have a decrypting relay when doing MLE, the decrypting relay decrypts all the client reports, then re-encrypts as a single data and forwards it to the core server.

Generally you will not need to use a decrypting relay unless you have many tens of thousands of agents or if your main Server CPU load is too high.

For more information about decrypting relays, see Creating top-level decrypting relays (on page 1076).

To enable a decrypting relay:

1. You will need to transfer the encryption key to the decrypting top-level relays before enabling MLE. The key can be found on the main Server with a default location of:

   "Program Files\BigFix Enterprise\BigFix Server\Encryption Keys\SHA1HASH.pvk"

   and should be copied on the decrypting relays to:

   "Program Files\BigFix Enterprise\BigFix Relay\Encryption Keys"
2. Since this is a private key, BigFix recommends transferring this file securely (for example, with a USB key). Should the key be exposed or compromised, a new encryption key can be generated using BigFixAdmin.
3. Once the encryption key has been copied to all the decrypting top level Relays, click the **Enable** button in the **Encryption** tab.

Changing the Client Icon

By default, the icon in the upper left corner of the client UI is the BigFix logo.

This same icon is shown in the tray when an action is pending and in the task bar when the program is running. You can change this icon to help you clarify to your users who is the source of the action, and also to comply with corporate branding and trademark requirements. Follow these steps to change the icon:

- On Windows systems:
  1. Run the BigFix Administration Tool from **Start > Program Files > BigFix > BigFix Administration Tool**.
  2. Click **System Options** tab.
  3. Click **Add Icon** and use the **Open** dialog to browse for your icon (.ico) file.

On Linux systems:

1. Identify the path of the new icon, for example: `/IEM/newicon.ico`.
2. From the `/opt/BESServer/bin` command prompt, start the command line:

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

3. From the `/opt/BESServer/bin` command prompt, run the following command:

   ```
   ./iem post /IEM/newicon.ico admin/icon
   ```

   where: */IEM/newicon.ico* represents the full path of the new icon and `admin/icon` is the parameter to use to upload the new icon.

The icon is propagated to the clients, but it is not incorporated into the interface until the client restarts. After that, when a client interface opens (in response to an action, a dashboard or an offer), it includes the graphic icon you specified.

Client UI main actions

The BigFix Client UI is used to display message boxes to the end users logged on to the client computer (including pre-action messages, action running messages, and restart/ shutdown messages).

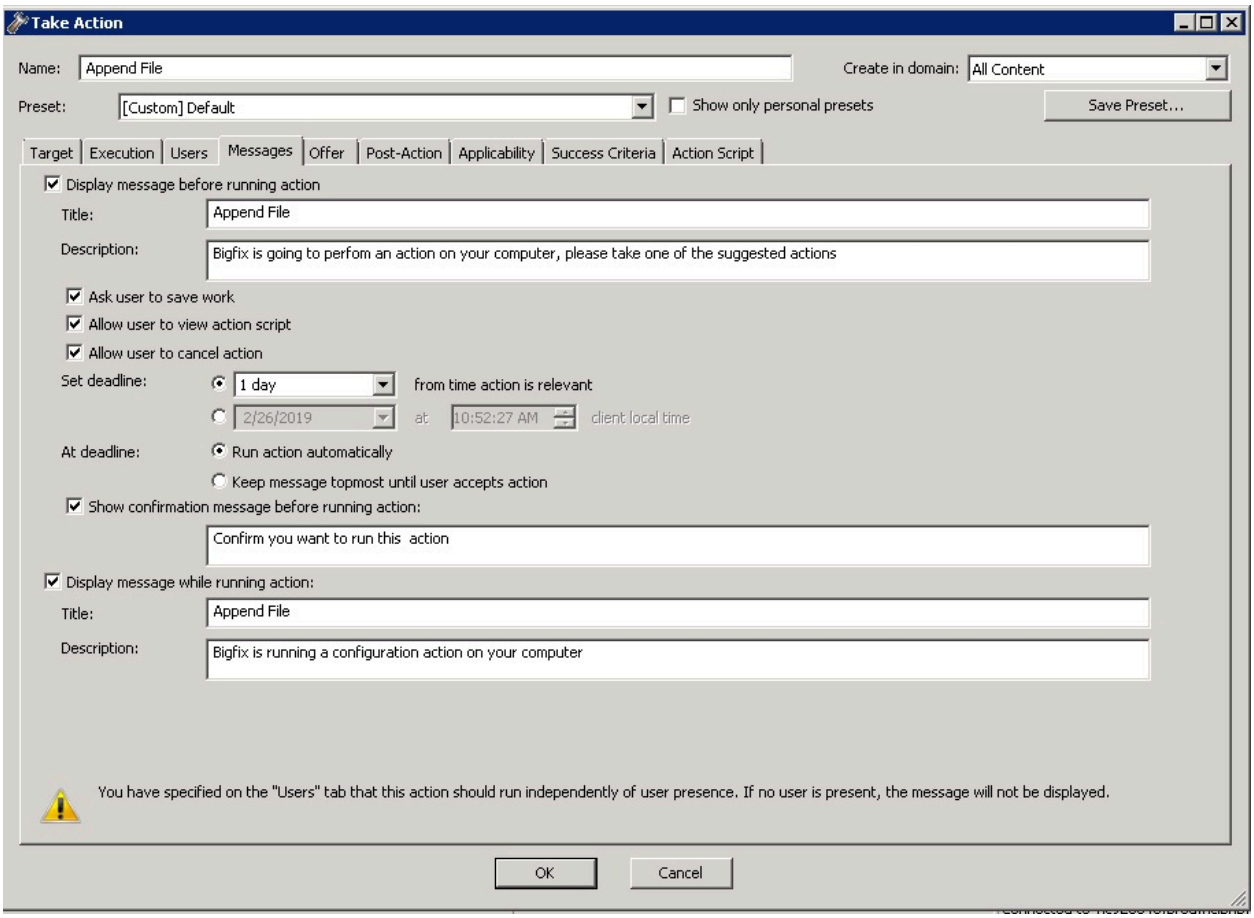
If the BigFix Client UI is not running in the user session, the end user will not see any BigFix messages.

The BigFix client UI displays messages in the client notification area, for example the task bar on Windows.

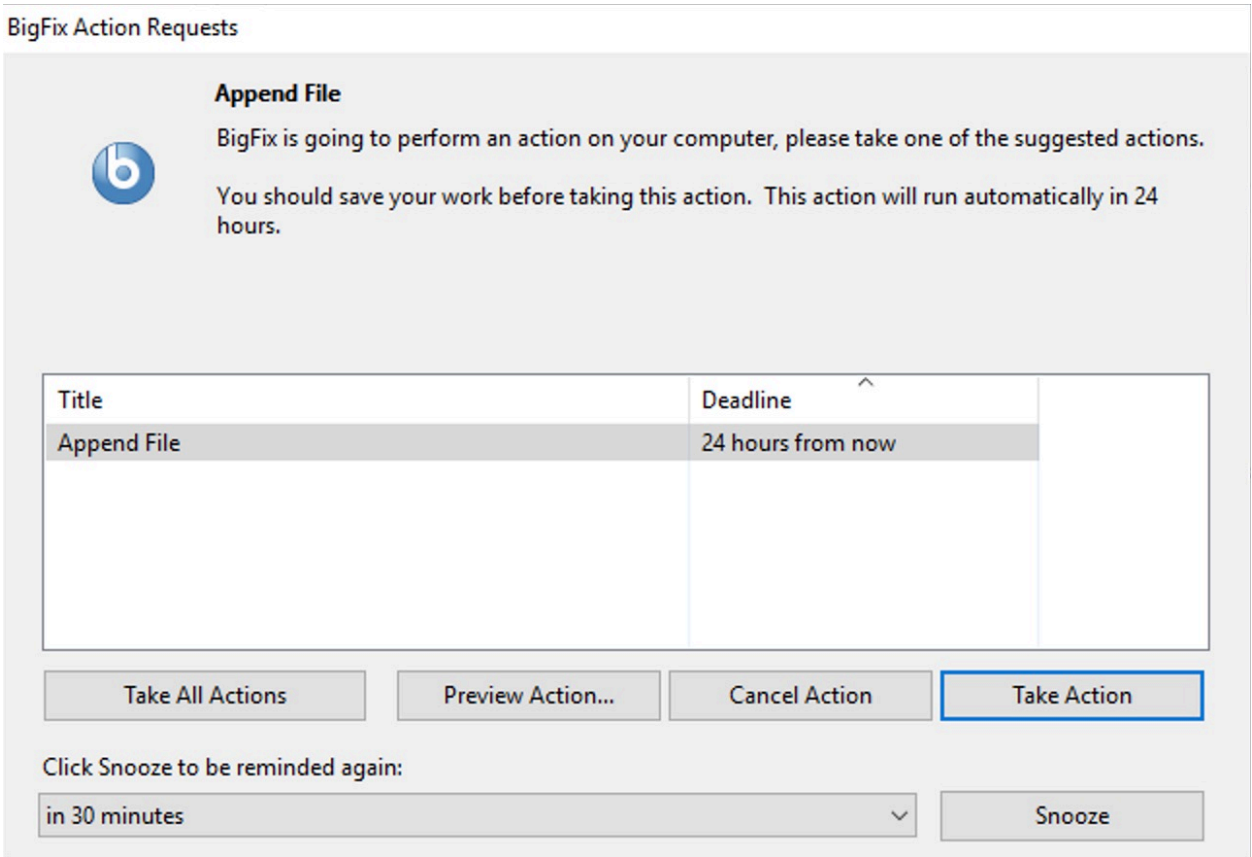
This topic describes in particular what the Client UI shows on the client computer when the **Messages** and **Offer** tabs of the **Take Action** panel are triggered from the BigFix Console.

Scenario 1 (Messages tab)

If the BigFix Console operator specifies the following settings in the **Messages** tab of the **Take Action** panel:



The screen capture below represents what an end user will see on the client computer:



The main actions that can be performed here by the end user are:

Take Action / Take All Actions

Performs the action on the client computer.

Preview Action

Displays the action script contained in the action.

Cancel Action

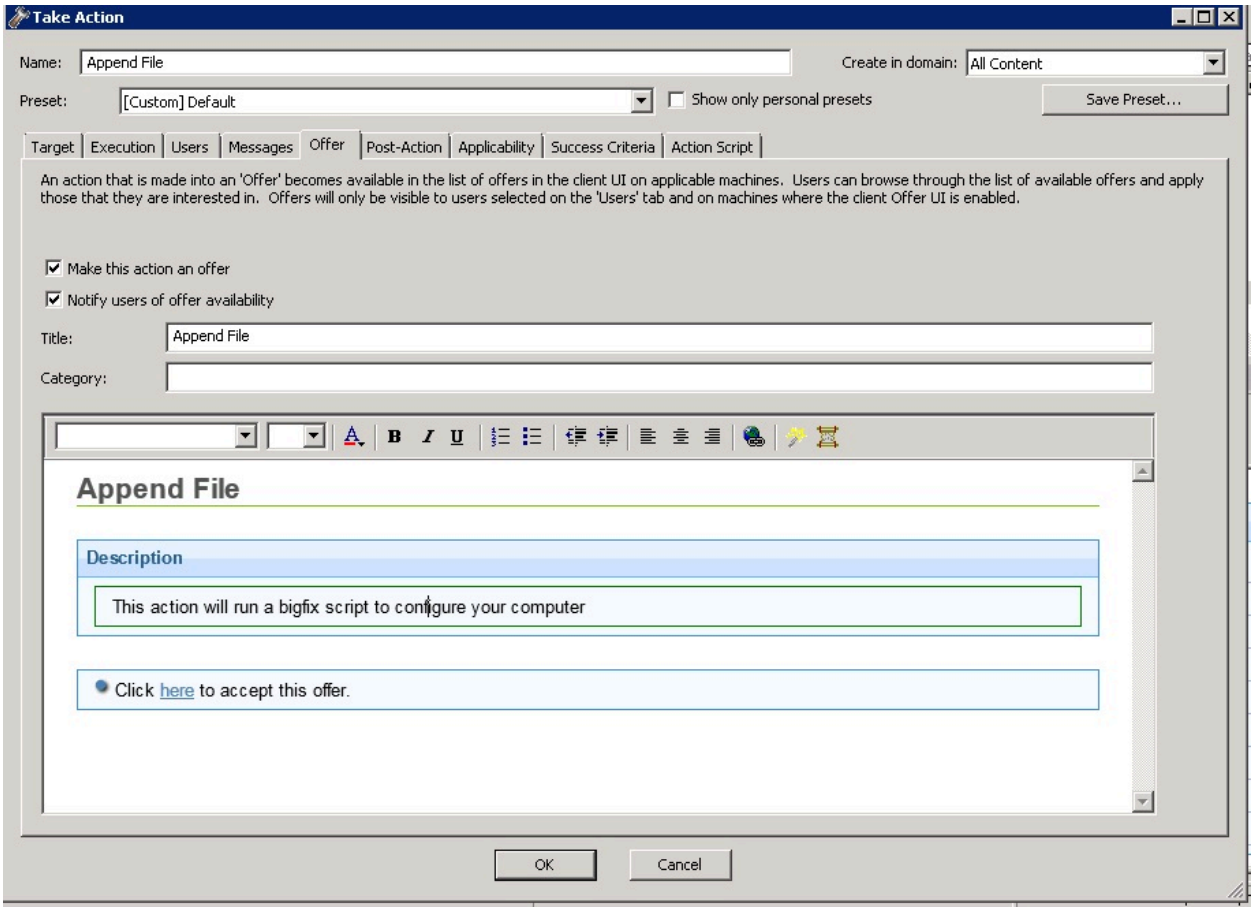
Deletes the action.

Snooze

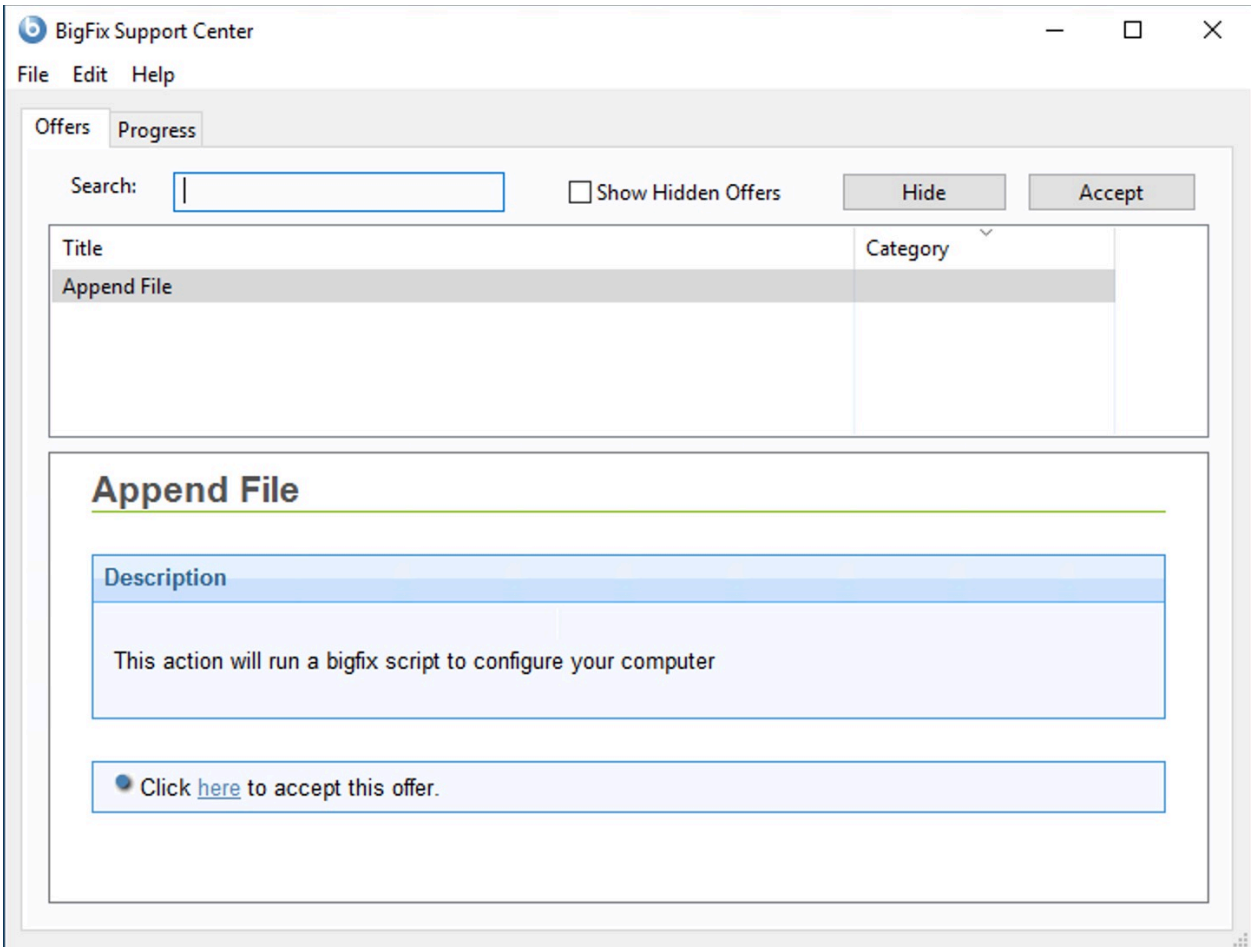
Postpones the action and allows you to choose when to be reminded about the action to perform.

Scenario 2 (Offer tab)

If the BigFix Console operator specifies the following settings in the **Offer** tab of the **Take Action** panel:



The screen capture below represents what an end user will see on the client computer:



The main actions that can be performed here by the end user are:

Hide

Hides the client notification panel.

Accept

Performs the action on the client computer.

Note: The main actions can be performed on the **Offers** tab, while the **Progress** tab shows you the status of all prior activities (previously received offers).

Optimizing the servers

BigFix operates efficiently, with minimal impact on network resources. However, there might be installations that stretch the recommended configurations, where there are too many clients for the allotted server power.

The best solution is to choose a server with the required characteristics for your environment; you might be able to modify some preferences to get better performance. Most of these optimizations involve a trade-off between throughput and responsiveness, so proceed with caution. Your HCL Software Support has more information about which modifications might be best for your particular deployment.

Here are some possible optimization techniques:

- Deploy **Relays** to reduce the load on the server. This is the most effective way to increase the performance and responsiveness of BigFix. Generally, the more relays, the better the performance (as a rule of thumb, one relay for 500 to 1000 clients is a good choice, although it can be much higher for a dedicated computer).
- Slow down the **Client heartbeat** from **File > Preferences**. This decreases the frequency of messages that are regularly dispatched by the clients to update their retrieved properties. Reducing this frequency reduces the amount of network traffic generated, but also decreases the timeliness of the retrieved properties. However, regardless of the heartbeat settings, the clients always send their latest information whenever they receive a refresh ping from the server or when they notice that a Fixlet is relevant.
- Slow down the **Fixlet List Refresh** rate from **File > Preferences**. This decreases the update frequency for the information displayed in the console. If there are many clients or consoles simultaneously connected or the database is very large, reducing this frequency can substantially reduce the load on the server. If multiple console operators are going to be simultaneously using the console, set the refresh rate to be something higher than the default (15 seconds) to reduce the load on the BigFix database. Consider changing it to 60-120 seconds or more if there are many console operators. The BigFix Administration Tool on the server allows you to set a global minimum refresh rate.

- Your database administrator might be able to help you with the following optimizations:
  - Change the SQL server Recovery Model for the BFEnterprise database to **Simple transaction logging**.
  - Reduce the percentage of memory allocated to SQL server from 100% to 85%, to ensure that the web server and operating system are not short of memory.

More performance recommendations can be found at the BigFix support site.

Optimizing the consoles

To be responsive, the console requires reasonable CPU power, memory, and cache space.

If you have a console that is taking a long time to load or that is performing sluggishly, there are several techniques you can use to speed it up:

- **Make sure you have sufficient memory**. The BigFix console benefits greatly from capacious memory to speed up the viewing, filtering, and sorting of content (Fixlet messages, tasks, actions, and so on). If your computer does not have enough physical memory, the console will run noticeably slower. You can check memory usage from the Task Manager (Ctrl-Shift-ESC). Select the Performance tab and refer to the Physical Memory section. If the available memory is less than 10% of the total memory, you are running low on RAM and can benefit by adding more.
- **Use high-speed network connections** between your consoles and servers, preferably with LAN connections of at least 100 MBPS. The BigFix Database can be sizeable for a large network, so running the console from a computer with a slow connection often results in very long load times.
- **Use remote control software**. With so much data to load and display, operating the console in a remote office over a slow link can be tedious. In situations like this, you might be able to benefit from solutions such as Citrix, Terminal Services, or other remote control software. Set up the remote control server on a computer with fast access to the server. Allow that machine to present instances of the console and have the branch office run these consoles remotely. The database stays in the main office,

and the remote office has optimal performance. For more information, see Remote Citrix / Terminal Services Configurationthe *BigFix Installation Guide*.

- **Delete old actions**. The BigFix database stores information about old actions, which the console loads in at startup and saves out at shutdown. If you do not need to track these old actions, you can delete them, allowing the console to load and close faster. Note that deleted actions continue to exist in the database, but are not loaded into the console or Web Reports and can be undeleted if necessary.
- For more information about how to enhance performance, see Performance Configurations.

Managing Bandwidth

File downloads consume the bulk of the bandwidth in a typical installation. You can control the bandwidth by throttling, which limits the number of bytes per second.

You can specify the bandwidth throttling on either the server, on the client, or on both (in which case the lower of the two values is used). This can be important whenever you have bandwidth issues, as in the following situations:

- A remote office with a thin channel
- Remote dial-in users or users on a slow connection
- A shared channel with higher-priority applications
- A WAN or LAN that is already saturated or has stringent load requirements

Bandwidth throttling settings (and other relay, server, and client settings) can be set using the tasks from the Support site. Select the **BigFix Management** domain and select the **BES Component Management** node in the navigation tree to see the entire task list.

Bandwidth throttling

Many network environments have limited bandwidth between certain geographic locations, between offices and home users using VPNs, and so forth. Deploying large Microsoft® patches (for example: the Windows 7 SP1 update is 537 MB!) can overwhelm limited bandwidth connections and cause bandwidth problems for certain users or applications.

To avoid issues related to bandwidth usage, BigFix uses the following mechanisms:

- Properly implemented and maintained Relay architecture
- Distributing patch deployments over time
- Applying bandwidth throttling configurations between components

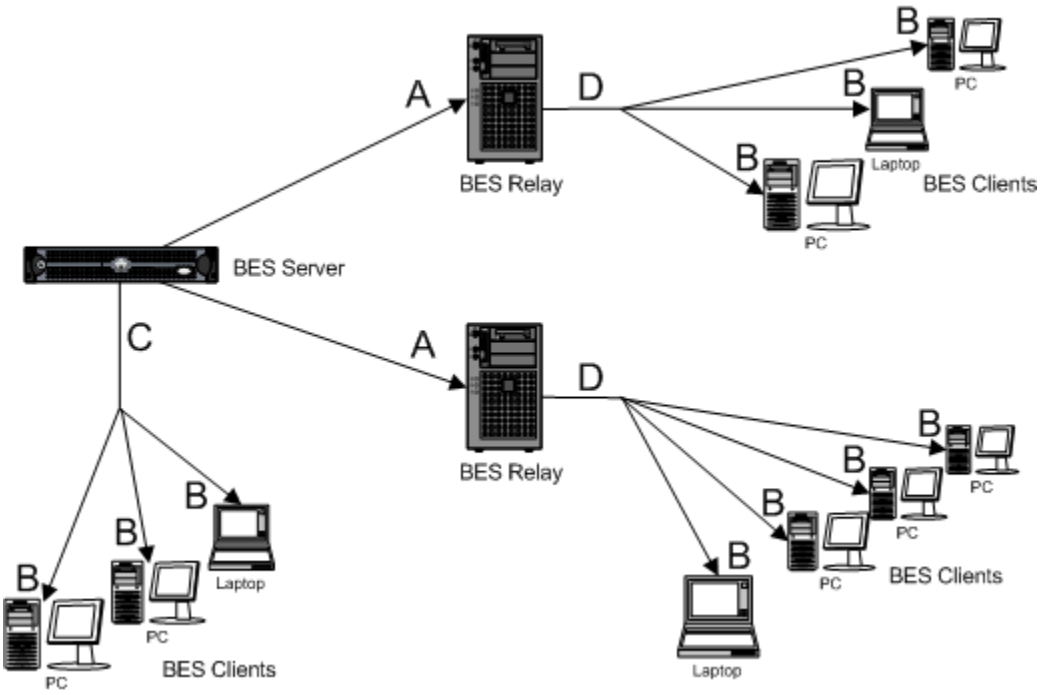
This section explains how to configure the BigFix components to use bandwidth throttling. The BigFix Console operator can control the maximum number of bytes per second that is used to send files over a network connection through client configuration settings and it can be done at the levels of the BigFix Server, Relay, or Client components.

You can configure the bandwidth throttling settings through tasks within the BES Support site. On the console, click **All Content > Fixets and Tasks > Tasks Only > By Site > BES Support** and search for the term throttle. All settings are set as registry key client settings on the endpoint in the client section of the registry, that is, `HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\BigFix\EnterpriseClient\Settings\Client` and with a `String [REG_SZ]` type of named value. The value is entered as a numeric; for example, 500 bps is entered as 500.

For details about the related configuration settings, see Bandwidth throttling (on page 818).

Throttling methods

The following diagram helps you determine the settings you can use to configure bandwidth throttling for a specific segment type.



Throttling through Relays (A)

Relays downloading from the BigFix Server

BigFix Relays can be configured to throttle file downloads when downloading from the BigFix Server. When BigFix Relay throttling is enabled, it downloads from the BigFix Server no more than the specified number of bytes per second. This setting is especially useful for Relays that have a slow connection to the Server (such as a Relay in a remote location).

You can configure client throttling by using the **_BESGather_Download_LimitBytesPerSecond** setting on the Relays. For details on how to configure this setting, see Bandwidth throttling (on page 818).

Total outgoing download traffic for the BigFix Relay

The BigFix Relay can be configured to throttle the cumulative file downloads anytime. When this throttling setting is enabled, a Relay sends out no more than the specified number of bytes per second for all file downloads (including Clients and child Relays). This setting is especially useful if there is a concern in a local area network that too much bandwidth is used when a patch is sent simultaneously to many clients.

Relay cumulative download throttling can be configured by using the **_BESRelay_HTTPServer_ThrottleKBPS** setting on the Relay. Set this number to the total number of kilobytes that the Relay gives to all of the clients combined per second.

Throttling through BigFix Clients (B)

Clients downloading files from the Server or Relay

BigFix Clients can be configured to throttle file downloads when downloading from the BigFix Server or BigFix Relays. When BigFix Client throttling is enabled, a client downloads from the Server or Relays at no more than the specified number of bytes per second. This setting is especially useful for individual computers that are on slow connections (such as traveling sales representatives or home users on dial-ups).

Client throttling can be configured by using the **_BESClient_Download_LimitBytesPerSecond** setting on the Clients. For details on how to configure this setting, see Bandwidth throttling (on page 818).

Throttling through the BigFix Server/Relays (C and D)

Total outgoing download traffic for the Server/Relays

The Server can be configured to throttle the cumulative file downloads at any given time. When this throttling setting is enabled, the Server/Relays send out no more than the specified number of bytes per second for all file downloads (including Clients and child Relays). This setting is especially useful if there is a concern in a local area network that too much bandwidth will be used when a patch is sent out to many Clients simultaneously.

BigFix Server cumulative download throttling can be configured by using the **_BESRelay_HTTPServer_ThrottleKBPS** setting on the Server. For details on how to configure this setting, see Bandwidth throttling (on page 818).



Note: For any changes to the configuration settings to take effect, you must restart the BES Root Server or BES Relay services as appropriate.

Throttling options

BigFix has a great number of settings that control how different components throttle traffic between one another. Throttling is always done for specific components, and some components might not have throttling available.

Throttled Components

These are any files that are downloaded directly from the `wwwrootbes` directory of the Relay, Server, or WebReports. Files downloaded for actions (for example, patches, service packs, and so forth) are downloaded this way. The contents of Fixlet sites are also downloaded this way during the gather interaction (although the site directory listing comes in separately).

Un-throttled Components

There are however, a number of "throttle-like" capabilities around client registration. The amount of downstream UDP traffic generated can be rate limited on the Relay/Server side, the frequency of client registration can be turned down on the client side, and the amount of ICMP traffic generated during relay selection can be rate limited on the client side. Note that client registration is not bandwidth intensive, although it can be a significant source of load. Download request plugin, status reporting plugins, the "Web Reports" plug-in, etc. For the Client, we expect this traffic to be negligible. However, things like the download status report generate a lot of traffic between the Console and the Root Server, and they cannot be throttled. When the Client asks the Relay "please tell me the latest contents of site X" (logged as 'GatherActionMV command received. Version difference, gathering'), the interaction is not throttled. The response of the Relay is typically small (anywhere from 0-~40k). If absolutely necessary, you can turn down the gather intervals on Clients to get this information less frequently, but this traffic should usually be negligible.

There are also things you can do to control the amount of traffic going upstream through the "PostResults" interface. The most basic "throttling" mechanism is simply to turn up the minimum report interval on the Client or lengthen the heartbeat interval. You can also put a crude limit on the amount of traffic a Relay can send up through a combination of the "ResultSizeLimit" and "ResultTimeLimit" Relay settings. However, you should check with Customer Support before doing so: it's unlikely that you'll get the behavior you expect.

Posting results takes more bandwidth than client registration, but is still much smaller than the amount of bandwidth used by the download and upload components. For most deployments, the amount of traffic should be negligible.

Throttling Minimum Transfer Rates

During throttled communication, the BigFix Client will send chunks of data and then wait longer than necessary before sending the next chunk. The BigFix Client can vary the amount of data in each chunk along with the amount of time to wait. By lowering the amount of data per chunk and maximizing the amount of data between chunks, a minimum transfer rate will be established for throttling.

The BigFix Relay component actually does the throttling work so if you need to upgrade BigFix to change the minimum throttling rate, it is important to upgrade the BigFix Relays along with the BigFix Clients to the newer version.

The effective minimum throttling rate is 100 Bytes/Sec.

All throttling settings are given in either BPS (Bytes per Second) or KBPS (KBPS). Given the effective minimum values listed above, if you set the throttling rate below the effective minimum it doesn't go that low, it only goes to the effective minimum value. Also notice that the effective minimum is less than 1 KBPS but the minimum value you can set KBPS throttling setting to is only 1 so you can't reach the effective minimum. IE., in some cases the effective minimum rate is controlled by the setting and in other cases the effective minimum is controlled by the client's throttling limits.

The settings used to specify the chunk size are:

_BESClient_UploadManager_ChunkSize

The value is specified in bytes (Default: 131072, that corresponds to 128 KB).

"0" means that the upload is done in a single chunk.

In a conflict between this computer and the upstream computer, the size of the chunk is set to the smaller of the two.

_BESRelay_UploadManager_ChunkSize

The value is specified in bytes (Default: 131072, that corresponds to 128 KB).

"0" means that the upload is done in a single chunk.

In a conflict between this computer and the upstream computer, the size of the chunk is set to the smaller of the two.

Upload Throttling

The uploads generated by the "Upload Manager" can be throttled from the Client side or from the Relay/Server side. This component only supports static throttling. There is a "PostFile" setting that sets an overall throttle on incoming connections, and an "UploadManager" setting that sets the throttle on outgoing connections (there is only one outgoing connection at a time). When both are set, the child is responsible for using the lesser of the two values.

_BESRelay_PostFile_ThrottleKBPS

"0" means "no limit" (Default: 0)

This setting is not sufficient to throttle PostFile, indeed it needs _BESClient_UploadManager_ThrottleKBPS to be set on Client with a non-0 value to work. The limit will be the lesser of the two values.

At the start of an upload interaction, the PostFilePlugIn divides this number by the total number of uploads currently in progress on the Relay and sends the result down to the child, who is responsible for respecting the resulting limit.

Same setting on both Server and Relay

_BESRelay_UploadManager_ThrottleKBPS

Default: 0

Only relevant on Relay (Server has no one to upload *to*, although DSA may change that at some point)

When Relay uploads files to its parent, it will limit itself to this rate. It provides the limitation by breaking the interaction into chunks and doing a connection for each chunk, with waits in between (this in contrast to download throttling, which maintains a single connection over the length of the interaction).

"0" means "no limit"

_BESClient_UploadManager_ThrottleKBPS

Default: 0

Only relevant on Client

When Client uploads files to its parent, it will limit itself to this rate. It provides the limitation by breaking the interaction into chunks and doing a connection for each chunk, with waits in between (this in contrast to download throttling, which maintains a single connection over the length of the interaction).

"0" means "no limit"

These are files uploaded from the endpoints through the "Upload Manager".

Two major axes of download throttling settings

"Server Side" or "Client side"

Server-side throttling (Server/Relay/Web Reports) is expressed as an amount of bandwidth to be shared among all connecting children

Client-side throttling (Client/Relay) is expressed as an amount of bandwidth to be used on a single upstream connection. Note that Clients may use more bandwidth if they have multiple simultaneous upstream connections.

Static throttling

Server-side and Client-side static throttling settings are described.

Server Side

Server-side static throttling settings control the total amount of download traffic that a server will send out to Clients using static throttling. The amount of bandwidth allocated to any given write connection is simply the "ThrottleKBPS" setting divided by the number of active write connections. Note that plug-in connections do not count as "write" connections.

However, file downloads with static throttling enabled do count as "write" connections.

Note: Server-side settings are in KBPS. For Relay and Root Server:

- _BESRelay_HTTPServer_ThrottleKBPS
    - Default: 0
    - "0" means "no limit"
- For Web Reports:
    - _WebReports_HTTPServer_ThrottleKBPS
    - Default: 0
    - "0" means "no limit"

Client Side

Client-side static throttling is the simplest of these settings. A "Client" (could be a Client or a Relay) simply tells its parent "please send me files at this speed" and the parent obliges. Settings are in BPS. For a BES Client:

- _BESClient_Download_LimitBytesPerSecond
    - Default: 0
    - "0" means "no limit"

For a BES Relay downloading files from its parent:

- _BESGather_Download_LimitBytesPerSecond
    - Default: 0
    - "0" means "no limit"

Throttle Groups

"Throttle Groups" are a part of the static throttling functionality that allow a set of Clients reporting to the same Parent Relay to throttle themselves as a group instead of individually (or along with every other connection through Server-side throttling).

"Throttle Groups" across Clients that report to different Parent Relays is not supported.

When a Client identifies itself as part of a "throttle group", it sends up the name of the group it belongs to, along with the speed it would like the entire group to have. So for example a Client might say "I am in the 'remote' group, and we would like to be given 10000 BPS as a whole". When the Server sends data down to that Client, it throttles based on the total

number of connections in the Clients group. So if there are five active connection from the 'remote' group, our Client will get 2000 BPS. Note that different Clients can send up different values for the "limit bytes per second", so another Client could say "I am in the 'remote' group, and we would like to be given 5000 BPS as a whole", and it would be given 1000 BPS at the same time our first Client was given 2000 BPS. The special group "ipaddress" will cause the Server to group this connection along with other connections from the same IP address. This is the default for Relay upstream traffic. Clients default to the group "", so that their "LimitBytesPerSecond" setting is shared among all of their currently active file downloads.

- _BESGather_Download_ThrottleGroup (valid on Windows Server only)
  - Default: "ipaddress"
  - The parent will consider this Relay to be part of whichever group is specified here
- _BESClient_Download_ThrottleGroup
  - Default: computer id as string
  - The parent will consider this Client to be part of whichever group is specified here
  - This is a string value: older versions of ClientSettings documentation incorrectly claimed this was a numeric value.

Managing Downloads

BigFix uses several methods to ensure that downloads are efficient and make the best use of available bandwidth. Among other techniques, caching is used extensively by all the BigFix elements, including servers, relays, and clients.

When an action on a client runs a download file command, the existence of the file is checked first in the client local cache. If the client cannot find it locally, it requests the file from its parent, typically a relay. In turn, the relay checks its own cache. If it finds the file, it immediately sends it down to the requesting client. Otherwise, it passes the request up to its parent, which might be another relay and the process continues. Ultimately, a server retrieves the file from an internal server or the Internet, caches it, and then passes it back

down the chain. After receiving the file, each relay in the chain caches it, and continues to forward it down to the original client, which also caches it.

If the agent runs the download now command while performing the action, the file is requested and collected from the URL specified in the action script.

Each cache retains the file until it runs out of space. At that point, the cache is purged of the least-recently used (LRU) files to provide more space. You can view the relay cache size and other relay information by activating the **Analysis ID# 227 BES Relay Cache Information** available from the BES Support Site. The default cache size is 1 GB, but you can change it by using the **Task ID# 148 BES Relay/Server Setting: Download Cache Size**, also from the BES Support site.

There might be situations that require files to be manually downloaded and cached, typically because such files are not publicly available, in which case you must download the files directly from the source. Review the **Fixlet Description** tab for more information about specific manual cache requirements. You can pre-populate the download cache by copying files to the download cache location `__Download`. You can also delete these files manually.

The caches are stored as sub-folders of the program folder, which is created by default at `%PROGRAM FILES%\BigFix Enterprise` on Windows systems, and `/var/opt/BES Server` on Linux systems. The server download cache is `BES Server\wwwrootbes\bfmirror \downloads\sha1`, and the client download cache is found at `BES Client\__BESData \__Global\__Cache\Downloads`.

As well as the download cache, relays maintain an action cache (also 1 GB) holding all the files needed for each Action, and clients maintain a Utility cache.

For information about troubleshooting relays, including bandwidth and downloading, see Relay Health.

The client collects the file by requesting it from the url listed in the action script in one of the following ways:

- When the complete set of downloads can be computed by parsing the action script, the complete set of downloads is computed by the server. The agent can ask the relay with a single request if the prefetch downloads are available for a specific action. In this request, the agent sends up the action ID, and the server response indicates

all the files are available, or they are not. If these are all available, the agent starts requesting the files by their ordinal number (1 indicates the first file in the script, 2 indicates the second file in the script, etc.). If the files are not available, the relay informs the agent they are not and begins the process of fetching them, and the agent notifies that it is waiting for downloads to be available and put itself into a pending downloads state for that action for 10 minutes, at which time it asks the relay again, if the downloads are available for the specific action.

When the downloads for an action become available on a relay, a notification is sent to the children of the relay, which uses the notification to accelerate requesting the downloads again. If notification messages are blocked for any reason, the agents 10 minute 'ask the relay again' behavior will eventually result in the agent detecting that the downloads are available, and begin to collect them. Child relays are also notified by their parent when the downloads based on the action ID and the ordinal numbers become available. They use this notification to accelerate their own request for the downloads again.

- For downloads where any of the download url, size, and hash value are listed in the action script such that only the agents can compute them, the agents query their parent relay using an itemized downloads available request. The request contains a list of download items the particular agent needs. The relay and client behave the same way as described above, delaying subsequent requests, waiting for notifications

Resuming a download

If the download fails for connection problems, the download process is resumed as follow:

- If the client is downloading from a BigFix Relay or Server, the download can be restarted at 10,000 byte chunks. This means that, when the client process is restarted, it verifies the 10,000 byte blocks already received, and then it resumes the download after the last verified block.
- If the client is running a direct download from another server's URL, when the client process restarts, the download starts again from the beginning.

Downloading directly from the Internet site

In addition to the existing client settings for Download Direct, starting from Version 10 Patch 1, you can configure your clients to download specific resources directly from the site where they are located, to mitigate the network impact and bandwidth requirements for relays serving VPN-based clients.

You can specify that all resource requests to a specific set of domains must be downloaded directly from the Internet and not from the relay. Use the client setting named **_BESClient_Download_Direct_Domainlist** to specify the list of domains for which the direct download is desired.



Note: If the PeerNest is enabled, the behavior remains unchanged. The resource is still requested from the peer URL.



Note: This setting should be used as an alternative to using **_BESClient_Download_Direct**. The **_BESClient_Download_Direct** setting, in fact, will force all resources to be downloaded directly from the Internet, regardless of the domain specified.

If the direct download from the Internet fails, you can specify that the clients attempt to download the file from the BigFix Relay or Server. Use the client setting named **_BESClient_Download_DirectRecovery** to enable this behavior.



Note: The setting has effect only if the client settings **_BESClient_Download_Direct** or **_BESClient_Download_Direct_Domainlist** are enabled.

For more details about these settings, see Download (on page 766).

Enable Direct Download based on network

Starting from Version 10 Patch 7, a new feature enables you to allow the Direct Download only for BigFix Clients connected to a specific subnet.

You can specify the list of subnets that allow the Direct Download with the new setting **_BESClient_Download_Direct_SubnetList**. The setting accepts only subnets specified in CIDR notation format, for example: `192.1.77.0/25;192.1.0.0/16`.

In case of computers with multiple network interfaces, the subnet considered when checking the allowed list is the subnet of the IP Address connected to the BigFix Relay.

In case of direct downloads in progress, if the Client reregisters using a new IP address that does not belong to any of the subnets in the list, then the Client interrupts the ongoing download.

When the Client interrupts the direct download in progress, the following error is logged:

```
The direct download (Action <action_id>) was canceled after Relay Select:
the address connected to the relay is changed.
```

Note: The **_BESClient_Download_Direct_SubnetList** setting provides added value to all situations in which a Direct Download is expected, for instance when the **_BESClient_Download_Direct** setting is enabled or the URL belongs to the **_BESClient_Download_Direct_Domainlist** setting.

For more details about this setting, see Download (on page 766).

Restart download after Relay switch

Starting from Version 10 Patch 7, a new feature allows you to interrupt the download in progress on a Relay switch. By default, if a BigFix Client moves to a new relay while a download operation is in progress (from the former relay), the file download continues from the former relay, if that is still reachable by the Client.

Only if the former relay is no longer reachable, the download fails, and a new download is attempted from the new Relay.

Enabling the new setting named **_BESClient_Download_ResetOnRelaySwitch** allows you to stop the download from the former relay, even if it is still reachable and, then, restart the download from the new Relay.

In the BigFix Client log file, after the relay switch, when the Client interrupts the download in progress from the former Relay, the following error is logged:

```
The download from Relay (Action <action_id>) was canceled after Relay
 Select:
the relay is changed.
```

For more details about this setting, see Download (on page 766).

Automatic URL redirection from HTTP to HTTPS

In addition to the existing configuration settings for Download, starting from Version 10 Patch 8, the URL redirection from HTTP to HTTPS will be handled both on the Bigfix server/ relay and on the BigFix client (direct download).

To support the download from an HTTPS URL, it will be necessary to provide appropriate trusted certificates to verify the remote server identity.

In case of a download from an HTTPS URL, the certificate of the remote server will be validated using the CA bundle distributed through the BES Support site. The CA bundle is the file that contains root and intermediate certificates of trusted authorities.

Before this new feature, the CA bundle was pre-installed on the BigFix server and already used for gathering purposes. With this new feature, the CA bundle will be distributed and kept up-to-date through the BES Support site for download purposes.

For more details about these settings, see Download (on page 766).

For more details about customizing HTTPS for downloads, see Customizing HTTPS for downloads (on page 129).

Relay Drive Space Protection From Downloads

Starting from Version 11, an optimization for the BigFix Relay was introduced.

To prevent the BigFix Relay ActiveDownloads folder from filling up, a new setting named `_BESRelay_Download_ActiveDownloadsMaxSizeMB` was created. This setting represents the maximum allowed size of the folder contents. For more information about this setting, see Download (on page 766).

The ActiveDownloads folder is used by the BigFix Relay to store the contents it is downloading, before caching them for the other BigFix Clients.

Introducing this cap means that the BigFix Relay does not allow to download files that exceed the limit you set. Before start downloading a file, the BigFix Relay checks the ActiveDownloads free space, and, depending on the setting value, allows the download only if the file size does not exceed the remaining space.

Suspending downloads for expired actions

By default, if an action expires while the Relay is downloading some files for that action, the BigFix Relay completes the downloads.

Starting from Version 11 Patch 5, you can choose to suspend these downloads using the new setting named `_BESRelay_Download_SuspendExpiredAction`.

When set to 1, the downloads for expired actions are suspended. You have to restart the BigFix Relay after changing its value.

If more than one action is required to download the same file, the BigFix Relay suspends the download only if all these actions expire.

If a new action requires to download a suspended file, the BigFix Relay resumes the suspended download, avoiding to restart from the beginning.

For more information about this setting, see Download (on page 766).

Enforcing the use of HTTPS for downloads

Starting from Version 11 Patch 5, the `_BESRelay_Download_UseHttps` setting allows you to control how server download URLs are handled. By default (`0`), the URL protocol is used as-is; setting it to `1` forces HTTP URLs to use HTTPS, while setting it to `2` tries HTTPS first and falls back to HTTP if needed (only for http:// URLs).



Note: At Version 11 Patch 5 level, this setting applies only to the Root Server.

Starting from Version 11 Patch 6, this setting applies also to the BigFix Relay.

For more information about this setting, see HTTPS (on page 575).

Enabling data pre-cache

You can specify for each Client if the data to download to run an action must be pre-cached on the Client or not and how.

You can decide whether the data requested to run an action on a Client can start to be downloaded:

After all constrains are satisfied.

If you do so, after all constraints are satisfied, the action needs to wait for the entire data to be downloaded, in the *pre-fetch* area, before starting to run. In this case the data download is a constraint itself. When the data download to the pre-fetch area completes, the action satisfies all the constraints so the data is moved in the *__Download* area and the action can start running. The risk is that the download might take longer than expected and, in the worst case, the time window during which the action can run could elapse before the data download completes, preventing the action from running.

Before all constrains are satisfied.

In this case the data download starts as soon as the action becomes relevant for the Client and before all constraints, such as *start time*, are evaluated. The data is downloaded on the Client disk in the *pre-cache* area. When all the constraints are met, the downloaded data is moved from the pre-cache area to the *pre-fetch* area. When the action is ready to start, the data is moved in the *__Download* area and the action starts running.

By doing so you allow the action to start earlier and, in case of an offer displayed to the user, you shorten the time that a user must wait for the offer to be downloaded after acceptance.

The potential risks in this case consist of the possibility of an action deadlock due to a lack of disk space, and, if running a group action, the possibility that the group will never starts because the disk space configuration does not allow the simultaneous existence of all downloads of the group in the Client system.

Starting from BigFix V9.5.10, you can prevent the risk to affect the group action processing by using the client setting **_BESClient_Download_PreCacheStageContinueWhenDiskLimited**. If you set **_BESClient_Download_PreCacheStageContinueWhenDiskLimited=1**, on that Client a group action can start, assuming that all other constraints are met, as long as the first sub action requiring downloads can collect all downloads for that sub action, and the action processing can continue even when the pre-cached downloads for all sub action cannot be available on the system at the same time due to disk space requirements (**DiskLimited** or **DiskFreeLimited** constraints).



Note: This setting does not affect single action processing, In other words, a single action or sub-action can still be constrained and blocked from running by insufficient disk space to hold downloads required for that specific action.

By default, on a BigFix Client **_BESClient_Download_PreCacheStageContinueWhenDiskLimited=0**, which means that a group action can start only after all the sub action downloads are available at the same time on the system.

Dynamic download White-lists

Dynamic downloading extends the flexibility of action scripts, adding the ability to use relevance clauses to specify URLs.

As with static downloads, dynamic downloads must specify files with the confirmation of a size or sha1. However, the URL, size, and sha1 are allowed to come from a source outside of the action script. This outside source might be a manifest containing a changing list of new downloads. This technique makes it easy to access files that change quickly or on a schedule, such as antivirus or security monitors.

This flexibility entails extra scrutiny. Because any client can use dynamic downloading to request a file, it creates an opportunity for people to use your server to host files indiscriminately. To prevent this, dynamic downloading uses a white-list. Any request to

download from a URL (that is not explicitly authorized by use of a literal URL in the action script) must meet one of the criteria specified in a white-list of URLs that is contained in the following file:

On Windows systems:

```
<Server Install Path>\Mirror Server\Config
\DownloadWhitelist.txt
```

On Linux systems:

```
<Server Install Path>/Mirror Server/config/
DownloadWhitelist.txt
```

This file contains a newline-separated list of regular expressions using a Perl regex format, such as the following:

```
http://.*\.site-a\.com/.*
http://software\.site-b\.com/.*
http://download\.site-c\.com/patches/JustThisOneFile\.qfx
```

The first line is the least restrictive, allowing any file at the entire site-a domain to be downloaded. The second line requires a specific domain host and the third is the most restrictive, limiting the URL to a single file named "JustThisOneFile.qfx". If a requested URL fails to match an entry in the white-list, the download immediately fails with status NotAvailable. A note is made in the relay log containing the URL that failed to pass. An empty or non-existent white-list causes all dynamic downloads to fail. A white-list entry of "*" (dot star) allows any URL to be downloaded.

Creating custom client dashboards

You can create custom Client Dashboards, similar to those in the console. Dashboards are HTML files with embedded Relevance clauses that can analyze the local computer and print out the current results.

Clients with a dashboard have an extra tab to display the resulting report. Note that dashboard global variables can be accessed from custom dashboards owned by other operators regardless of permissions.

To create a Client Dashboard, you must create a new folder named __UISupport (note the leading underlines) in the __BESData folder. This is a subfolder of the client folder, so the final pathname looks like:

Program Files/BigFix Enterprise/BES Client/__BESData/__UISupport

Place the Dashboard file (named `_dashboard.html`) and any accompanying graphics files into this folder. The next time the client starts, it incorporates these files into its interface, adding to the **Dashboard** tab. When you click this tab, the Dashboard calculates the latest values of each Relevance clause and displays them.

Any further changes to your custom dashboard must be performed in the __UISupport folder and promoted by restarting the client computer.

The Relevance statements are embedded in the HTML inside special tags with the form:

```
<?relevance statement ?>
```

For example, to find and print the time, use the following:

```
<?relevance now ?>
```

When the client displays the page containing this statement, the client evaluates the Relevance clause “now?" and substitutes the value for the tag. The following sample HTML prints out the word “Date:?? and then the current date and time:

```
<html>
 <body>
 Date: <?relevance now ?>
 </body>
</html>
```

To refresh the Relevance evaluation, add this line to the file:

```
<html>
 <body>
 Date: <?relevance now ?>
 <A href="cid:load?page=_dashboard.html"> Refresh </A>
```

```
 </body>
</html>
```

This link, labeled **Refresh**, causes the page to reload. When it does, it reevaluates the relevance clauses. It is easy to see how you would add other Relevance expressions to this page.

For example, to print out the operating system and the computer name, add these two lines:

```
<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>
```

You can use style sheets to format the output. You can use the default style-sheet, **offer.css** for some preset formatting. Here is an example of a Dashboard with a title, a header, a refresh link, and a section of retrieved property values:

```
<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>
```

For the offer.css to work correctly, the following graphics files must be copied to the __UISupport directory from the Client directory:

```
      bodyBg.jpg,
      bodyHeaderBg.jpg
```

```
      bullet.gif
      sectionHeaderBG.gif
```

When run from the Client, this dashboard produces the following output:



To learn more about Relevance expressions, see the Relevance Language Reference.

Geographically locating clients

Because clients are often deployed in remote offices, it is useful to create a property that lets the clients report their own location.

You can create a location property in BigFix using the **Location Property Wizard**.

1. In the console, go to the **BigFix Management** domain, click **Computer Management** , and then click **Location Property Wizard** . A wizard document opens.
2. The wizard creates a named property allowing the clients to identify themselves based on their subnet, IP range, or other information. Read the instructions in the wizard to create the property.

Locking clients

You can change the locked status of any BigFix client in the network. This lets you exclude specific computers or groups of computers from the effects of Fixlet actions.

This could be useful, for example, if you wanted to exclude certain development computers from any changes or updates. It also provides a powerful technique for testing new Fixlet actions on a limited set of unlocked computers, while keeping the rest of the network locked down. client computers can be locked forever (until explicitly unlocked) or for a defined period of time.

Changes are made to the locked status of a client by sending an action. As a consequence, the Console operator must supply proper authentication to lock or unlock any computer.

Even though a client is locked, there is still a subset of commands that can be accepted by the client:

- clock changes
- relay select
- notify client
- client setting actions (through the Edit Settings dialog in the Console)
- administrator management (add/delete)
- subscribe and unsubscribe sites
- utility
- unlock actions
- any actions from the BES Support site.

To lock or unlock a computer, follow these steps:

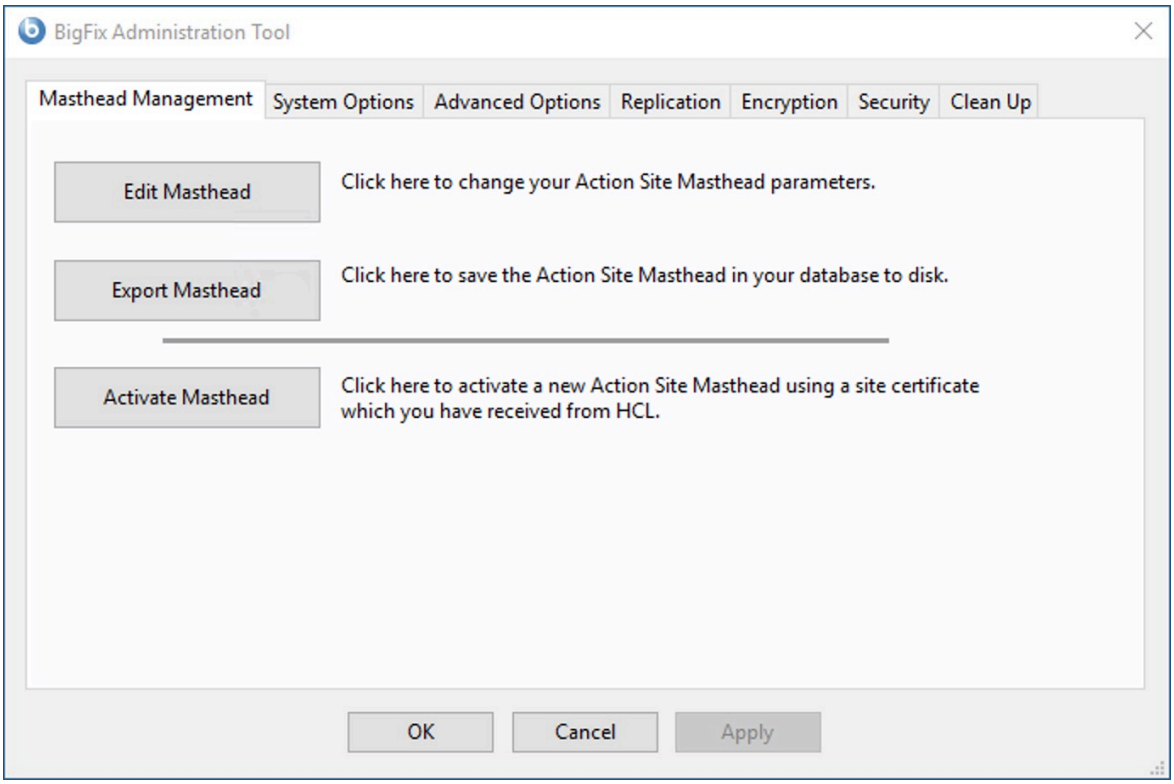
1. Click the **Computers** icon in the Domain Panel navigation tree to see the List Panel of networked BigFix client computers.
2. Select the computers that you want to lock.
3. Right-click and select **Edit Computer Settings** from the pop-up menu (or select **Edit Computer Settings** from the **Edit** menu). The Edit Setting dialog opens.
4. Click the check box to either lock or unlock the computer.

Although the console does not provide an explicit interface for setting an expiration date on the lock, you can create a custom action to do this. For more information, see the BigFix Developer site.

Editing the Masthead on Windows systems

You can change default parameters stored in the masthead by using the **BigFix Administration Tool**.

1. Launch the program from **Start > Programs > BigFix > BigFix Administration Tool**.
2. Browse to the private key (`license.pvk`) and click **OK**.
3. Select the **Masthead Management** tab and click **Edit Masthead**.



4. Enter the parameters of the masthead file that contains configuration and license information together with a public key that is used to verify digital signatures. This file is saved in your credential folder.
   You can edit the following options:

   Server Port Number:

   In general, you do not need to change this number. 52311 is the recommended port number, but you can choose a different port if that is more convenient for your particular network. Typically, you choose a port from the IANA range of private ports (49152 through 65535). You can use a reserved port number (ports 1-1024), but this might reduce the ability to monitor or restrict traffic correctly and it prevents you from using port numbers for specific applications. Do not change the server port number *after* installing the clients and creating the masthead, because BigFix might not work correctly. For additional information, see Modifying port numbers.

Gathering Interval:

This option determines how long the clients wait without hearing from the server before they check whether new content is available. In general, whenever the server gathers new content, it attempts to notify the clients that the new content is available through a UDP connection, circumventing this delay. However, in situations where UDP is blocked by firewalls or where network address translation (NAT) remaps the IP address of the client from the servers perspective, a smaller interval becomes necessary to get a timely response from the clients. Higher gathering rates only slightly affect the performance of the server, because only the differences are gathered; a client does not gather information that it already has.

Initial Action Lock:

You can specify the initial lock state of all clients, if you want to lock a client automatically after installation. Locked clients report which Fixlet messages are relevant for them, but do not apply any actions. The default is to leave them unlocked and to lock specific clients later on. However, you might want to start with the clients locked and then unlock them on an individual basis to give you more control over newly-installed clients. Alternatively, you can set clients to be locked for a certain period of time (in minutes).

Action Lock Controller:

This parameter determines who can change the action lock state. The default is **Console**, which allows any Console operator with management rights to change the lock state of any client in the network. If you want to delegate control over locking to the end user, you can select **Client**, but this is not recommended.

Exempt the following site URL from action locking:

In rare cases, you might need to exempt a specific URL from any locking actions. Check this box and enter the exempt URL. You can specify only one site URL and it must begin with `http://`.

Note: Baseline components are not exempt from action locking because they can come from different sites.

Last fallback Relay for all clients (replacing Root Server):

You might need to define a fallback relay for your clients when they do not connect to any relay specified in their settings. Select this check box and specify the fallback relay of your environment in one of the following formats:

- Hostname. For example, *myhostname*.
- Fully qualified domain name (FQDN). For example, *myhostname.mydomain.com*.
- IP address. For example, *10.10.10.10*.

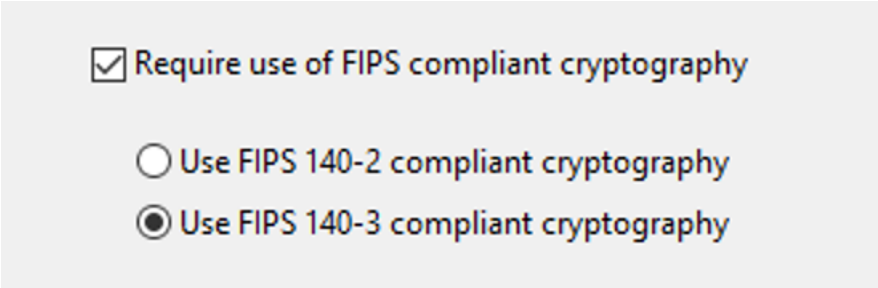
If you do not select this check box and define a fallback relay, the root server of your environment is used.

Note: Before specifying a fallback relay, ensure that any client or relay reporting directly to the root server has the root server defined as a relay. This setting will not prevent endpoints from selecting the root server. Set *_BESRelay_Register_Affiliation_AdvertisementList* on the BES Root Server to a group name that will not be set on any clients, such as DoNotSelectMe.

Require use of FIPS 140-2 compliant cryptography

Check this box to be compliant with the Federal Information Processing Standard in your network. This changes the masthead so that every BigFix component attempts to go into FIPS mode.

Note: Starting from BigFix version 11.0.7 and later, this check box is named **Require use of FIPS compliant cryptography**. This check box, when enabled, allows you to select either FIPS 140-2 or FIPS 140-3 mode. The two radio buttons are named **Use FIPS 140-2 compliant cryptography** and **Use FIPS 140-3 compliant cryptography**.



Note: BigFix Platform pre-11 components will no longer work if you enable FIPS 140-3.

Note: If you enable FIPS 140-3, BigFix Platform 11 components up to 11.0.6, will run in FIPS 140-2 mode.

Note: If you enable FIPS 140-3, BigFix Server requires at minimum the usage of the TLS 1.2 protocol with the Extended Master Secret (EMS) extension to successfully connect to HTTPS servers.

Allow use of Unicode filenames in archives

This setting specifies the codepage used to write filenames in the BigFix archives. Check this box to write filenames UTF-8 codepage.

Do not check this box to write filenames using the local deployment codepage, for example Windows-1252 or Shift JIS. If you run a fresh install of BigFix V9.5, by default, the filenames are written in UTF-8.

Note: If you upgraded your BigFix environment to V9.5, by default, the filenames are written in the local deployment code page.

5. Click **OK** to enter the changes.

Note: The masthead changes do NOT affect clients that are already deployed, but you can export the masthead using the Administration Tool (**Masthead Management** tab) and replace the masthead in the BES Installers directory of the BigFix server (default directory: `<drive>:\Program Files\BigFix Enterprise\BES Installers`) so that newly deployed or installed clients use these changes.

Editing the Masthead on Linux systems

To modify the masthead, up to BigFix version 11.0.6, run the following command as super user.

```
./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> ]
```

To modify the masthead, starting from BigFix version 11.0.7, run the following command as super user.

```
./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> ]
 [ -advRequireFIPS_140_2_Crypto=<true|false> ]
[ -advRequireFIPS_140_3_Crypto=<true|false> ]
[ -advEnableFallbackRelay=0 | -advEnableFallbackRelay=1
 -advFallbackRelay=<host> ]
```

where:

`-sitePvkLocation=<path+license.pvk>`

Specifies the private key file (*`filename`*`.pvk`). This private key file and its password are required to run the Administration Tool. Only users with access to the site level signing key and password are able to create new BigFix operators.



Note: The notation `<path+license.pvk>` used in the command syntax stands for *`path_to_license_file`*`/license.pvk`.

`-sitePvkPassword=<password>`

Specifies the password associated to the private key file (*`filename`*`.pvk`). This setting is optional, if you omit it you will be asked to specify the password interactively when the command runs.

`-display`

Displays the current settings for the masthead.

`-advGatherSchedule (optional, integer)`

Determines how long the clients wait without hearing from the server before they check whether new content is available. In general, whenever the server gathers new content, it attempts to notify the clients that the new content

is available through a UDP connection, circumventing this delay. However, in situations where UDP is blocked by firewalls or where network address translation (NAT) remaps the IP address of the client from the servers perspective, a smaller interval becomes necessary to get a timely response from the clients. Higher gathering rates only slightly affect the performance of the Server, because only the differences are gathered; a client does not gather information that it already has. Valid values are:

```
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)`

Determines who can change the action lock state. The default is **Console**, which allows any Console operator with management rights to change the lock state of any client in the network. If you want to delegate control over locking to the user, you can select **Client**, but this is not recommended. Valid values are:

```
0=console,
1=client,
2=nobody
```

`-advInitialLockState (optional, integer)`

Specifies the initial lock state of all clients. Locked clients report which Fixlet messages are relevant for them, but do not apply any actions. The default is to leave them unlocked and to lock specific clients later on. However, you might

want to start with the clients locked and then unlock them on an individual basis to give you more control over newly-installed clients. Alternatively, you can set them to be locked for a certain period of time. Valid values are:

```
0=Locked,
1=timed (specify duration),
2=Unlocked
```

`-advInitialLockDuration (optional, integer)`

Defines the period of time in seconds the clients must be locked.

`-advActionLockExemptionURL (optional, string)`

In rare cases, you might need to exempt a specific URL from any locking actions. Check this box and enter the exempt URL.

Note: You can specify only one site URL and it must begin with `http://`.

`-advRequireFIPScompliantCrypto (optional, boolean)`

Implements the Federal Information Processing Standard on your network up to BigFix version 11.0.6. This changes the masthead so that every BigFix component attempts to go into FIPS mode.

Note: Enabling FIPS mode prevents the use of some authentication methods when connecting to a proxy. If you selected to use a proxy to access the Internet or to communicate with BigFix subcomponents, ensure that the proxy configuration is set up to use an authentication method other than `digest`, `negotiate` or `ntlm`.

`-advRequireFIPS_140_2_Crypto (optional, boolean)`

Starting from BigFix version 11.0.7, this value replaces the `-advRequireFIPScompliantCrypto` value. Implements the Federal Information

Processing Standard 140-2 on your network. This changes the masthead so that every BigFix component attempts to go into FIPS mode.

Note: Enabling FIPS mode prevents the use of some authentication methods when connecting to a proxy. If you selected to use a proxy to access the Internet or to communicate with BigFix subcomponents, ensure that the proxy configuration is set up to use an authentication method other than `digest`, `negotiate` or `ntlm`.

`-advRequireFIPS_140_3_Crypto (optional, boolean)`

Starting from BigFix version 11.0.7, this value implements the Federal Information Processing Standard 140-3 on your network. This changes the masthead so that every BigFix component attempts to go into FIPS mode.

Note: Enabling FIPS mode prevents the use of some authentication methods when connecting to a proxy. If you selected to use a proxy to access the Internet or to communicate with BigFix subcomponents, ensure that the proxy configuration is set up to use an authentication method other than `digest`, `negotiate` or `ntlm`.

Note: BigFix Platform pre-11 components will no longer work if you enable FIPS 140-3.

Note: If you enable FIPS 140-3, BigFix Platform 11 components up to 11.0.6, will run in FIPS 140-2 mode.

Note: If you enable FIPS 140-3, BigFix Server requires at minimum the usage of the TLS 1.2 protocol with the Extended Master Secret (EMS) extension to successfully connect to HTTPS servers.

`-advEnableFallbackRelay (optional,boolean)`

Enables or disables a fallback relay for your clients when they do not connect to any relay specified in their settings. If you do not define a fallback relay, the root server of your environment is used.

`-advFallbackRelay (optional, string)`

Defines the host name of the fallback relay of your environment in one of the following formats:

- Hostname. For example, *myhostname*.
- Fully qualified domain name (FQDN). For example, *myhostname.mydomain.com*.
- IP address. For example, *10.10.10.10*.

Note: Before specifying a fallback relay, ensure that any client or relay reporting directly to the root server has the root server defined as a relay. This setting will not prevent endpoints from selecting the root server. Set *_BESRelay_Register_Affiliation_AdvertisementList* on the BES Root Server to a group name that will not be set on any clients, such as DoNotSelectMe.

Obfuscating server passwords

You can replace the passwords on the server with obfuscated passwords, by indicating the type of password that you want to replace and the new password.

The password is obfuscated and stored in the registry on Windows systems and in the configuration files on Linux systems.

Run the following command to obfuscate server password:

On Windows systems:

```
BESAdmin.exe /updatepassword /type:<type> [ /password:<password>
 ]
/sitePvkFile:<path+license.pvk> [ /sitePassword:<pvk_password> ]
```

where:

`type:<type>`

Specifies one of the following types of password:

server_db

The password to be updated and recorded obfuscated is related to the connection with the server database

dsa_db

The password to be updated and recorded as obfuscated is related to the connection with the DSA database

`password:<password>`

Specifies the password to be obfuscated and then recorded.

`sitePvkFile:<path+license.pvk>`

Specifies the private key file (*`filename`*`.pvk`). This private key file and its password are required to run the Administration Tool. Only users with access to the site level signing key and password are able to create new BigFix operators.

Note: The notation `<path+license.pvk>` used in the command syntax stands for *`path_to_license_file`*`\license.pvk`.

`sitePassword:<password>`

Specifies the password associated to the private key file (*`filename`*`.pvk`). This setting is optional, if you omit it you will be asked to specify the password interactively when the command runs.

On Linux systems:

where:

`type=<type>`

Specifies one of the following types of password:

server_db

The password to be updated and recorded obfuscated is related to the connection with the server database

dsa_db

The password to be updated and recorded as obfuscated is related to the connection with the DSA database

`password=<password>`

Specifies the password to be obfuscated and then recorded.

`sitePvkLocation=<path+license.pvk>`

Specifies the private key file (*`filename`*`.pvk`). This private key file and its password are required to run the Administration Tool. Only users with access to the site level signing key and password are able to create new BigFix operators.

Note: The notation `<path+license.pvk>` used in the command syntax stands for *`path_to_license_file/`*`license.pvk`.

`-sitePvkPassword=<password>`

Specifies the password associated to the private key file (*`filename`*`.pvk`). This setting is optional, if you omit it you will be asked to specify the password interactively when the command runs.

Modifying Global System Options

To modify a few basic system defaults, such as the minimum refresh time and the Fixlet visibility, perform the following steps.

On Windows systems:

1. Launch the Administration Tool from **Start > Programs > BigFix > BigFix Administration Tool**.
2. Select the **System Options** tab.
3. At the top, you can set the global **Minimum Refresh**. The default is 15 seconds, which is a good balance between responsiveness and low network load. If you find that these communications are impacting your network, you can increase the minimum to 60 seconds or more.
4. External sites are visible to all console operators by default, but you can change that in the section marked **Default Fixlet Visibility**. Click the lower button to make external content invisible to all except Master Operators. On the BigFix console, master operators can show hidden external content sites by clicking the **Show Hidden Content** button available in console toolbar.

On Linux systems:

1. From the `/opt/BESServer/bin` command prompt start the command line:

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

2. From the `/opt/BESServer/bin` command prompt run the following command:

```
./iem get admin/options  > /appo/options.xml
```

3. In the `/appo/options.xml` file:

```
<?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>
```

edit the following keywords to set the minimum refresh time in seconds and the external sites as visible to all the console operators or to only the Master operators:

```
<MinimumRefreshSeconds>15</MinimumRefreshSeconds>
<DefaultFixletVisibility>Visible</DefaultFixletVisibility>
```

If you change the value assigned to these keywords, you must restart the BigFix console active sessions to activate changes.

Note: On the BigFix console, Master operators can show hidden external content sites by clicking the **Show Hidden Content** button available in console toolbar.

4. Upload the modified file by running the following command:

```
./iem post /appo/options.xml admin/options
```

Extending the BigFix License

When you first request your action site license, you are issued a license for a specific period of time. Before your license expires, BigFix warns you, giving you sufficient time to renew your license.

When you are coming close to the expiration date, BigFix notifies you by showing a message in the BigFix Console. Similarly, if you start to exceed the number of clients allocated by your license, BigFix alerts you. To extend your license expiration or add new client licenses to your installation, follow these steps:

1. Notify your HCL representative (if you have not paid for the extended license, you must talk to your sales person or reseller to buy an extended license).
2. Your server checks daily for a new version of your license. If you want to force your server to check immediately, in the console, go to the **BigFix Management** domain, click **License Overview** node, and click the **Check for license update**.

For additional information about how to manage licenses see Managing licensesthe Managing licenses chapter of the Administration Guide.

Re-creating Site Credentials

Private and public key encryption creates a chain of signing authority from the BigFix root down through the Site Administrator and including each console operator.

If you lose your site credential or change the IP address of your server, the chain is broken. The consequences are serious: you must start again with a new request to HCL for a site certificate. Then you must reinstall the entire system, including all the clients (contact your support technician for details about how you might migrate your clients to a new server) and re-create all the users. If this happens, contact your support technician. To protect your site certificate, follow these important rules:

- **Do not lose the certificate (license.crt)** and **the private key for your site (license.pvk)**. Follow standard procedures for backing up and securing critical confidential information.
- **Do not change the IP address and hostname or port number of the server**, because it is the primary identifier for your site certificate. Any change to the IP address or port number that was specified when the license was requested negates the license and necessitates a fresh installation of the BigFix system. If you plan to decommission a server, be sure to apply the same IP address and port number to the replacement server.
- **Do not forget your password.** Follow your corporate standards for noting and storing your password.

Note: The BigFix Site Administrator can change the password of the site-level key, if they know the current password.

Creating special custom sites whose name begins with FileOnlyCustomSite

The PropagateFiles.exe tool can be used to create special custom sites whose name begins with FileOnlyCustomSite that are generated to propagate files to agents.

To log in by using `PropagateFile.exe`, you must enable the **Can use Console** permission.

This procedure describes how to create special custom sites whose name begins with FileOnlyCustomSite.

1. Create a dummy directory containing an empty file, for example: `C:\dummy` with `foo_file.txt`.
2. Navigate to the BigFix server directory. For example, `C:\Program Files (x86)\BigFix Enterprise\BES Server`.
3. To authorize a user to write in to the special custom site, run the `PropagateFiles.exe` tool as follows:

   ```
   PropagateFiles.exe CreateFileOnlyCustomSiteUserAuthorization <site
    admin pvk> <site admin password> <serverURL> <operator
    name> <operator password> <site name>
   ```

   where:

   - *site admin pvk*: The full path to the BigFix license file (`license.pvk`).
   - *site admin password*: The password for the license file.
   - *serverURL*: The server URL that is copied from the masthead file.
   - *operator name*: The name of an existing user.
   - *operator password*: The user password for the user.
   - *site name*: The name of the special custom site. The site must contain only files and the name must begin with `FileOnlyCustomSite`.

For example, `PropagateFiles.exe CreateFileOnlyCustomSiteUserAuthorization C:\licenses\licence.pvk lcs_password http://bigfixserver:52311 bigfixuser bfu_password FileOnlyCustomSite_FOO`.

Alternatively, you can use your Windows credentials to authenticate:

```
PropagateFiles.exe CreateFileOnlyCustomSiteUserAuthorization
 Windowsauthentication <site admin pvk> <site admin
 password> <serverURL> <site name>
```

4. To propagate the dummy directory contents to the custom site, run the following command:

   ```
   PropagateFiles.exe UpdateFileOnlyCustomSite <serverURL> <operator
    name> <operator password> <pvk file location> <directory to
    propagate> <site name>
   ```

   For example, `PropagateFiles.exe UpdatefileOnlyCustomSite http://bigfixserver:52311 bigfixuser bfu_password C:\licenses\licence.pvk C:\dummy FileOnlyCustomSite_FOO`.

   Alternatively, you can use your Windows credentials to authenticate:

   ```
   PropagateFiles.exe UpdateFileOnlyCustomSite
    Windowsauthentication <serverURL> <directory to propagate> <site
    name>
   ```

   Attention: This operation replaces the custom site contents with your directory contents.

5. To distribute the custom site with its file content to the targets, run a custom action with an action script.

   ```
   custom site subscribe CustomSite_<site name> as "<site name>" on
    "{parameter "action issue date" of action}"
   ```

For example:

```
custom site subscribe CustomSite_FileOnlyCustomSite_FOO as
 "FileOnlyCustomSite_FOO" on "{now}"
```

A special custom site called `CustomSite_FileOnlyCustomSite_FOO` is created.

Special custom sites are not visible on the BigFix console. To verify that the site has been correctly created, check the following files:

- `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`

Support of a custom certificate authority

In case you configured your deployment with a custom certificate authority, in order to successfully run PropagateFiles.exe you must copy the root certificate into the following directory of the machine where you need to run the tool:

`%LOCALAPPDATA%\BigFix\PropagateFiles\<besserver>\CustomSSLCertificates`

Configuring Client CPU Utilization

How to configure the amount of CPU the BigFix client uses on an endpoint machine.

Applies to

BigFix Platform

Problem

CPU utilization by the BigFix client and how to govern the amount of CPU the BigFix client uses.

Resolution

The amount of CPU the BigFix client uses on an endpoint machine during the evaluation cycle is governed by the following client settings:

- _BESClient_Resource_WorkIdle
- _BESClient_Resource_SleepIdle
- _BESClient_Resource_ScaleWorkTimeWithCPUCores

By default, the _BESClient_Resource_WorkIdle setting is set to 10 and the _BESClient_Resource_SleepIdle setting is set to 480.

The BigFix client will do work (evaluate relevance) for a designated amount of time then go to sleep for a designated amount of time. The WorkIdle setting controls how many milliseconds to work before going to sleep in each cycle and the SleepIdle setting controls how many milliseconds to sleep after performing work in cycle. If the WorkIdle setting is high in comparison to the SleepIdle setting, then the BigFix client will evaluate Fixlet relevance faster, but the CPU usage will be higher. By default, WorkIdle is 10 milliseconds and SleepIdle is 480 Milliseconds. Since 10 is 2% of 480 you can expect the BigFix client to use at most 2% of the CPU. Both WorkIdle and SleepIdle have maximum values of 500.

To determine the upper bound on the amount of CPU the agent will use with your custom settings, use the formula:

Max agent %CPU = workidle / (workidle + sleepidle)

Example (default settings): 10 / (10 + 480) = 2%

You can easily manage these settings and adjust the amount of CPU the BigFix client will use via Task # 168 in the BES Support site.

The task offers you the ability to set the client CPU usage to any of 5 different modes (using 5 different actions and preset values in the action script).

_BESClient_Resource_-WorkIdle	_BESClient_Resource_-SleepIdle	CPU Mode	CPU Upper bound
2	500	very low	< 0.5%

10	480	default	< 1-2%
25	460	medium	< 5%
50	450	high	< 10%
100	400	very high	< 20%

All of these calculations apply to the single processor where the agent runs, so if you have multiple processors the overall % of agent CPU is reduced significantly because it is divided by the number of processors. For example, if you want your agent to use less than 5% of CPU and the relevance 'number of processors' returns 4, you must set workidle to 100 and sleepidle to 400 because [100 / (100 + 400)] / 4 = 5%.

Doing these calculations might be cumbersome, and this is where the setting `_BESClient_Resource_ScaleWorkTimeWithCPUCores` comes into play, starting from Patch 7. If set to 1 (enabled) on a machine with multiple cores available, the work/sleep settings for all modes (Idle, Normal, FastHashVerify) will be applied across all available cores, thus allowing for longer bursts and faster evaluation times, without the burden of manually re-calculating them given different environments.

In case of a system with a single processor, the CPU Mode to a value different from the default is Not Recommended.

For AIX components using an LPAR:

- Max agent % CPU calculation for AIX LPAR with Entitled Capacity for Mode of Uncapped and Mode of Capped.
- - Usage _BESClient_Resource_Entitlement=100 for AIX LPARs with mode of Uncapped can be considered to use to eliminate reduction of Max Agent CPU% by Entitled Capacity.

For the behavior of pSeries Hypervisor see the example below:

With Mode: Uncapped, Entitled Capacity : 0.10, if the AIX LPAR uses 0.10 CPU core, then 100% usage. If the physical CPU cores in the

pSeries pool are not fully utilized, then pSeries hypervisor will allow the AIX LPAR to use more of CPU cores. You can also use pSeries LPAR weighting to give certain AIX LPARs like production higher priority for CPU resources than other AIX LPARs like development.

BigFix computer settings _BESClient_Resource_WorkNormal, _BESClient_Resource_SleepNormal when the BigFix agent is running a task that can be set.

Questions and Answers:

Question 1:

For Red Hat, SLES, Windows:

Max agent % CPU = workidle / (workidle + sleepidle)

0.0476 = 20 / (20 + 400)

0.0476 x 100 = 4.76%

...

_BESClient_Resource_WorkIdle Statement "The "Entitled Capacity : 0.10," means we have 1/10th of a CPU so all calculations for the workidle/sleepidle need to have 0.1 multiplied against them as the system has told us we have 10% of a full CPU."

For AIX LPARs is the calculation the following?

Max agent % CPU = (workidle / (workidle + sleepidle)) x Entitled

Capacity

0.00476 = (20 / (20 + 400)) x 0.10

0.00476 x 100 / = 0.476% of CPU available to AIX LPAR can be used

Answer:

It is true that the BigFix Max agent % CPU with be multiplied by the AIX LPAR Entitled Capacity to further reduce BigFix Max agent % CPU for AIX LPARs with mode of Uncapped or mode of Capped. The BigFix Max agent % CPU is based on a single physical CPU core.

Question 2:

For AIX LPAR with Mode: Uncapped from lparstat -i, can I turn off lparstat -i Entitled Capacity to reduce Max agent %CPU by setting the BigFix AIX computer _BESClient_Resource_Entitlement to 100?

Answer:

Yes

Question 3:

For BigFix agent on AIX LPAR with _BESClient_Resource_Entitlement to 100, will the behaviour be the same as Linux on Intel and Windows on VMware for limiting Max agent %CPU to a single physical CPU core?

Answer:

Yes.

There is also BigFix computer setting _BESClient_Resource_WorkNormal, _BESClient_Resource_SleepNormal when the BigFix agent is running a task that can be set.

Question 4:

Why BigFix Client process keeps on using high CPU% after the completion of the action?

Answer:

BigFix Client leaves on using high CPU% when it has no more actions to run and has completed an evaluation of the action sites.

Note:

- Test these settings on a limited set of endpoint machines before deploying them to the majority of your deployment endpoint machines.
- In practice, the CPU usage is typically lower than this ratio (because the agent will often be waiting for IO and it will yield its CPU time).
- If you adjust the CPU usage away from the default and experience issues, switch the CPU usage back to the default values.
- These CPU settings strictly govern BES Client evaluating content in the evaluation mode of the client's cycle. The BigFix agent might use up to 50%



of the CPU at times during the execution parts of its operation (i.e. initiating an install). These CPU spikes are normal and are expected to be short lived to where they would not be typically noticeable. If the BES Client experiences a sustained CPU spike it may mean there is a problem in the client executing on a problematic action or a systemic issue. This would need to be investigated and the problem determined through analyzing the BigFix Client logs and/or BigFix Client debug log.

Allowing the clients to use additional CPU to complete download operations

After download actions, the BigFix Client performs the sha code evaluation on downloaded files. For large files, the time taken to evaluate can become very long, especially on a BigFix Client constrained to default 2% CPU. The optimization of this time is achieved by allowing the BigFix Client to temporarily use additional CPU during this operation.

Starting from BigFix Version 10 Patch 2, you can speed up the operations to evaluate the sha code in downloaded files by temporarily directing the BigFix Client to use additional CPU. This results in a consistent time optimization for the download phase as the time required for the evaluation decreases as the CPU used increases.

The `_BESClient_Resource_WorkFastHashVerify` and `_BESClient_Resource_SleepFastHashVerify` configuration settings allow you to manage the amount of CPU that the BigFix Client can use.

For more details about these settings, see CPU Usage (on page 619).

The `_BESClient_Download_FastHashVerify` configuration setting enables the BigFix Client to temporarily use additional CPU for the Hash Code Evaluation (default increase to 25%).

For more details about the setting, see Download (on page 766).

Customizing Client UI message strings

You can customize the Client UI message strings. It is possible to alter the English text output and also to change any of the localizable string translations by using the XLAT

files for the Client UI. The Client UI XLAT files default location for a 64-bit machine is `C:\Program Files (x86)\BigFix Enterprise\BES Client\BESLib\Reference`.

Proceed as follows:

1. Locate the file named `<lang>.xlat` under `C:\Program Files (x86)\BigFix Enterprise\BES Client\BESLib\Reference` where `<lang>` is the identifier of the language you want to address (e.g. ITA).
2. Browse `C:\Program Files (x86)\BigFix Enterprise\BES Client\BESLib\Reference\<lang>.xlat` and identify the line with the translation string that you want to modify.
3. Save this line with the modified translation into `C:\Program Files (x86)\BigFix Enterprise\BES Client\__BESData\__UISupport\_brand<lang>.xlat`.
4. Restart the BigFix client.

Note: The file `C:\Program Files (x86)\BigFix Enterprise\BES Client\BESLib\Reference\ENU.xlat` (English xlat) is empty. However, you can follow the procedure described above by looking for the English strings in any of the other language files.

Note: After the upgrade to Version 10, you need to align the English strings of the custom XLAT files to the ones of the upgraded `C:\Program Files (x86)\BigFix Enterprise\BES Client\BESLib\Reference\<lang>.xlat` files.

Chapter 24. Migrating the BigFix Server (Windows/MS-SQL)

This section details the steps and operational procedures necessary for migrating the BigFix Server from existing hardware onto new computer systems.

Typical use cases for these steps include:

- Hardware refresh
- OS or SQL Server upgrades
- 32-bit to 64-bit architecture migration
- Remote SQL server migration

Due to the complexity and risks of migrating BigFix Servers, it is strongly recommended that an BigFix Technician help in performing the BigFix Server Migration process.

Considerations for migration

This section provides some notes and guidelines for the migration of the BigFix Root or application server.

General notes and guidelines

- The migration should first be performed and tested in a segregated test/development environment, if possible.
- If leveraging BigFix Disaster Server Architecture (DSA - Disaster Server Architecture), the replica/secondary server should be migrated before the primary BigFix Server.
- Custom settings that have been applied to the BigFix Server will need to be implemented again after migration. Typical examples include: Web Reports HTTPS configurations, Download Gather Cache Size, etc…
- Download plug-ins and other extensions/applications will also need to be re-installed in any new installation location.
- Typical examples include: Unmanaged Asset Importer, Wake on LAN medic, Upload service, Automation Plan Engine.

Pre-migration checklist

- Ensure that a strategy has been determined to allow the Clients to continue to connect to the new BigFix Server per the GatherURL specified in the masthead (corresponding to Assumption #1 above).
- Back up the BFEnterprise and BESReporting SQL databases.
- Back up the site level credentials such as license.crt, license.pvk, and the masthead. If using <8.1 then you should also back up user/operator credentials such as publisher.pvk and publisher.crt.
- Document the authentication method to the MSSQL database (SQL versus NT).
- If using NT Authentication, document the NT Domain/service account used for BigFix Server services.
- If using SQL Authentication, document the SQL account used for SQL Authentication Registry values.
- Document (consider taking a screenshot) the ODBC connections: bes_BFEnterprise, bes_EnterpriseServer, enterprise_setup, and LocalBESReportingServer. For 64-bit Windows systems, use the 32-bit version of the ODBC tool (C:\Windows \SysWOW64\odbcad32.exe) to configure the System DSNs.
- If migrating the Primary BigFix Server, consider implementing the following prior to the migration to reduce downtime:
    - Change the following BigFix Client settings on all clients:
        - _BESClient_Report_MinimumInterval = 3600.

          This setting will reduce the amount of incoming data from the endpoints to allow the system to recover more quickly and reduce potential downtime.
        - _BESClient_RelaySelect_ResistFailureIntervalSeconds = 21600

          This value represents the amount of time BES Clients will wait after its relay appears down before performing BES Relay selection. This can prevent unnecessary automatic relay selection during the migration.

    - Change the heartbeat in the BigFix Console to 6 hours: Console Preferences

      Note: This is another way to reduce the amount of incoming data from the endpoints.

- Carefully review the migration steps.

Migrating the BigFix root server

How to migrate the root server.

The following scenarios are assumed to be true prior to performing the BigFix Server migrations:

- If migrating the Primary/Master BigFix server, the new BigFix server will have to leverage the same DNS name/alias or IP address that is specified in the masthead/license (https://support.hcl-software.com/csm?id=kb_article&sysparm_article=KB0023184), otherwise the BigFix infrastructure will not be able to communicate with the new BigFix server. If this is not possible, a new license may need to be obtained, and an infrastructure migration be performed rather than a server migration. This is a crucial element of the migration strategy, and requires proper planning!
    - If the masthead leverages an IP address, the new Server will have to leverage the same IP address.
    - If the masthead leverages a host name, the new Server may have to leverage the same host name.
    - If the masthead leverages a DNS name/alias (per best practice), the alias will have to be re-pointed to the new BigFix server as part of the migration process. If leveraging a DNS name/alias within the masthead, perform a DNS switch for the DNS name so that the alias now points to the new BigFix Server. Wait for the DNS switch to propagate (this might take some time depending on your DNS services/infrastructure).
- The existing BigFix server is operating normally before the migration.

- The new BigFix server has been built, meets the requirements of an BigFix server, and is properly configured to serve as a BigFix server. In particular, the OS and database platforms should be supported for the given BigFix version being migrated.
- The installation folders are in the same location and path for the original BigFix /DSA servers and the new BigFix /DSA servers (if not, some manual modification of files will be necessary, which is outlined in the steps below).
- The migration is performed off-hours to minimize potential impact or down-time.

1. To facilitate migration verification, note the current actionsite version.
    - For any BigFix server version: https://support.hcl-software.com/csm?id=kb_article&sysparm_article=KB0023338.
    - With v8.2 and above, the actionsite version can also be obtained from the Server's Diagnostics page (http://<BigFixServer:port>/rd), select the 'Get Current Version' request type under Site Gathering Information, select the actionsite URL from the dropdown, click Submit, and note the actionsite version.
2. Stop and consider disabling all BES Services on the original Server.
3. Run the Server Backup procedure as described in Server Backup.
4. Run the Server Recovery procedure as described in Server Recovery.
5. Run the Verify restore results procedure as described in Verifying restore results.
6. Verify that the actionsite version being hosted by the new BigFix Server matches the one noted in Step 1 using the same steps outlined in Step 1.
7. Check the relay selection settings on all top-level Relays. If any setting points to the original BigFix Server using an IP Address or hostname, they may need to be re-pointed to the new BigFix server.
8. Uninstall the BigFix Server software from the old BigFix Server computer. Do NOT restart the BES Services on this computer. Attempting to use the old BigFix Server may cause errors on the new BigFix Server if it is used again.
9. Run BESAdmin.exe /resetDatabaseEpoch to force the consoles to refresh their cache with the new server.
10. Reset the Client settings and heartbeat to settings prior to shutting down the BigFix Server services.

Migrating databases

This procedure applies to remote database installations.

Before you begin

- Back up the BFEnterprise and BESReporting SQL databases (A current backup must be taken immediately prior to the move. You must not have any differences between the backup and production database).
- Back up the "BFEnterprise Full Database Index Reorganization" job.
  - Open SQL Server Management Studio.
  - Expand **SQL Server Agent**, and expand **Jobs**.
  - Right-click the **BFEnterprise Full Database Index Reorganization** job and select **Script Job as**.
  - Select **CREATE To**, then select New Query Editor Window, File, or Clipboard to select a destination for the script. Typically, the destination is a file with a .sql extension.
- Document the authentication method to the MSSQL database (SQL versus NT).
  - If using NT Authentication, document the NT Domain/service account used for BigFix Server services.
  - If using SQL Authentication, document the SQL account used for SQL Authentication Registry values.
- Document (consider taking a screenshot) the ODBC connections: bes_BFEnterprise, enterprise_setup, and LocalBESReportingServer. Record both 32-bit and 64-bit information in order to replicate them.

  Note: For 64-bit Windows systems, you must use the 32-bit version of the ODBC tool to configure the 32-bit System DSNs.

- Use ODBC wizard on the Root Server to test a basic connection to new database location (new MS-SQL Server).
- Consider implementing the following prior to the migration to reduce downtime:

    - Change the following BigFix Client settings on all clients:
        - _BESClient_Report_MinimumInterval = 3600 * This setting will reduce the amount of incoming data from the endpoints to allow the system to recover more quickly and reduce potential downtime.
        - _BESClient_RelaySelect_ResistFailureIntervalSeconds = 21600 * This value represents the amount of time BES Clients will wait after its relay appears down before performing BES Relay selection. This can prevent unnecessary automatic relay selection during the migration.
    - Change the heartbeat in the BigFix Console to 6 hours: Console Preferences

    Note: This is another way to reduce the amount of incoming data from the endpoints.

- Carefully review the migration steps.

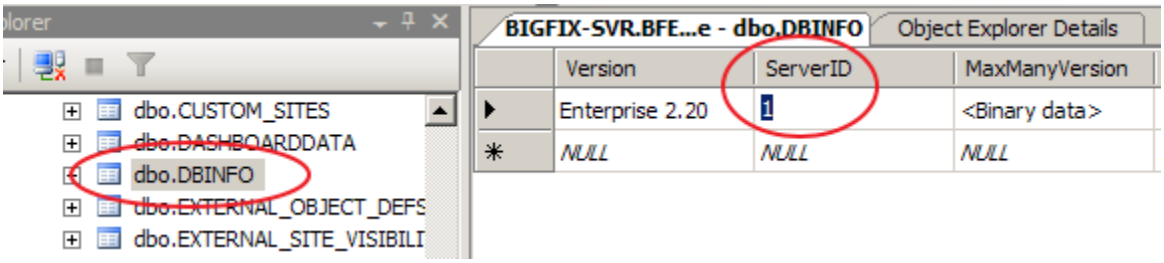
Procedure

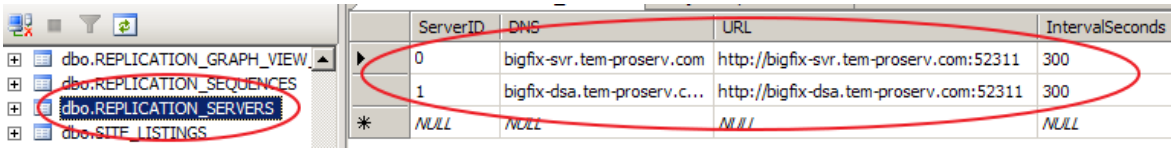
1. Stop all BES Server services.
2. Detach the BFEnterprise and BESReporting databases from the current SQL Server instance databases.
3. Move the BFEnterprise and BESReporting databases to the new SQL Server instance.
4. Attach the BFEnterprise and BESReporting databases to the new SQL Server instance.
5. Restore the "BFEnterprise Full Database Index Reorganization" job from the script.
    - Open SQL Server Management Studio.
    - On the File menu, Open the file containing the scripted job.
    - Execute the script to create the "BFEnterprise Full Database Index Reorganization" job.
6. Modify the ODBC System DSNs (bes_BFEnterprise, enterprise_setup, and LocalBESReportingServer) to point to the new SQL server instance. This modification will allow you to avoid re-installing the BigFix Server application.

- Use ODBC connection wizard to test connection.
- Update and verify both 32-bit and 64-bit configurations.

Note: For 64-bit Windows systems, you must use the 32-bit version of the ODBC tool to configure the System DSNs.

7. If leveraging DSA, use SQL Server Management Studio to connect to the BFEnterprise database and examine the DBINFO and REPLICATION_SERVERS tables:





If DNS aliases are being leveraged for the servers, this should not change. If is using hostnames, and the hostnames are changing, these column values may need manual modification.

8. If passing from a local to a remote database or vice versa, you need to handle the value for the UseRemoteDB setting. The setting "HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\BigFix\Enterprise Server\UseRemoteDB" must be set to "0" if the BigFix database is local or set to "1" if the BigFix database is remote.
9. If passing from Windows to SQL authentication, you need to add the following entries in the registry:
   - for the BigFix server:

     [HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\BigFix\Enterprise Server\Database]

     "User"="the new SQL account"

     "Password"="the SQL account password"

- for Web Reports:

  [HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\BigFix\Enterprise Server\FillAggregateDB]

  "LocalDBDSN"="LocalBESReportingServer"

  "Username"="the new SQL account"

  "Password"="the SQL account password"

10. Run the BigFix Administration Tool without using any options.
11. Start all BES Server services.

Verifying the migration

To make sure that your BigFix Server has been successfully migrated, perform the procedure outlined in this section.

1. Check the BigFix Diagnostics Tool to make sure all services are properly started.
2. Log in the BigFix Admin tool (if it opens normally database connectivity is verified and the tool can be closed).
3. Log in with the BigFix Console and verify that the logins work properly and the database information was properly restored.
4. BigFix Clients and BigFix Relays should soon notice that the Server is available and will be reporting data to the server. Full recovery with all Agents reporting will usually take anywhere from a few minutes to many hours (depending on the size of the deployment and how long the Server was unavailable). In any circumstance, at least some Agents should be reporting updated information within an hour or so.
5. After verifying some agents are reporting properly, send a "blank action" (Tools > Take Custom Action, target "All Computers", click OK) to all computers. The blank action will not make any changes to the Agent computers, but the Agents will report that they received the blank action. If the most Agents respond to a blank action, it is a very strong indicator that everything is working well because sending an action tests many core components and communication paths of BigFix.
6. Log in to Web Reports and ensure the data was restored properly.
7. Contact BigFix Support with any issues or questions.

Chapter 25. Migrating the BigFix Server (Linux)

This section provides basic information on migrating your BigFix Server from existing Linux hardware onto new systems.

The procedures referenced below are meant for the server components only. You must backup and restore additional applications and/or server customization in your setup, if any, separately.

Note: Due to the complexity and risks of migrating BigFix Servers, it is strongly recommended that you take assistance from a trained BigFix Technician while migrating the BigFix Server.

Before you begin

The following scenarios are assumed to be true prior to performing the BigFix Server migrations:

- If migrating the Primary/Master BigFix server, the new BigFix server will have to leverage the same DNS name/alias or IP address that is specified in the masthead/license (https://support.hcl-software.com/csm?id=kb_article&sysparm_article=KB0023184), otherwise the BigFix infrastructure will not be able to communicate with the new BigFix server. If this is not possible, a new license may need to be obtained, and an infrastructure migration be performed rather than a server migration. This is a crucial element of the migration strategy, and requires proper planning!
    - If the masthead leverages an IP address, the new Server will have to leverage the same IP address.
    - If the masthead leverages a host name, the new Server may have to leverage the same host name.
    - If the masthead leverages a DNS name/alias (per best practice), the alias will have to be re-pointed to the new BigFix server as part of the migration process. If leveraging a DNS name/alias within the masthead, perform a DNS switch for

the DNS name so that the alias now points to the new BigFix Server. Wait for the DNS switch to propagate (this might take some time depending on your DNS services/infrastructure).

- The existing BigFix server is operating normally before the migration.
- The new BigFix server has been built, meets the requirements of an BigFix server, and is properly configured to serve as a BigFix server. In particular, the OS and database platforms should be supported for the given BigFix version being migrated.
- The installation folders are in the same location and path for the original BigFix /DSA servers and the new BigFix /DSA servers (if not, some manual modification of files will be necessary, which is outlined in the steps below).
- The migration is performed off-hours to minimize potential impact or down-time.

Procedure

1. To facilitate migration verification, note the current actionsite version.
   - For any BigFix server version: https://support.hcl-software.com/csm?id=kb_article&sysparm_article=KB0023338.
   - The actionsite version can also be obtained from the Server's Diagnostics page (http://<BigFixServer:port>/rd), select the 'Get Current Version' request type under Site Gathering Information, select the actionsite URL from the dropdown, click Submit, and note the actionsite version.
2. Stop and consider disabling all BES Services on the original Server.
3. Run the Server Backup procedure as described in Server Backup.
4. Run the Server Recovery procedure as described in Server Recovery.
5. Run the Verify restore results procedure as described in Verifying restore results.
6. Verify that the actionsite version being hosted by the new BigFix Server matches the one noted in Step 1 using the same steps outlined in Step 1.
7. Check the relay selection settings on all top-level Relays. If any setting points to the original BigFix Server using an IP Address or hostname, they may need to be re-pointed to the new BigFix server.
8. Uninstall the BigFix Server software from the old BigFix Server computer. Do NOT restart the BES Services on this computer. Attempting to use the old BigFix Server may cause errors on the new BigFix Server if it is used again.

9. Run ./BESAdmin.sh -resetDatabaseEpoch to force the consoles to refresh their cache with the new server.
10. Reset the Client settings and heartbeat to settings prior to shutting down the BigFix Server services.

Relocating databases on a remote server

How to relocate the database and upgrade it after its relocation.

General guidelines

If you want to move your local BigFix database, located on the same machine where the BigFix Server is installed, to another remote server, or you need to relocate it from a remote DB2 server to another one, take into account the following guidelines.

Since you need to back up BigFix databases from the current DB2 Server and to restore them on the new DB2 Server, it is strongly suggested to back up/restore using the same DB2 level.

If needed, perform the DB2 upgrade only after restoring the BigFix databases.

Relocating the BigFix databases on a different server is currently supported only using the same instance name (default is db2inst1).

For more details about the DB2 installation requirements and configurations, refer to the following links:

Database requirements

Installing and configuring DB2

Migrating BigFix databases

To migrate BigFix databases, perform the following steps:

1. Verify that the starting and the destination DB2 are at the same level and the new DB2 system uses the same instance (default: db2inst1).
2. Stop all BigFix services.
3. Run the DB2 database backup commands:

```
BACKUP DB BFENT COMPRESS
BACKUP DB BESREPOR COMPRESS
```

4. Restore the BigFix databases on the new DB2 system:

```
RESTORE DB BFENT
RESTORE DB BESREPOR
```

5. Catalog the new databases on the BigFix Server. From the BigFix Server, run the following DB2 commands:

```
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
```

   Where:

   {host} is your DB2 remote server hostname and {port} is the DB2 remote server port used.

6. Update the BigFix Server database settings (as needed) in `/var/opt/BESServer/besserver.config`.

```
[Software\BigFix\EnterpriseClient\Settings\Client\_BESServer_Database_
DatabaseAddress]
value = <new_hostname>

[Software\BigFix\EnterpriseClient\Settings\Client\_BESServer_Database_
Port]
value = "<new_port_number>"
```

7. If Web Reports is installed, update the following settings (as needed) in `/var/opt/BESWebReportsServer/beswebreports.config`.

```
[Software\BigFix\Enterprise Server\FillAggregateDB]
DatabaseAddress = <new_hostname>
Port = <new_port_number>
```

8. Update the DB2 password on the BigFix Server, (if the password of the db2inst1 user on the new DB2 server is different from the one you have on the old DB2 server) as described in Changing the database password.
9. Start the BigFix Server services (except the WebUI) using the commands:

   ```
   systemctl start besserver
   systemctl start besfilldb
   systemctl start besgatherdb
   systemctl start beswebreports
   systemctl start bespluginportal (if installed)
   systemctl start besclient
   ```

   Up to BigFix Version 11.0.1, for backward compatibility reasons, you can also run the commands:

   ```
   /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. Verify that the components start and the connection with the new configured database works.
11. If the WebUI is installed, run the BES Support Fixlet ID 2687 from the BES Console to update the "BigFix Server Database Host" with the new Database Configuration specifications (it will also start the WebUI service) .
12. Update the DNS field in the REPLICATION_SERVERS table with the new DB2 Server hostname.

Upgrading the database after its relocation

If you need to perform the DB2 upgrade, after you did a relocation from local to remote, you can follow these steps:

1. Remove the local DB2 Server from the BigFix Server machine.
   a. Stop all BigFix services.
   b. From the BigFix Server run the following DB2 commands:

   ```
   UNCATALOG DATABASE BFENT
   UNCATALOG DATABASE BESREPOR
   UNCATALOG NODE TEM_REM
   ```

   c. Proceed uninstalling the DB2 Server. For details, see the following IBM documentation: Uninstalling DB2 database products.
   d. Remove DB2 users and groups.
2. Install the IBM Data Server at the same version you are going to upgrade the server.
3. On the new DB2 Server, upgrade the DB2 Server to the target version following the IBM documentation.

   Note: Do not create a new instance during the new DB2 version installation.

4. After the DB2 Server was upgraded, define the remote node and catalog the remote databases on the BigFix Server:

   ```
   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. Test the database connection from the BigFix Server.
6. Start all BigFix services.

Chapter 26. Migrating the BigFix Server from Windows to Red Hat 9 with SQL Server

Starting from BigFix Version 11.0.1, you can migrate BigFix Platform from a Windows operating system to a Red Hat 9 operating system with Microsoft SQL Server 2022 Database, both using a local or a remote database.

For details about the steps to perform to migrate a BigFix Windows Server to Red Hat 9.x with Microsoft SQL Server 2022 Database, see How to migrate BigFix Windows Server to Red Hat 9.x with MS SQL 2022 Database (hcl-software.com).

This scenario applies to BigFix Platform Version 11.0.1.104 or later versions.

Chapter 27. Configuring logs

This page describes how to enable, disable, and configure the logs of the various BigFix components and the audit log files.

For the default paths of the BigFix log files, see Logging paths.

If a log configuration setting is stored as a BigFix Client setting, it is also documented in the List of settings and detailed descriptions (on page 404).

Root Server log

This log is written by the BESRootServer service and its default name is `BESRelay.log`. The BESRootServer service is in charge of network communication and directly interacts with other BigFix components. It takes input from the BigFix Console. It downloads BigFix content sites so that the GatherDB service can process them. It downloads files that are mentioned as prefetch items in Fixlets and Tasks. It sends content updates to BigFix Clients and issues management actions to them. It receives reports from BigFix Clients, allowing the FillDB service to process them. It communicates with top-level BigFix Relays. It answers queries from the BigFix Explorer, etc. The Root Server log can help troubleshoot issues relative to: BESRootServer database connection, file downloads, network interactions with other BigFix components, TLS certificates, etc.

The settings to configure this log are stored as BigFix Client settings.

On Windows, each client setting is a registry key that is a child of `HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\BigFix\EnterpriseClient\Settings\Client`, is named exactly like the setting, and contains a registry value named `value` of type `REG_SZ`, which stores the setting value.

On Linux, client settings affecting the BigFix Server may be stored in the configuration file `/var/opt/BESServer/besserver.config`, or in the BigFix Client configuration file `/var/opt/BESServer/besclient.config`, which takes precedence. Each client setting is represented by a section whose name starts with `SOFTWARE\BigFix\EnterpriseClient\Settings\Client\` and ends with the setting name. That section contains a line with the text `value=` followed by the setting value.

- `value` (`REG_SZ`) of `_BESRelay_HTTPServer_LogFilePath`, the full file path of the log
- `value` (`REG_SZ`) of `_BESRelay_HTTPServer_LogFileSizeLimit`, the maximum log file size in bytes before rotation occurs, defaults to 52428800 (50 MB)
- `value` (`REG_SZ`) of `_BESRelay_HTTPServer_LogFileRotationLimit`, the maximum number of files used for log rotation, defaults to 10
- `value` (`REG_SZ`) of `_BESRelay_Log_Verbose`, 0 to disable verbose logging, 1 to enable it, defaults to 0

Root Server and Relay HTTP logs

The BigFix Server and BigFix Relay components can record all HTTP(S) requests on a dedicated log. If this functionality is enabled, each day a new log file named like `YYYYMMDD.log` will be created to record all HTTP(S) requests that will be received that day. Logging this information can consume a large amount of space and may severely affect performances, so it is should only be done for troubleshooting purposes.

The settings to configure these logs are stored as BigFix Client settings.

- `value` (`REG_SZ`) of `_BESRelay_HTTPServer_HttpLogDirectoryPath`, the path of the folder containing the log file, defaults to empty
- `value` (`REG_SZ`) of `_BESRelay_HTTPServer_HttpLogExpirationDays`, the number of days that a log should be kept for, defaults to 0

No logging occurs if `_BESRelay_HTTPServer_HttpLogDirectoryPath` is empty or missing. On a fresh installation, that setting is created empty, so HTTP logs are disabled by default.

Root Server and Relay minidumps

On Windows, in addition to the above logs, you can enable the creation of minidumps in case an unexpected exception occurs in the BES Root Server and BESRelay services.

The setting to control this functionality is stored as a BigFix Client setting.

- `value` (`REG_SZ`) of `_BESRelay_HTTPServer_WriteMiniDumps`, 0 to disable the writing of minidumps, 1 to enable it, defaults to 0

FillDB log

This log is written by the FillDB service (BESFillDB on Linux) and its default name is `FillDB.log`. The FillDB service processes the BigFix Client reports received by the BESRootServer service. FillDB elaborates that data and updates the main database to reflect the current properties of managed devices, applicability of BigFix content, progress of management actions, and more. The FillDB service also performs the replication of content from other BigFix Server installations, if a DSA is configured. The FillDB log can help troubleshoot issues relative to: FillDB database connection, report processing, and DSA replication.

On Windows, the settings to configure this log are stored in the registry key `HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\BigFix\Enterprise Server\FillDB` and each setting is represented by a registry value.

On Linux, the same settings are stored in the configuration file `/var/opt/BESServer/besserver.config`, in the section `[SOFTWARE\BigFix\Enterprise Server\FillDB]` and each setting is represented by a simple key-value pair.

- `DebugOut` (`REG_SZ`), the full file path of the log, defaults to empty
- `EnableLogging` (`DWORD`), 0 to disable the log, 1 to enable it, defaults to 1
- `EnabledLogs` (`REG_SZ`), the types of information to log, either "all" or a subset of "database;critical;debug;replication", defaults to "critical"
- `LogFileSizeLimit` (`DWORD`), the maximum log file size in bytes before rotation occurs, defaults to 104857600 (100 MB)

No logging occurs if `DebugOut` is empty or missing.

The maximum number of rotating files used for this log is 10 and it is not configurable.

FillDB Performance log

This log provides provides a stream of performance measurements regarding FillDB operations, mainly regarding the amount of processed reports and the time it took to update the database with the information it received.

On Windows, the settings to configure this log are stored in the registry key `HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\BigFix\Enterprise Server\FillDB` and each setting is represented by a registry value.

On Linux, the same settings are stored in the configuration file `/var/opt/BESServer/besserver.config`, in the section `[SOFTWARE\BigFix\Enterprise Server\FillDB]` and each setting is represented by a simple key-value pair.

- `EnablePerformanceLogging` (`DWORD`), 0 to disable the log, 1 to enable it, defaults to 1
- `PerformanceDataPath` (`REG_SZ`), the full file path of the log, defaults to empty
- `PerfLogFileSizeLimit` (`DWORD`), the maximum log file size in bytes before rotation occurs, defaults to 104857600 (100 MB)

No logging occurs if `PerformanceDataPath` is empty or missing. On a fresh installation, that setting is created empty, so the FillDB performance log is disabled by default.

The `PerfLogFileSizeLimit` affects all FillDB performance logs.

The maximum number of rotating files used for this log is 10 and it is not configurable.

FillDB Query Performance log

This log provides provides a stream of measurements regarding the operations that FillDB performs to execute a BigFix Query, including its interactions with the BES Root Server.

On Windows, the settings to configure this log are stored in the registry key `HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\BigFix\Enterprise Server\FillDB` and each setting is represented by a registry value.

On Linux, the same settings are stored in the configuration file `/var/opt/BESServer/besserver.config`, in the section `[SOFTWARE\BigFix\Enterprise Server\FillDB]` and each setting is represented by a simple key-value pair.

- `EnableQueryPerformanceLogging` (`DWORD`), 0 to disable the log, 1 to enable it, defaults to 1
- `PerfLogFileSizeLimit` (`DWORD`), the maximum log file size in bytes before rotation occurs, defaults to 104857600 (100 MB)
- `QueryPerformanceDataPath` (`REG_SZ`), the full file path of the log, defaults to empty

No logging occurs if `QueryPerformanceDataPath` is empty or missing. On a fresh installation, that setting is created empty, so the FillDB Query performance log is disabled by default.

The `PerfLogFileSizeLimit` affects all FillDB performance logs.

The maximum number of rotating files used for this log is 10 and it is not configurable.

GatherDB log

This log is written by the GatherDB service (BESGatherDB on Linux) and its default name is `GatherDB.log`. The GatherDB service processes the BigFix content site data (Fixlets, etc.) downloaded by the BESRootServer service and imports it into the main database. The GatherDB log can help troubleshoot issues relative to: GatherDB database connection and BigFix site import.

On Windows, the settings to configure this log are stored in the registry key `HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\BigFix\Enterprise Server\GatherDB` and each setting is represented by a registry value.

On Linux, the same settings are stored in the configuration file `/var/opt/BESServer/besserver.config`, in the section `[SOFTWARE\BigFix\Enterprise Server\GatherDB]` and each setting is represented by a simple key-value pair.

- `DebugOut` (`REG_SZ`), the full file path of the log, defaults to empty
- `VerboseDebugOut` (`DWORD`), 0 to disable verbose logging, 1 to enable it, defaults to 0

No logging occurs if `DebugOut` is empty or missing.

There is no setting for enabling log rotation.

Server audit log

This log contains the audit events recorded by the BigFix Server and its name is always `server_audit.log`.

The BigFix Server audit log file contains information about:

- logins and logouts via BigFix Console, REST API, Web Reports, WebUI, etc
- activities performed by the BigFix operators
- BigFix Actions sent to the BigFix clients and their cancellations

The settings to configure this log are stored as BigFix Client settings.

- `value` (`REG_SZ`) of `_Audit_Logging_LogMaxSize`, the maximum log file size in bytes before rotation occurs, defaults to 104857600 (100 MB)
- `value` (`REG_SZ`) of `_BESRootServer_Audit_Verbosity`, whether SSL connections are logged, either "all" or empty, defaults to empty
- `value` (`REG_SZ`) of `_BESServerAudit_Logging_LogDirectoryPath`, the path of the folder containing the log file, introduced in Version 11.0.4, default depends on OS

When the size limit is reached, the log file is renamed following the pattern `server_audit.YYYYMMDDHHMM.log`.

Important: The renamed log files are never deleted. To free space, you must manually delete old log files.

BESAdmin log

This log is written by the BigFix Administration Tool (BESAdmin) and its default name is `BESAdminDebugOut.txt`. The BigFix Server installer ships with BESAdmin and uses it to perform preinstall and pre-upgrade checks. During a fresh installation or an upgrade, the installer deploys BESAdmin and uses it to conduct several configuration or upgrade operations. BESAdmin is the tool to enact configuration, maintenance, and management tasks on a BigFix Server and its database. Moreover, BESAdmin can manage certificates for the BigFix Explorer and WebUI. The BESAdmin log can help troubleshoot issues relative to: installation, upgrade, database maintenance, certificate rotation, etc.

On Windows, the settings to configure this log are stored in the registry key `HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\BigFix\EnterpriseServer\BESAdmin` and each setting is represented by a registry value.

On Linux, the same settings are stored in the configuration file `/var/opt/BESServer/besserver.config`, in the section `[SOFTWARE\BigFix\Enterprise Server\BESAdmin]` and each setting is represented by a simple key-value pair.

- `DebugOut` (`REG_SZ`), the full file path of the log, default depends on OS
- `EnableLogging` (`DWORD`), 0 to disable the log, 1 to enable it, defaults to 1
- `EnabledLogs` (`REG_SZ`), the types of information to log, a subset of "critical;debug;database;performance", defaults to "critical;debug;database"
- `LogFileSizeLimit` (`DWORD`), the maximum log file size in bytes before rotation occurs, defaults to 10485760 (10 MB)

BESAdmin audit log

This log contains the audit events recorded by the BigFix Administration Tool (BESAdmin) and its name is always `besadmin_audit.log`.

The settings to configure this log are stored as BigFix Client settings.

- `value` (`REG_SZ`) of `_BESAdminAudit_Logging_LogDirectoryPath`, the path of the folder containing the log file, default depends on OS and user that ran the program
- `value` (`REG_SZ`) of `_BESAdminAudit_Logging_LogMaxSize`, the maximum log file size in bytes before rotation occurs, defaults to 104857600 (100 MB)

When the size limit is reached, the log file is renamed following the pattern `besadmin_audit.YYYYMMDDHHMM.log`.

Important: The renamed log files are never deleted. To free space, you must manually delete old log files.

Important: On Windows, when the audit log files for the BigFix Administration Tool are generated in their default location, these files cannot be deleted from the Audit Cleaner Removes old audit log files function when the BigFix Administration Tool is scheduled or is launched with a Windows user different from the one with which the audit log files have been generated.

BESTools log

This log records the execution of BigFix Server maintenance tools andits default name is `BESTools.log`. In the past, BES Tools was a set of programs used to perform clean up operations on the BigFix Server and its database. Those tools, like the BES Computer Remover, are now part of the BigFix Administration Tool (BESAdmin). The same log is used regardless of whether those tools are run directly via the BESAdmin interface or as scheduled activities.

The settings to configure this log are stored as BigFix Client settings.

- `value` (`REG_SZ`) of `_BESTools_Logging_LogMaxSize`, the maximum log file size in bytes before rotation occurs, defaults to 5242880 (5 MB)
- `value` (`REG_SZ`) of `_BESTools_Logging_LogPath`, the full file path of the log, default depends on OS

Console log

This log is written by the BigFix Console.

On Windows, the settings to configure this log are stored in the registry key `HKEY_CURRENT_USER\Software\BigFix\Enterprise Console` and each setting is represented by a registry value.

- `DebugOut` (`REG_SZ`), the full file path of the log, defaults to empty
- `EnableLogging` (`DWORD`), 0 to disable the log, 1 to enable it, defaults to 1
- `EnabledLogs` (`REG_SZ`), the types of information to log, either "all" or a subset of "critical;debug;performance;timing", defaults to "critical;debug"

No logging occurs if `DebugOut` is empty or missing. On a fresh installation, that setting is created empty, so the Console log is disabled by default.

Fixlet Debugger log

This log is written by the Fixlet Debugger.

On Windows, the settings to configure this log are stored in the registry key `HKEY_CURRENT_USER\Software\BigFix\FixletDebugger` and each setting is represented by a registry value.

- `DebugOut` (`REG_SZ`), the full file path of the log, defaults to empty
- `EnableLogging` (`DWORD`), 0 to disable the log, 1 to enable it, defaults to 1
- `EnabledLogs` (`REG_SZ`), the types of information to log, either "all" or a subset of "critical;debug", defaults to "debug"

No logging occurs if `DebugOut` is empty or missing. On a fresh installation, that setting is created empty, so the Fixlet Debugger log is disabled by default.

Relay log

This log is written by the BESRelay service and its default name depends on the OS. The BigFix Relay acts as a middleman between the BigFix Server and a group of BigFix Clients, creating a hierarchical network structure and taking on part of the Server workload. Multiple BigFix Relays may be chained to create a deeper hierarchy. The BESRelay service forwards BigFix content and administrative commands from the Server to the Clients, and forwards Client reports back to the Server. It can distribute a single file to multiple Clients. The Relay log can help troubleshoot issues relative to: file distribution, network traffic forwarding, TLS certificates, etc.

The settings to configure this log are stored as BigFix Client settings.

They are the same settings used to configure the Root Server log.

Web Reports server log

This log is written by the BESWebReportsServer service. Web Reports can aggregate, filter, and elaborate data from one or more BigFix Servers. It provides dashboards for real-time monitoring and the ability to generate reports about the whole BigFix deployment and the devices in it. Web Reports users can access its UI and design custom reports to show the data they want. The Web Reports log can help troubleshoot issues relative to: Web Reports database connection, communication with BigFix Servers, data aggregation, and report creation.

On Windows, the settings to configure this log are stored in the registry key `HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\BigFix\EnterpriseServer\BESReports` and each setting is represented by a registry value.

On Linux, the same settings are stored in the configuration file `/var/opt/BESServer/beswebreports.config`, in the section `[SOFTWARE\BigFix\Enterprise Server\BESReports]` and each setting is represented by a simple key-value pair.

- `Debug` (`REG_SZ`), 0 to exclude details about errors occurring in the Web Reports UI, 1 to print them, defaults to 0
- `EnabledLogs` (`REG_SZ`), the types of information to log, either "all" or a subset of "database;critical;debug;memory;performance;timing;store_usage", defaults to "critical;debug;memory;store_usage"
- `LogFileSizeLimit` (`DWORD`), the maximum log file size in bytes before rotation occurs, defaults to nothing (unlimited)
- `LogOn` (`DWORD`), 0 to disable the log, 1 to enable it, defaults to 1
- `LogPath` (`REG_SZ`), the full file path of the log, defaults to empty

No logging occurs if `LogPath` is empty or missing. On a fresh installation of Version 11.0.5 or earlier, that setting is created empty, so the Web Reports log is disabled by default.

The maximum number of rotating files used for this log is 10 and it is not configurable.

BigFix Explorer log

This log is written by the BESExplorer service and its default name is `BESExplorer.log`.

The settings to configure this log are stored as BigFix Client settings.

- `value` (`REG_SZ`) of `_BESExplorer_Logging_EnableLogging`, 0 to disable the log, 1 to enable it, defaults to 1
- `value` (`REG_SZ`) of `_BESExplorer_Logging_EnabledLogs`, the types of information to log, a subset of "critical;debug;database;performance", defaults to "critical"
- `value` (`REG_SZ`) of `_BESExplorer_Logging_LogMaxSize`, the maximum log file size in bytes before rotation occurs, defaults to 5242880 (5 MB)
- `value` (`REG_SZ`) of `_BESExplorer_Logging_LogPath`, the full file path of the log

BigFix MCP Server log

These logs are written by the `bes-mcp-server` service. The MCP Server utilizes two primary logs: the application log (default name `BESMCPServer.log`) and the audit log (`mcp_audit.log`).

The settings to configure the application log are stored in the `config.yaml` file located in the MCP Server data directory:

- `log.file_path`: the full file path of the log to enable file logging.
- `log.max_size_mb`: the maximum log file size in megabytes before rotation occurs, defaults to 100.
- `log.max_backups`: the maximum number of old log files to retain, defaults to 3.
- `log.max_age_days`: the maximum number of days to retain old log files, defaults to 28.
- `log.compress`: a boolean value indicating whether the rotated log files should be compressed, defaults to false.

Note: Changes to the `config.yaml` file require a restart of the `bes-mcp-server`service to take effect.

Plugin Portal log

This log is written by the BESPluginPortal service and its default name is `BESPluginPortal.log`.

The settings to configure this log are stored as BigFix Client settings.

- `value` (`REG_SZ`) of `_BESPluginPortal_HTTPServer_LogFilePath`, the full file path of the log, defaults to empty
- `value` (`REG_SZ`) of `_BESPluginPortal_HTTPServer_LogFileRotationLimit`, the maximum number of files used for log rotation, defaults to 10
- `value` (`REG_SZ`) of `_BESPluginPortal_HTTPServer_LogFileSizeLimit`, the maximum log file size in bytes before rotation occurs, defaults to 52428800 (50 MB)

- `value` (`REG_SZ`) of `_BESPluginPortal_Log_EnabledLogs`, the types of information to log, either "all" or a subset of "critical;debug;events;timing", defaults to "all"
- `value` (`REG_SZ`) of `_BESPluginPortal_Log_Verbose`, 0 to disable verbose logging, 1 to enable it, defaults to 0

For more information, see BigFix MCM Logging.

Client log

The BigFix Client can keep several logs.

The settings to configure them are stored as BigFix Client settings.

For more information, see Data Collection: BigFix Client.

Format of the audit log messages

Audit entries are presented in a single line and contain the same number of field delimiters. Field delimiters are present even if no value exists for a specific field. Since the format of the audit fields is subject to change over time, each line has a version number as the first entry.

The audit log messages are in the following format:

```
<format-version>|<timestamp>|<message-priority>|<username>|<event-source>|<
event-label>|<event-type>|<ip-address>|<message>
```

"|" is the field separator.

- `format-version`: The version of the message format. For example, *1*.
- `timestamp`: The timestamp of the log message, which can be the server timezone or UTC.
- `message-priority`: The priority of the log. Possible values are:
  - EMERG (emergency, system non-functioning or unusable)
  - ERROR (error condition)
  - WARN (warning)
  - INFO (informational message)

- `username`: The username of the event initiator. In case it is not a user event, then the field is set to *SYSTEM*.
- `event-source`: The source from which the event originates.

  Possible values are: CONSOLE, RESTAPI, WEBUI, WEBREPORTS, PLUGINPORTAL, ADMINTOOL, EXPLORER.
- `event-label`: The event or the artifact that is affected.

  Possible values are: USER, SITE, ACTION, ROLE, COMPUTER, AUTHZ, SETTING, DATABASE, FIXLET, TASK, ANALYSIS, BASELINE, COMPUTERGROUP, TOKEN.
- `event-type`: The type of the event.

  Possible values are: CREATE, DELETE, UPDATE, LOGIN, LOGOUT, SEARCH.
- `ip-address`: The IP address of the component which initiated the event request. For SYSTEM, this is the server IP address.
- `message`: The actual log message.

Starting with BigFix version 9.5.11, the server audit logs include also the following items:

- Messages for deletion of computers from the console or through API.
- Messages for deletion of BigFix Actions.

Audit log examples

Following are a few examples of the log messages in the new format:

```
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
```

In case of audit entries other than those introduced in 9.5.11 or later, the messages are formatted as follows: `<format-version>|<timestamp>|<message-priority>||||||<message>`. For example:

```
1|Tue, 05 Sep 2017 10:57:06 -0700|INFO||||||user "johndoe" (1): Successful log in. (Data Connection)
```

Starting from BigFix 11.0.5, we introduced new audit logs on the BigFix Server related to the following activities:

- Update Site Subscription (computers)
- Create/Update/Delete Custom Sites
- Create/Update/Delete Computer Groups
- Create/Update/Delete Content (Fixlets, Tasks, Analyses, Baselines)
- Globally Hide/Unhide Content.

Here are some examples of the new audit logs introduced:

When creating a new Fixlet:

```
1|Wed, 04 Jun 2025 18:18:17
 +0200|INFO|BFAdmin|CONSOLE|FIXLET|CREATE|fe80::c3c3:14d0:994c:9467|
fixlet "Fixlet_Name" (Fixlet_ID) created
```

When modifying a Task:

```
1|Wed, 04 Jun 2025 18:18:17
 +0200|INFO|BFAdmin|CONSOLE|TASK|UPDATE|fe80::c3c3:14d0:994c:9467|
task "Task_Name" (Task_ID) modified
```

When deleting a Baseline:

```
1|Wed, 04 Jun 2025 18:18:17
 +0200|INFO|BFAdmin|CONSOLE|BASELINE|DELETE|fe80::c3c3:14d0:994c:9467|
baseline "Baseline_Name" (Baseline_ID) removed
```

When setting to hide/unhide globally Content:

```
1|Wed, 04 Jun 2025 18:41:22
 +0200|INFO|BFAdmin|CONSOLE||UPDATE|fe80::c3c3:14d0:994c:9467|
Fixlet Id "ID" hide Globally
1|Wed, 04 Jun 2025 18:41:24 +0200|INFO|BFAdmin|CONSOLE||UPDATE|
```

```
fe80::c3c3:14d0:994c:9467|Fixlet Id "ID" unhide Globally
```

Chapter 28. List of advanced options

The following lists show the advanced options.

Advanced options that you can specify in the Advanced Options tab of the BigFix Administration Tool on Windows (BESAdmin.exe), or when running the `BESAdmin.sh` script on Linux using the following syntax:

```
./BESAdmin.sh -setadvancedoptions -sitePvkLocation=<path+license.pvk>
[-sitePvkPassword=<password>]
{ -list | -display
| [ -f ] -delete option_name
| [ -f ] -update option_name=option_value }
```

Note: The notation `<path+license.pvk>` used in the command syntax stands for *`path_to_license_file`*`/license.pvk`.

These options are typically supplied by your HCL Software Support.

Advanced options for disabling functions

Use these options if you want to disable specific capabilities on the console.

disableNmoSiteManagementDialog

If set to "1", the site management dialog is unavailable to non-master operators (NMOs).

disableNMOComments

If set to "1", NMOs cannot add comments. NMOs will still be able to view comments.

disableNmoManualGroups

If set to "1", NMOs cannot add or remove computers from manual groups, and see manual groups that none of their computers are members of.

disableGlobalRelayVisibility

If set to "1", NMOs cannot see relays in the relay-selection drop-downs in the console that don't belong to them. The exception is if they view a machine that is currently configured to report to a relay not administered by them, in this case that relay appears in the list as well.

disableNmoRelaySelModeChanges

If set to "1", NMOs cannot toggle automatic relay selection on and off.

disableDebugDialog

If set to "1", the keyboard sequence CTRL-ALT-SHIFT-D cannot be used to open up the console's debug dialog.

disableComputerNameTargeting

If set to "1", the third radio option "target by list of computer names" is removed on the targeting tab of the take action dialog.

allowOfferCreation

If set to "0", the 'Offer' tab in the Take Action Dialog is disabled. Offer presets in Fixlets are ignored by the console.

disableNmoCustomSiteSubscribe

If set to "1", the "Modify Custom Site Subscriptions" menu item is disabled for all NMOs

Advanced options for password policies

Use these settings to enforce password policies in your BigFix environment.

passwordComplexityRegex

Specifies a *perl-style* regular expression to use as a password complexity requirement when choosing or changing operator passwords. These are some examples:

- Require a 6-letter or longer password that does not equal the string 'bigfix'.

```
(?![bB][iI][gG][fF][iI][xX]).{6,}
```

- Require a 6-letter or longer password containing lowercase, upper case, and punctuation.

```
(?=.*[[:lower:]])(?=.*[[:upper:]])(?=.*[[:punct:]]).{6,}
```

- Require an eight-character or longer password that contains 3 of the following 4 character classes: lowercase, uppercase, punctuation, and numeric.

```
((?=.*[[:lower:]])(?=.*[[:upper:]])(?=.*[[:punct:]])|
(?=.*[[:lower:]])(?=.*[[:upper:]])(?=.*[[:digit:]])|
(?=.*[[:lower:]])(?=.*[[:digit:]])(?=.*[[:punct:]])|
(?=.*[[:digit:]])(?=.*[[:upper:]])(?=.*[[:punct:]])).{8,}
```



Note: The Site Administrator passwords are not affected by this complexity requirement.

passwordComplexityDescription

Specifies a description of the password complexity requirement. This string is displayed to the user when a password choice fails the complexity requirements set using the **passwordComplexity** option. An example of password complexity description is "Passwords must have at least 6 characters." If you do not set this value but you set **passwordComplexityRegex** setting, the description set in **passwordComplexityRegex** is displayed to the user.

passwordsRemembered

Specifies the number of unique new passwords that can be set for an user account before an old password can be reused. The default value is "0".

maximumPasswordAgeDays

Specifies the number of days that a password can be used before the system requires the user to change it. The default value is "0" (no maximum).

minimumPasswordLength

Specifies the least number of characters that a password for a user account can contain. The default value is "6". This is an usage example of this option:

```
./BESAdmin.sh -setadvancedoptions -sitePvkLocation=LOCATION
-sitePvkPassword=PASSWORD -update minimumPasswordLenth=9
```

enforcePasswordComplexity

If set to '1' or 'true', the passwords must meet the following minimum requirements:

- They must not contain the user's account name or parts of the user's full name that exceed two consecutive characters.
- They must be at least six characters long.
- They must contain characters from three of the following four categories:

```
English uppercase characters (A through Z)
English lowercase characters (a through z)
Base 10 digits (0 through 9)
Non-alphabetic characters (for example, !, $, #, %)
```

If you specify also the **minimumPasswordLength** setting, then the effective minimum password length will be the higher value between six and the value of **minimumPasswordLength**.

Complexity requirements are enforced when passwords are changed or created. The default value is "0".

accountLockoutThreshold

Specifies the number of incorrect logon attempts for a user name before the account is locked for **accountLockoutDurationSeconds** seconds. The default value is "5".

accountLockoutDurationSeconds

Specifies the number of seconds that an account gets locked after **accountLockoutThreshold** failed log on attempts. The default value is "1800".

Note: Web Reports has similar password controls, but they have to be set separately ('Users'->'User Options').

Advanced options for targeting restrictions

Use these advanced options to specify the targeting restrictions globally. If you to set them for a specific user, add those settings in the registry key of the BigFix Console computer under the hive `HKEY_CURRENT_USER\Software\BigFix\Enterprise Console\Targeting` as a DWORD.

The options listed in the following table take effect only if the corresponding registry keys are not set on the consoles or if the keys are set to the default values.

targetBySpecificListLimit

Specifies the maximum number of computers that can be targeted by individual selection. The default value is 10000.

Note: This option is also used to set the target limit for the REST API request. If the restriction is exceeded, the Server responds to the request with error code "413 - Content too large".

targetBySpecificListWarning

Specifies the threshold for the number of computers that can be targeted by individual selection before the console displays a warning message. The default value is 1000.

targetByListSizeLimit

Specifies the maximum number of bytes that can be supplied when targeting by textual list of computer names. The default value is 100000.

Here is the correspondence between the name of the advanced option and the name of the related registry setting:

```
targetBySpecificListLimit => SpecificListLimit
targetBySpecificListWarning => SpecificListWarning
targetByListSizeLimit => ByListSizeLimit
```

The following example restricts to 9000 = 0x2328 the SpecificListLimit setting (correspondent to the targetBySpecificListLimit advanced option):

```
{[HKEY_CURRENT_USER\Software\BigFix\Enterprise Console\Targeting]
"SpecificListLimit"=dword:00002328}
```

 **Note:** Do not increase the default values.

Advanced options for authentication

Use these settings to manage user authentications to the console.

loginTimeoutSeconds

Specifies the amount of idle time in seconds before the console requires the user to authenticate again to take certain actions. The timer is reset every time the user authenticates or does an action that would have required authentication within the idle time threshold. The default value is infinity.

loginWarningBanner

Specifies the text to show to any user after he/she logs into the Console or Web Reports. The user must click **OK** to continue. This is a usage example of this option:

```
./BESAdmin.sh -setadvancedoptions
 -sitePvkLocation=/root/backup/license.pvk
-sitePvkPassword=pippo000 -update loginWarningBanner='new messag
e'
```

timeoutLockMinutes

Specifies how many idle time minutes must elapse before the console requires to authenticate again. This setting is different from **loginTimeoutSeconds** because **timeoutLockMinutes** hides the entire console to prevent any other user to see or use it. The idle time refers to the lack of any type of input to the session including key buttons, mouse clicks, and mouse movements.

This option does not take any effect on the console if an operator accesses it using the Windows session credentials (Windows authentication).

timeoutLogoutMinutes

Specifies how many idle time minutes must elapse before the console is closed. This setting is different from **loginTimeoutSeconds** and **timeoutLockMinutes**, because **timeoutLogoutMinutes** closes the console completely. The idle time refers to the lack of any type of input to the session including key buttons, mouse clicks, and mouse movements.

This option was introduced with BigFix V9.5.11.

Note: Non efficient mime advanced option is no longer supported by the BigFix V9.5 server. Existing actions continue to run on clients but the server is no longer able to generate non efficient mime actions.

Advanced options for customizing BigFix Query

You can optionally set some parameters to customize the BigFix Query feature.

To avoid using too much space available in the database to store the BigFix Query requests and their results, you can customize the following advanced option in the BigFix Administration Tool on the BigFix server:

queryHoursToLive

Determines how many hours the BigFix Query requests are kept in the database. The default value for this option is **1440**, which corresponds to 60 days. Valid values are from 0 to 8760, that means 1 year.

queryResultsHoursToLive

Determines how many hours the BigFix Query results are kept in the database. The default value is **4** hours, and the valid values are from 1 to 336 (two weeks). If you enter value that lies outside this range, the default value is used.

queryPurgeBatchSize

The entries in the database that represent requests and results for which **queryHoursToLive** or **queryResultsHoursToLive** elapsed, are deleted from the database in buffers. This advanced option determines the number of database entries contained in each of these buffers. The default value for this option is **100000** bytes, which means 100 KB.

Other advanced options

Use these options to customize other aspects of your BigFix environment.

automaticBackupLocation

If set to an existing path, accessible both by `root` and by the database instance owner, by default `db2inst1`, this option enables the BigFix Server to run automatically the backup of the `BFENT` and `BESREPOR` databases before and after running the upgrade process.

This option is available only for Linux BigFix Servers V9.5.3 and later.

For more information, see Automatic DB2 databases backup upon upgrade.

clientIdentityMatch

This advanced option can help you to avoid having duplicate computer entries when the endpoints are detected as possible clones by the BigFix Server. The BigFix Server can use the existing computer information to try to match the identity of a Client and reassign the same `ComputerID` to computers that might have been rolled back or restored.

If **clientIdentityMatch=0**, the BigFix Server performs strict clone detection. This means that, if the BigFix Server receives a registration request from a Client that was rolled back or restored, the Server invalidates the old `ComputerID`, resets the old Client definition, and assigns a new `ComputerID` to the registering Client. This is the default behavior and is the same way the BigFix Servers earlier than V9.5.7 operate.

If **clientIdentityMatch=100,** the BigFix Server performs an additional check before assigning a new `ComputerID` to a registering Client to avoid creating cloned computer entries. This means that the BigFix Server tries to determine if the information about the rolled-back Client sufficiently matches the data held for that `ComputerID`. If the identity of the Client is matched, the Client keeps using the old `ComputerID` and its identity is not reset.

For more information, see Avoiding duplicates when a Client is restored.

includeSFIDsInBaselineActions

If set to "1", it requires the console to include source Fixlet IDs when emitting baseline actions. Emitting these IDs is not compatible with 5.1 clients.

defaultHiddenFixletSiteIDs

This option allows to selectively change the default Fixlet visibility on a per-site basis. It only takes effect when global default Fixlet hiding is not in use. You specify a comma-separated list of all the site IDs to be hidden by default. The list of sites IDs is in the SITENAMEMAP table in the database. If among the sites to be hidden is listed also the siteID for the 'BES Support' site, the dynamic content becomes no longer usable.

defaultOperatorRolePermissions

This option allows you to change the default permissions that apply when you create operators and roles. It can take the following values:

- 0: Operators and roles are created with the default permissions that applied until BigFix V9.5.10.
- 1: Operators and roles are created with minimum default permissions. The same default settings apply even when you do not set any value.
- 2: Operators and roles are created with minimum default permissions as in the previous case, except that **Show Other Operators' Actions** is set to **Yes** and **Unmanaged Assets** is set to **By Scan Point** (for operators). In the case of roles, however, **Unmanaged Assets** is always set to **Show None**. The **Access Restriction** for the operators is set to **Always allow this user to log in**. The login privilege **Can use Console** is set to **Yes** both for operators and roles.

This option was introduced with BigFix V9.5.11.

enableRESTAPIOperatorID

This option allows you to display operator resource URLs with the operator ID instead of the operator name. For example, `https://BigFix_Server_URL:52311/api/operator/<Operator_ID>`. To enable the option, set it to true or 1.

This option was introduced with BigFix V9.5.10.

showSingleActionPrePostTabs

If set to "1", the 'Pre-Action Script' and 'Post-Action Script' tabs of the Take Action Dialog shows up even on single actions.

propertyNamespaceDelimiter

Specifies the separator for retrieved properties. By default, retrieved properties are separated into namespaces by the character sequence '::'. The character sequence used to indicate a separator can be changed using this deployment option.

DefaultFixletVisibility

If set, this option allows you to specify either to make Fixlets, tasks and analysis gathered from external sites globally visible or to make them globally hidden. By default, they are globally visible to all Console operators.



Note: On Windows platforms only, this option is also available in the "System option" tab of the BigFix Administration Tool.

MinimumRefreshSeconds

If set, this option allows you to specify the minimum amount of time after which console operators are allowed to set their automatic refresh interval. This amount of time is specified in seconds. By default, it is set to 5 seconds.



Note: On Windows platforms only, this option is also available in the "System option" tab of the BigFix Administration Tool.

minimumConsoleRequirements

Specifies if the minimum requirements that must be satisfied by the machines running the database that the console connect to. Its value consists of a comma separated list of one or more of the following requirement strings:

"RAM:<min MB MO ram>/<min MB NMO ram>"

Requires that the console runs on a machine with at least the specified amount of physical RAM. Two different values must be supplied; one for master operators and another for non-master operators. Both values must be less than 2^32. For example, "RAM:2048/1024" .

"ClientApproval"

States that the BES Client must determine if a machine is suitable for login. A machine is considered suitable for login if one of the following settings is specified locally:

- **"moConsoleLoginAllowed"**
- **"nmoConsoleLoginAllowed"**

The console must run as an account with permissions to read the client registry keys stored under HKEY_LOCAL_MACHINE to log in when using the **"ClientApproval"** option.

disableRunningMessageTextLimit

If set to a value other than "0", the console users can enter more than 255 characters in the running message text in the Take Action Dialog.

useFourEyesAuthentication

If set to "true", you can set the approvers for user actions in console user document. The approver must confirm the action on the same console where the user is logged on.

masterDatabaseServerID

By default, the database with server ID 0 is the master database. This is the database that BESAdmin needs to connect to. Use this option to change the master database to a different machine.

enableWakeOnLAN

If set to "1", the console shows the "right click WakeOnLAN" functionality in the computer list. By default the functionality is not shown.

enableWakeDeepSleep

If set to "1", the console shows the "right click Send BESClient Alert Request" functionality in the computer list. By default the functionality is not shown. During Deep sleep, all UDP messages except this specific wake up message are ignored.

requireConfirmAction

If set to "1", every time an action is taken a confirmation pop-up window with a summary of the action details is displayed. The information listed in the pop-up window is:

```
Action Title
Estimated endpoints targeted
Start time
End time
Originated by or Source
```

The summary lists the need of doing a restart or a shutdown as well, if the action requires it. By default the confirmation window is not displayed.

Note: When you enable this option, the displayed value for the **Estimated targeted computers** might not be correct, if you performed the action from a wizard of a BigFix Application such as, for example, Server Automation or OSD.

You must restart the BigFix Console after configuring this option.

blockIncomingMCPRequests

If set to "1" (enabled), requests identified as MCP requests are rejected with HTTP 403. The BigFix Root Server identifies an MCP request by the presence of the `BES-MCP-Identifier` HTTP header. For backward compatibility, the legacy `X-BES-MCP-Identifier` header is also recognized. The BigFix Root Server must be restarted after changing this setting.

Chapter 29. Security Configuration Scenarios

Starting from Version 11, BigFix provides the capability to configure several security options.

SHA-384

The first setting enforces SHA-384 as the only hashing algorithm for digital signatures. You can set this security option only after you install or upgrade all the BigFix components to Version 11.

Note: When you set this option you configure a very restricted security environment. You can enable or disable this security setting at any time by running the BigFix Administration tool. For additional information, see Security.

In a complex environment, you can enforce SHA-384, only after all the DSA servers are upgraded to BigFix V11 or above and have a new license.

SHA-256 Downloads

You can set a check for verifying the file download integrity using the SHA-256 algorithm. If you do not set this option, the file download integrity check is run using the SHA-1 algorithm.

TLS 1.3

Select this option to require only the TLS 1.3 communication among the BigFix components. If this option is not selected, the BigFix server and relay will accept also the TLS 1.2 communication.

For additional information, see Security.

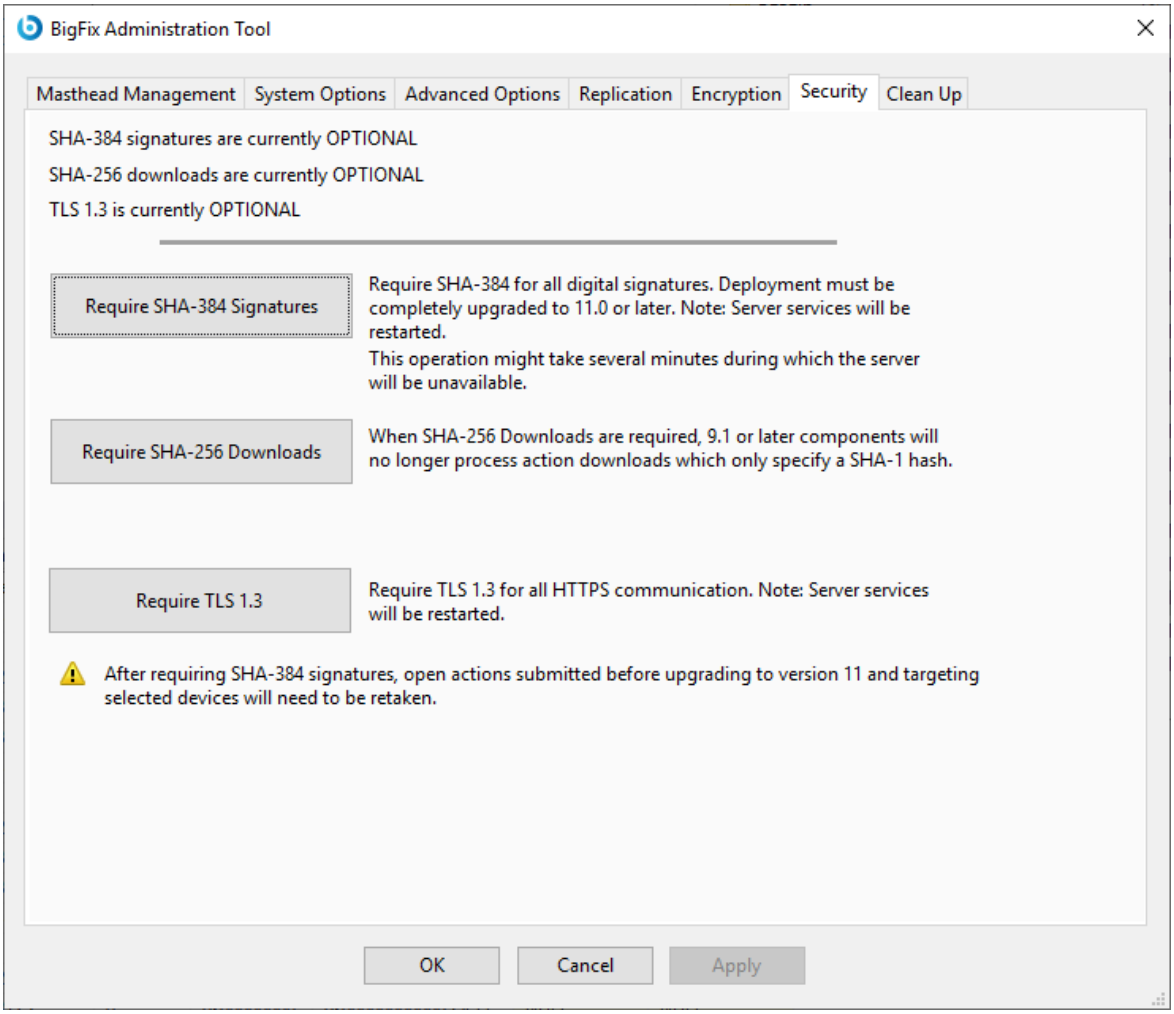
FIPS 140-2 or FIPS 140-3

To achieve compliance with the FIPS 140-2 or FIPS 140-3 standard, perform the steps described in Editing the Masthead on Windows systems (on page 1114) and Editing the Masthead on Linux systems (on page 1119).

On Windows Systems

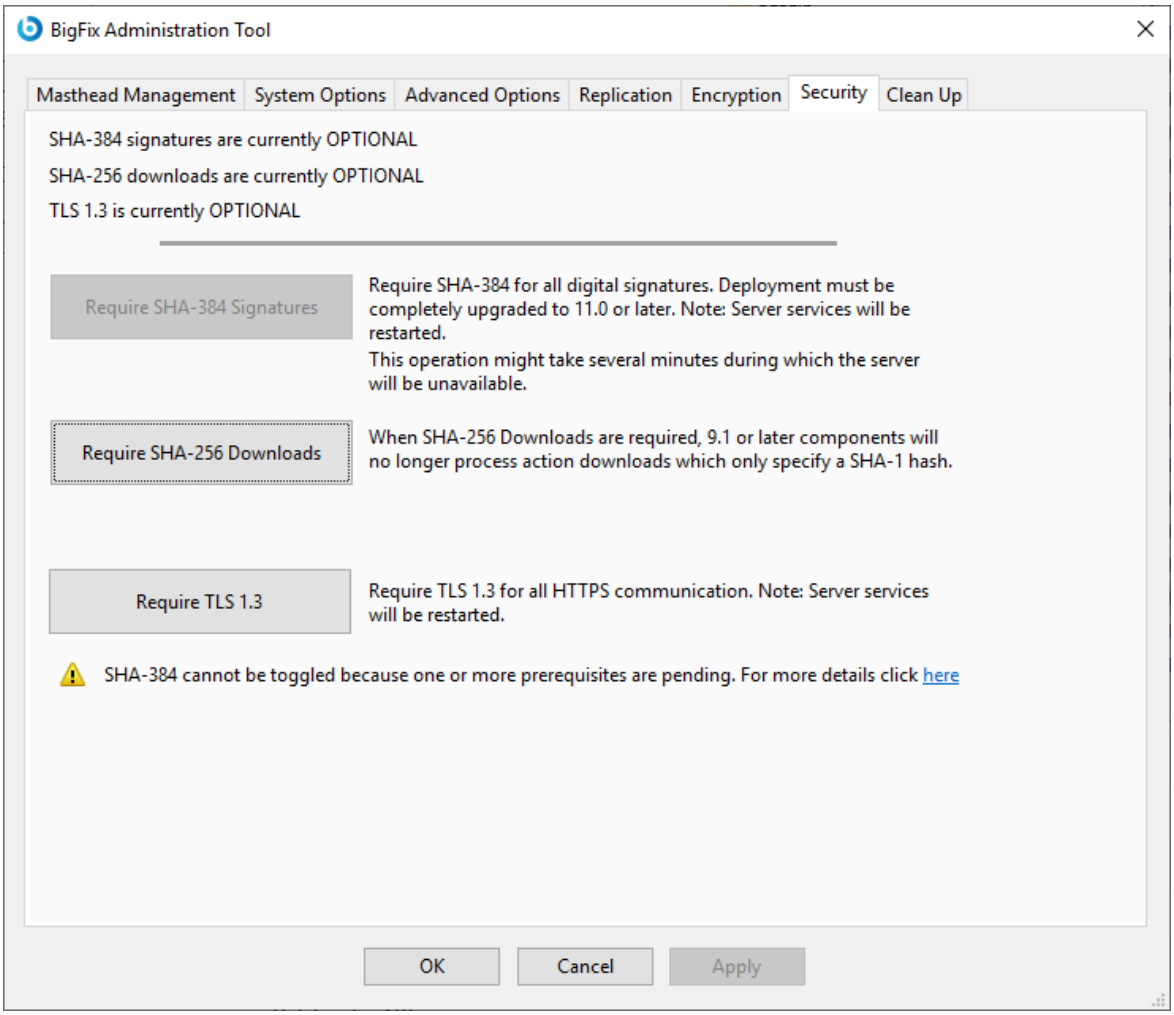
You can enforce the SHA-384 for all digital signatures by performing the following steps:

1. Run the Administration Tool by clicking **Start > All Programs > BigFix > BigFix Administration Tool**.
2. Browse to the location of your site license (`license.pvk`) and click **OK**.
3. Select the **Security** tab. The following window is displayed:

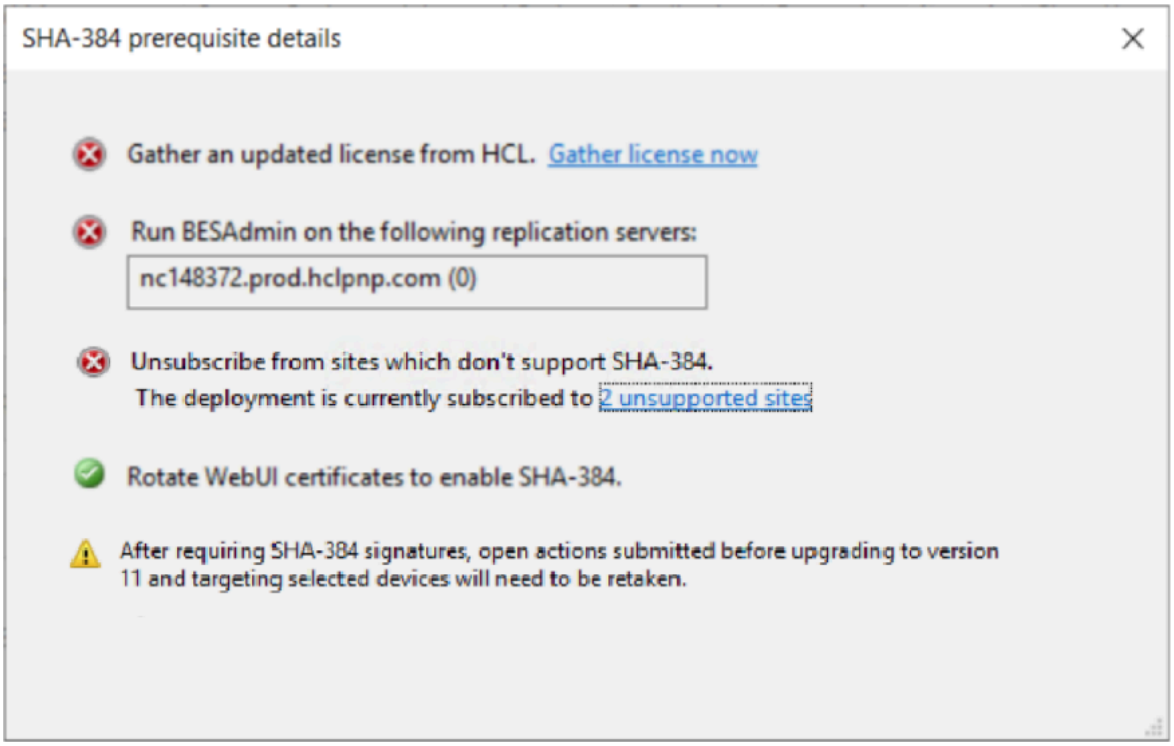


You can now enforce the SHA-384 for all digital signatures.

If the SHA-384 option is greyed out, a warning at the bottom of the page will inform you about the issues.



When clicking the link in the warning, a pop up will be displayed with a list of conditions that must be satisfied in order to be able to enforce SHA-384.

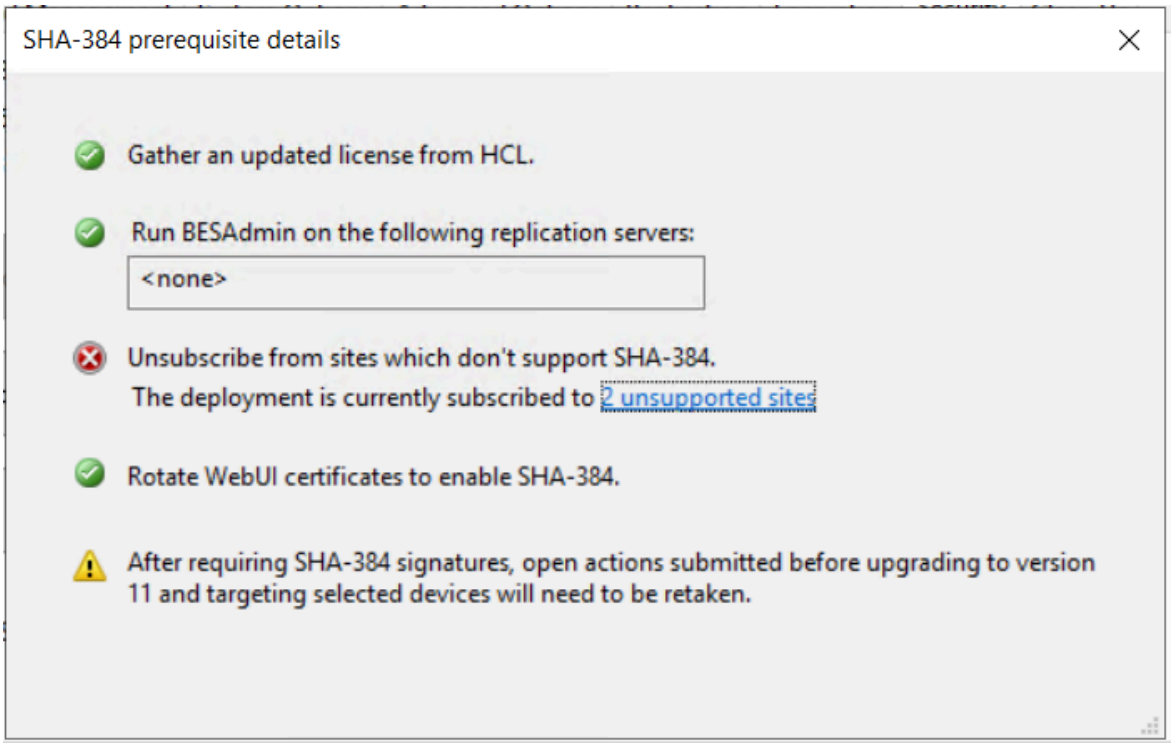


If the **Gather an updated license from HCL** is not satisfied, click Gather license now so that your updated license is gathered from the HCL site and is distributed to the BigFix clients. This step ensures that you use the updated license authorizations if you specified an existing license file during the installation steps.

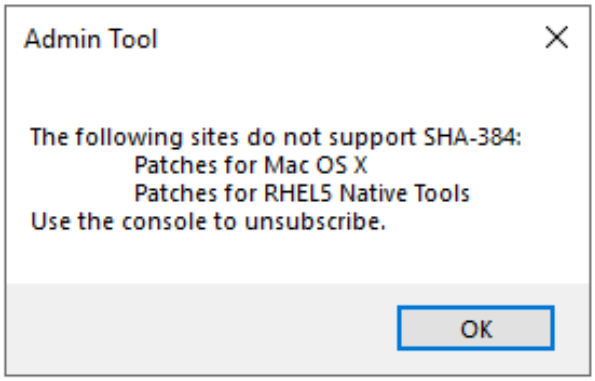
If the **Run BESAdmin on the following replication servers** is not satisfied, it means that one of these conditions regarding all the BigFix servers involved in a Disaster Server Architecture (DSA) is not satisfied:

- They are at Version 11.
- They have the updated license.

If the **Unsubscribe from sites which do not support SHA-384** is not satisfied, click the link to see the list of sites which do not support SHA-384. To solve the issue you need to unsubscribe these sites from the BigFix Console.



In this example, 2 sites must be unsubscribed.



If the **Rotate WebUI certificates to enable SHA-384** is not satisifed, you need to regenerate all WebUI certificates created before upgrading to Version 11.

SHA-256 Downloads

To ensure that data has not changed after you download it using the SHA-256 algorithm, click **Require**. If you do not select this option, the integrity check of the downloaded files is run using the SHA-1 algorithm.

On Linux Systems

You can enforce SHA-384 after you install BigFix V11 or upgrade it to V11, by running the following command as super user:

```
./BESAdmin.sh -securitysettings -sitePvkLocation=<path+license.pvk>
-sitePvkPassword=mypassw0rd -requireSHA384Signatures
```

Note: The notation `<path+license.pvk>` used in the command syntax stands for *`path_to_license_file`*`/license.pvk`.

The full syntax of the `./BESAdmin.sh -securitysettings` is the following:

```
./BESAdmin.sh -securitysettings -sitePvkLocation=<path+license.pvk>
   [-sitePvkPassword=<password>]
   { -status | {-requireSHA384Signatures |-allowSHA256Signatures }
   | {-requireSHA256Downloads|-allowSHA1Downloads} | {requireTLS13 |
 -allowTLS12 ] } }
```

where:

status

Shows the status of the security settings in your BigFix environment.

Example:

```
./BESAdmin.sh -securitysettings -sitePvkLocation=/root/backup/li
cense.pvk
-sitePvkPassword=mypassw0rd -status

SHA-384 signatures are currently OPTIONAL
```

```
SHA-256 downloads are currently OPTIONAL
TLS 1.3 is currently OPTIONAL
Masthead TLS cipher list: (TLS cipher list not set)
The visibility status of the From field is currently set to: hid
e.
The local operators are currently enabled.
```

requireSHA384Signatures | allowSHA256Signatures

Enables or disables the security option that adopts the SHA-384 cryptographic digest algorithm for all digital signatures.

requireSHA256Downloads | allowSHA1Downloads

Ensures that data has not changed after you download it using the SHA256 | SHA1 algorithm.

requireTLS13 | allowTLS12

With this option, you can require the use of TLS 1.3 for all HTTPS communication.

Chapter 30. Enabling Microsoft Control Flow Guard on BigFix Server

Starting from BigFix version 11.0.3, the BigFix Server implements the Microsoft Control Flow Guard (CFG) security feature on Windows systems; the BigFix Server executables:

- BESAdmin.exe
- BESRootServer.exe
- FillDB.exe
- GatherDB.exe

are compiled with the Control Flow Guard options.

For more details about the Microsoft Control Flow Guard feature, see Control Flow Guard for platform security.

When you install or upgrade to BigFix version 11.0.3, the Microsoft Control Flow Guard feature on the above executables is disabled by default.

You can enable the feature either using the Windows Exploit Protection as described in Configure system-level mitigations with the Windows Security app or using the new BigFix Administration Tool command line options named `checkcontrolflowguard` and `setcontrolflowguard` as described in BESAdmin Windows Command Line.

The **Use strict CFG** option is **not supported**; when enabled, Windows OS is expecting that all the libraries loaded by the executable are built with CFG parameter, producing unpredictable behaviors.

Note: The following BigFix processes do not start when the **Use strict CFG** option is active and the FIPS mode is enabled:

- BESAdmin
- FillDB

- BESRootServer
- GatherDB

Chapter 31. External Repository Sites

Up to BigFix 11.0.6, managing custom content across multiple root servers and environments presents significant challenges:

- Custom content is packaged and published separately.
- Repository history is not the site version source of truth.
- Cross-environment tracing of author and commit is difficult.

Starting with BigFix 11.0.7, integrating BigFix directly with external source control systems (Git) allows BigFix to treat repositories as "External Sites" and gather content directly at the source.

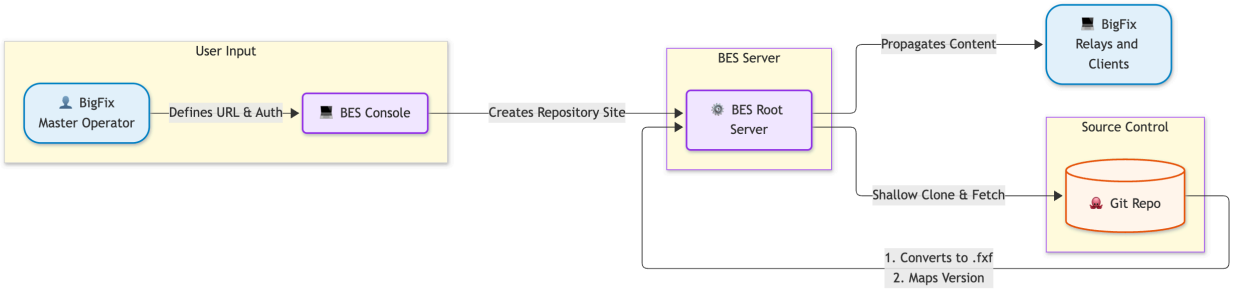
This enhancement aligns BigFix with standard DevOps practices, providing:

- **Unified Source of Truth**: Centralized versioning and clear cross-environment history tracking.
- **DevOps Integration**: Full compatibility with modern CI/CD and automation pipelines.
- **Enhanced Governance**: Improved visibility into content evolution while maintaining strict content confidentiality.

In BigFix 11.0.7, the BigFix Server treats the Git repository as the source of truth for a new site type called **External Repository Site**:

- A new External Site type recognized by a repository-specific gather URL.
- Repository URL and branch define the source identity.

With this enhancement, the BigFix Server is responsible for clone/fetch, content export, version mapping, and publication.



The content flow is as follows:

- Initial synchronization uses a shallow clone; later cycles reuse the local repository and perform incremental fetches.
- The generated site behaves like traditionally published external content: signed listing first, then file hash validation.
- A manual Gather operation triggers an immediate repository check; normal scheduling uses the configured polling period.
- Repository SSH access and known_hosts verification protect synchronization.

Before creating a site, ensure that:

- The BigFix Server and endpoints that use an External Repository Site run BigFix 11.0.7 or later.
- You have Master Operator privileges in the BigFix Console.
- The BigFix Server can reach an SSH Git repository containing BigFix site content, and the Git account has read access to it.
- The BigFix Server has the SSH credentials and trusted host keys described below.

Prepare SSH access on the BigFix Server

Repository access is read-only. The BigFix Server creates the `RepositorySiteGather/GitCredentials` directory under the BES Server storage directory at startup. Place the following files in that directory before importing the site:

```
RepositorySiteGather/
├── GitCredentials/
```

```
|   ├── id_rsa
|   ├── id_rsa.pub
|   └── known_hosts
└── Sites/  # published site data
```

1. Place an SSH private key in `id_rsa` and its matching public key in `id_rsa.pub`. Register the public key with a Git service account authorized to read the repository. For GitHub, the account needs read access to repository contents.
2. Obtain the Git host's public host key and verify its fingerprint with the Git administrator through a trusted channel. Add the verified key to `known_hosts`. Do not rely on an unverified key scan as proof of the host's identity.
3. Restrict access to the private key and ensure that the BES Server service account can read it. Confirm that the BigFix Server can reach the Git host over SSH.

BigFix 11.0.7 requires the credential filenames shown above. To generate an RSA PEM key pair on the BigFix Server, run:

```
ssh-keygen -m PEM -t rsa -b 4096 -f id_rsa -N ""
```

SSH credentials are managed on the BigFix Server, not in the BigFix Console dialog.

Create an External Repository Site in the BigFix Console

As a Master Operator, choose **Tools > Create External Repository Site...** in the BigFix Console. The dialog creates an External Repository Site masthead and imports the site. Repository SSH credentials are configured on the BigFix Server, not entered in this dialog. For details about the dialog, see Create Repository Site.

After creation, the Sites list shows the site type as **Repository**. Site details show **Repository Site**, the repository URL, and the branch. When at least one Repository Site has been gathered, License Overview shows an **External Repository Sites** tab.

Verify synchronization and gathering

After import, the BigFix Server checks the configured branch, performs an initial shallow clone, compiles and publishes the content, and the normal gather process imports it.

Subsequent checks reuse the local repository and fetch changes. Use **Gather** in the BigFix Console to check immediately instead of waiting for the scheduled poll.

Check the site details for the current version and the expected repository URL and branch. In the BigFix Server log, look for `Repository site cloned successfully` on the first sync or `Repository site updated successfully` on later syncs. In `GatherDB.log`, confirm that the site version was imported successfully. A 404 response during the first sync can be temporary until publishing finishes.

Repository Site identity and versions

The repository URL and selected branch are encoded in the gather URL for sites created through the BigFix Console:

```
bigfix://bigfix.com/repositorysite?repository_url={repository_url}&git_bran
ch={branch}
```

The Site ID is derived from the source configuration. The same repository URL and branch produce the same Site ID and gather URL across deployments; changing either changes the identity. A masthead created in the BigFix Console is trusted only in the deployment that issued it, even when the source identity matches.

The site version is the depth of the selected commit from the repository root. By default, a new commit at the selected branch HEAD results in a new published site version. If several commits arrive between checks, the published version advances to the depth of the latest selected commit and can skip intervening version numbers.

Updates are forward-only. To restore earlier content, create a new Git revert commit instead of rewinding the branch. If history is rewritten by a force-push, plan a site deletion and reimport to clear the BigFix Server's local synchronization state; do not expect the existing site to roll back automatically.

Note: You cannot create two sites with the same repository URL and branch.

Git Repository

Repository layout and content compilation

Organize the Git source repository as follows:

```
my-repository/
├── Fixlets/
|   ├── Analyses/
|   |   └── 1300- Some Analysis.bes
|   ├── Fixlets/
|   |   ├── FixletType1/
|   |   |   ├── digest.xml
|   |   |   └── 2110- Some Fixlet.bes
|   |   └── FixletType2/
|   |       └── 2210- Some Fixlet.bes
|   └── Tasks/
|       └── 3300- Some Task.bes
├── NonClientFiles/
└── OtherFiles/
```

`Fixlets` contains BigFix `.bes` files for Fixlets, Tasks, and Analyses. Each first-level directory beneath `Fixlets` is compiled into an `.fxf` file with the same name. The next directory level groups content into Digests within that file. For example, `Fixlets/Fixlets` becomes `Fixlets.fxf`, with `FixletType1` and `FixletType2` as Digests. A `digest.xml` applies to the directory where it is located.

Start content filenames with a numeric ID and a hyphen (space may be present or not), for example `2110- Some Fixlet.bes`. The numeric prefix becomes the displayed content ID. Files without a numeric ID or with duplicate IDs are not gathered and generate warnings in the BigFix Server log. Out-of-range IDs also generate export warnings and are not preassigned.

`NonClientFiles` holds server-side assets, scripts, and metadata; `OtherFiles` holds additional files gathered by the client. Both directories are optional.

Optional repository configuration files

The following XML files are optional. Place `siteConfig.xml` and `site.xml` at the repository root. Place `digest.xml` at the top of the directory whose content it configures. It can appear at `Fixlets/` or within a content directory.

- `siteConfig.xml` overrides the default directory names by using `BesFileDir`, `NonClientFilesDir`, or `OtherFilesDir`. Each element is optional and can occur once.
- `site.xml` defines optional site-wide `Relevance` applied to Fixlets, Tasks, Analyses, and computer groups in the site. Use CDATA when the expression contains characters that must otherwise be escaped in XML.
- `digest.xml` sets optional directory-specific `Relevance`, `DigestRelevanceEvaluationPeriod`, and `ChildEvaluationPeriod`. Supported period values include `PT0S`, `PT1M`, `PT5M`, `PT1H`, `P1D`, `P7D`, and `P30D`.

Configuration file examples and schemas

This `siteConfig.xml` example changes only the Fixlet source directory:

```
<?xml version="1.0" encoding="UTF-8"?>
<SiteConfig>
  <BesFileDir>Fixlets/TeamA</BesFileDir>
</SiteConfig>
```

This `site.xml` example applies global relevance:

```
<?xml version="1.0" encoding="UTF-8"?>
<Site>
    <Relevance>
        <![CDATA[name of operating system starts with "Win"]]>
    </Relevance>
</Site>
```

This `digest.xml` example sets relevance and evaluation periods for its content directory:

```
<?xml version="1.0" encoding="UTF-8"?>
<Digest>
    <Relevance><![CDATA[version of client >= "7.2"]]></Relevance>
  <DigestRelevanceEvaluationPeriod>PT5M</DigestRelevanceEvaluationPeriod>
  <ChildEvaluationPeriod>PT5M</ChildEvaluationPeriod>
</Digest>
```

The following XSD schemas describe the supported structure of each configuration file.

siteConfig.xml schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:element name="SiteConfig">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="BesFileDir" type="xs:string"
 minOccurs="0" maxOccurs="1" />
                <xs:element name="NonClientFilesDir" type="xs:string"
 minOccurs="0" maxOccurs="1" />
                <xs:element name="OtherFilesDir" type="xs:string"
 minOccurs="0" maxOccurs="1" />
            </xs:sequence>
        </xs:complexType>
    </xs:element>
</xs:schema>
```

site.xml schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:element name="Site">
        <xs:complexType>
            <xs:sequence>
```

```
                <xs:element name="Relevance" type="xs:string" minOccurs="0"
 maxOccurs="1" />
            </xs:sequence>
        </xs:complexType>
    </xs:element>
</xs:schema>
```

digest.xml schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:simpleType name="EvaluationPeriodType">
        <xs:restriction base="xs:string">
            <xs:enumeration value="PT0S" />
            <xs:enumeration value="PT1M" />
            <xs:enumeration value="PT5M" />
            <xs:enumeration value="PT15M" />
            <xs:enumeration value="PT30M" />
            <xs:enumeration value="PT1H" />
            <xs:enumeration value="PT2H" />
            <xs:enumeration value="PT4H" />
            <xs:enumeration value="PT6H" />
            <xs:enumeration value="PT8H" />
            <xs:enumeration value="PT12H" />
            <xs:enumeration value="P1D" />
            <xs:enumeration value="P2D" />
            <xs:enumeration value="P3D" />
            <xs:enumeration value="P5D" />
            <xs:enumeration value="P7D" />
            <xs:enumeration value="P15D" />
            <xs:enumeration value="P28D" />
            <xs:enumeration value="P30D" />
        </xs:restriction>
```

```
    </xs:simpleType>
    <xs:element name="Digest">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="Relevance" type="xs:string" minOccurs="0"
 maxOccurs="1" />
                <xs:element name="DigestRelevanceEvaluationPeriod"
 type="EvaluationPeriodType"
                                        minOccurs="0" maxOccurs="1" />
                <xs:element name="ChildEvaluationPeriod"
 type="EvaluationPeriodType"
                                        minOccurs="0" maxOccurs="1" />
            </xs:sequence>
        </xs:complexType>
    </xs:element>
</xs:schema>
```

Content guidelines

Guidelines for relevance nodes in .bes files.

When creating content in your Git source repository, you must comply with the following guideline:

- Always use standard `<Relevance>` nodes in your `.bes` files to define relevance conditions.

Correct example:

```
<Relevance>exists (operating system) whose (it as string as lowercase
 contains "Win" as
lowercase)</Relevance>
```

Note: GroupRelevance Clauses are not supported.

You cannot use `<GroupRelevance>` clauses in your source repository `.bes` files. If your files contain these nodes, repository synchronization will fail during the site validation and export phase.

Incorrect example (Unsupported):

```
<GroupRelevance JoinByIntersection="false">
    <SearchComponentPropertyReference PropertyName="OS"
 Comparison="Contains">
        <SearchText>Win</SearchText>
        <Relevance>exists (operating system) whose (it as string as
 lowercase contains "Win" as lowercase)</Relevance>
    </SearchComponentPropertyReference>
</GroupRelevance>
```

If you use the unsupported `<GroupRelevance>` structure, you will encounter an error in your synchronization logs similar to the following:

```
Failed to sync repository site git@github.com:repo-owner/repo-name.git:
 Site validation and export failed:
C:\Program Files\BigFix Enterprise\BES
 Server\RepositorySiteGather\Sites\repo-name\Source\Fixlets\Analyses\12345
-Some Analysis.bes:
GroupRelevance clauses in .bes files not supported. Use standard
 <Relevance> nodes instead.
```

To fix this issue, extract the core relevance statement from the `<SearchComponentPropertyReference>` block and place it directly inside a standard `<Relevance>` tag, as shown in the correct example above.

Troubleshoot Repository Sites

If you experience issues while configuring or syncing your repository sites, review the following common problems and their recommended troubleshooting steps:

- **SSH fetch fails:** Check network access from the BigFix Server, repository read permissions for the Git account, the BES Server service account's access to `id_rsa`, and the verified host entry in `known_hosts`. Look for `Failed to sync repository site` in the BigFix Server log.
- **A commit does not produce an update:** Check the tracked branch in site details and confirm that its HEAD has advanced. If the scheduled check has not run yet, use **Gather** to request an immediate check.
- **The site initially returns 404:** Allow the initial clone and publication to finish, then check the BigFix Server log for a successful clone and `GatherDB.log` for a successful import. Investigate a 404 that persists after these stages.
- **Expected content is missing:** Check the source directory names and any overrides in `siteConfig.xml`, the numeric filename prefixes and export warnings, and the site import messages in `GatherDB.log`.

Repository Sites are excluded from the AirGap Tool standard external-site map. An isolated BigFix Server must be able to reach an appropriate Git repository or mirror.

Chapter 32. Client certificate

To comply with the modern industry standards, starting from product version 10.0.7, the client certificate of the BigFix Agent will have a validity period of 13 months.

Except for a few specific cases described in the following sections, the transition to the 13-month client certificate and its subsequent management will be automatic and will not require any manual intervention.

Client Certificate Validity Period

Up to product version 10.0.6, BigFix Agents have client certificates with a validity period of 10 years and as long as the certificate is from a Client of our deployment and it is not revoked it will be a working Certificate even if expired.

Starting from product version 10.0.7, BigFix Agents are configured by **default** to use client certificates with a validity period of 13 months (e.g., 398 days), in compliance with modern industry standards.

Starting from product version 11.0.6, this validity period is **configurable** by the administrator using the BESAdmin **-setClientCertificatePolicy** command.

Client CA Certificate Validity Period

The BigFix Root server uses the Client CA certificate to sign and verify the Client certificates: every Client is assigned a certificate to certify its identity, and it is signed by this CA.

The Client CA certificate is generated by the BigFix Administration Tool, usually at installation time and it has a validity of 10 years.

To date, the Client CA certificate continues to be a working Certificate even if expired.

To date, the Client CA certificate could be recreated by rotating the server signing key through the BigFix Administration Tool and it will be refreshed with the Client certificate refresh. Note that this process could take hours on large deployments.

BigFix Agent Automatic Transition to 13-Month Client Certificates

On its first registration attempt, after being upgraded to BigFix version 10.0.7, a BigFix Agent will autonomously switch from a 10-year client certificate to a 13-month certificate, provided that the relay-chain, up to the BigFix Server, to which the Agent is connected, is entirely at 10.0.7 level (or later).

As long as this condition is not met, the BigFix Agent will keep the 10-year certificate, and will continue to use it as needed without this implying any limitation or compromising the BigFix Agent ability to autonomously switch to the 13-month certificate at a later time when it will have the possibility to perform a registration through a relay-chain that is entirely at 10.0.7 level (or later).

When a BigFix Agent 10.0.7 or later switches from the 10-year client certificate to the 13-month one, it will log the following two lines to the standard client log file (YYYYMMDD.log):

```
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 Agent Automatic Maintenance of 13-Month Client Certificates

The client autonomously requests a certificate update when the expiration date approaches. By **default**, this renewal attempt is initiated **45 days** before expiration.

Starting from product version 11.0.6, this renewal notification time is **configurable** by the administrator using the BESAdmin **-setClientCertificatePolicy** command.

When a BigFix Agent, at 10.0.7 level or later, gets an update of its 13-month client certificate, it will log the following two lines to the standard client log file (YYYYMMDD.log):

```
Client Certificate expires in N days, HH:MM:SS, refreshing it now.
Completed Client certificate update.
```

If a BigFix Agent is unable to update its certificate before it expires, if it is connected to an Authenticating Relay, it will be necessary to run the manual procedure, which is described in

How to Recover from an Expired Client Certificate (on page 1205), to allow it to reconnect to the BigFix deployment.

Token-Based Renewal for Offline/Expired Clients

Token-Based Renewal is a new security feature for BigFix , introduced with BigFix version 11.0.6, that allows a client with an expired certificate to securely obtain a new one from the BigFix Server, even when connecting via an Authenticating Relay. Normally, the Relay would reject expired certificates, but the Renewal Token provides a secure, temporary bypass for the certificate renewal only.

The Renewal Token is a unique, cryptographically strong, and time-limited token generated by the BigFix Server. It is valid for a period extending beyond the client certificate validity (e.g., certificate validity plus 30 days) and is stored securely on the client. The token is for one-time use; once a new certificate is obtained, the old token is invalidated, and a new one is generated for the next cycle.

Subject Alternative Name

Starting from BigFix version 11.0.3, the Subject Alternative Name X509v3 standard extension is added to the client certificate of the BigFix Agents. Its value corresponds to the hostname of the computer where the BigFix Agent runs.

On its first registration attempt, after being upgraded to version 11.0.3, a BigFix Agent automatically gets an updated client certificate with the Subject Alternative Name field, provided that the relay-chain, up to the BigFix Server, to which the Agent is connected, is entirely at 11.0.3 level (or later).

Because of the introduction of the Subject Alternative Name field, a BigFix Agent of version 11.0.3 or later automatically gets an updated client certificate in case the hostname of the computer changes.

How to Monitor the Status of Client Certificates

To monitor the status of the client certificate on the BigFix Agents of your deployment, you can activate the **Client Certificate Information** analysis of BES Support. For each BigFix Agent, the analysis will provide the following information:

- **Client Certificate Expiration Date**: The expiration date of the client certificate.
- **Client Certificate Overall Validity**: The overall validity of the client certificate.
- **Client Certificate Expires In**: The remaining validity of the client certificate.

How to Recover from an Expired Client Certificate

If a BigFix Agent has an expired client certificate and it can only reach an Authenticating relay on the network, you can manually run the following command on the BigFix Agent to allow it getting an updated certificate through the Authenticating relay:

BESClient -update-certificate <password> http://<relay>:52311

The command includes a password that the Authenticating relay will verify before forwarding the update request upwards. The Authenticating relay must be at 10.0.7 level or later.

The password on the Authenticating relay might be configured as:

- A single password set through the client setting **_BESRelay_Comm_ClientCertUpdatePassword** to be defined on the relay.
- A newline-delimited list of one-time passwords stored in a file named **ManualUpdateCertificatePasswords** and saved in the storage directory of the relay (value **StoragePath** of HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\BigFix \Enterprise Server\GlobalOptions).

Note: Starting with BigFix version 11.0.6, a new **Token-Based Renewal** mechanism is available to automatically recover clients with expired certificates, especially those connecting through an Authenticating Relay. This process uses a time-limited **Renewal Token**, with a default lifespan of 30 days, generated by the BigFix Server to securely bypass the standard certificate validation for the sole purpose of obtaining a new certificate. This is the primary automated recovery method. Manual recovery using the BESClient -update-certificate command remains available as a fallback if the token renewal fails.

Note: Passwords must have ASCII characters only.

Note: If you want to avoid specifying the clear-text password as part of the manual command, the following alternate syntax will prompt for entering the password without displaying it:

- **Windows**: cmd /c BESClient.exe -update-certificate http://<relay>:52311
- **Linux**: BESClient -update-certificate http://<relay>:52311
- **Mac**: BESAgent -update-certificate http://<relay>:52311

How to Force the Update of the Client Certificate

You have the possibility to force a BigFix Agent at 10.0.7 level or later to update its client certificate at any given moment by targeting it with an action that uses the following new actionscript command:

client certificate refresh

Since BigFix Agents will be able to maintain their client certificates autonomously, under normal conditions BigFix Operators are not expected to have to use this command. However, there might be cases where this command helps, for instance to address specific situations where the 45-day certificate update window is not suitable to guarantee that certificates will not expire, and so a BigFix Operator might want to anticipate the certificate update.

Note: It is important to point out that the command will force the BigFix Agent to request a certificate update regardless of the version of its relay-chain. As a consequence, in case the relay-chain is not entirely at 10.0.7 level or later, the command will make the BigFix Agent get an updated certificate with a validity of 10 years. In this case, the Agent will switch back to applying the logic, described in BigFix Agent Automatic Transition to 13-Month Client Certificates *(on page 1203)*, which allows the transition from a 10-year certificate to a 13-month one.

Configuring the root server with a custom CA

All certificates used in communications among BigFix Platform components are issued by Certificate Authorities (CAs) created by the Root Server.

Starting from BigFix Platform Version 11.0.4, a new feature allows you to configure the Root Server with a custom CA to be used to:

- issue a certificate used by the Root Server to satisfy API requests (e.g. requests coming from the Console or from Web Reports)
- issue certificates used in internal communications involving Clients, Relays and Plugin Portal
- possibly issue certificates used by WebUI and Explorer to communicate with the Root Server.

Using the BigFix Administration tool command named `setcustomca`, you can install a custom CA in the BigFix Platform, and, later on, update the already stored custom CA.

After installing a custom CA in the BigFix Platform, the following happens:

- a new certificate is created with the custom CA as issuer, and the Root Server is configured to use it to satisfy API requests (e.g. requests coming from the Console or from Web Reports)
- at its next registration, any client 11.0.4 will request and obtain from the Root Server a certificate issued by the custom CA
- clients of version 11.0.3 (or earlier) will keep on using their original certificates issued by the CA created by the Root Server
- relays 11.0.4 will still communicate with clients of earlier versions as they will store both certificate chains and will use the most appropriate one depending on the chain used by the connecting client.

To allow Web Reports to communicate with a Root Server configured with a custom CA, you need to copy the custom CA root certificate into the "CustomSSLCertificates" folder of Web Reports.

The custom CA can optionally be used to issue also all the certificates involved in the communications between Root Server and WebUI, and between Root Server and Explorer. To do this, after installing the custom CA, you need to run the BigFix Administration tool commands named `rotatewebuicredentials` and `rotateexplorercredentials` with the `rotateCA` option.

For more details about the `setcustomca`, `rotatewebuicredentials` and `rotateexplorercredentials` commands, see BESAdmin Windows Command Line and BESAdmin Linux Command Line.

Chapter 33. Client Authentication

Client Authentication (introduced in version 9) extends the security model used by BigFix to encompass trusted client reports and private messages.

This feature is not backward-compatible, and clients prior to version 9.0 will not be able to communicate with an authenticating relay or server.



Note: Some of the security options of the Client Authentication feature, can also be defined by setting the **minimumSupportedClient** and **minimumSupportedRelay** services as described in BESAdmin Windows Command Line for Windows systems, or BESAdmin Linux Command Line for Linux systems.

The original security model has two central capabilities:

- **Clients trust content from server.** All commands and questions that clients receive are signed by a key that is verified against a public key installed on the client.
- **Clients can submit private reports to server.** The client can choose to encrypt reports that it sends up to the server, so that no attacker can interpret what is contained in the report. This feature is disabled by default, and is switched on with a setting.

Client Authentication extends the security model to provide the mirror image of these two capabilities:

- **Server can trust reports from clients (non-repudiation).** Clients sign every report that they submit to the server, which is able to verify that the report does not come from an attacker.
- **Server can send private data to clients (mailboxing).** The server can encrypt data that it sends to an individual client, so that no attacker can interpret the data.

Communication using an authenticated relay is a two-way trusted and private communication channel that uses SSL to encrypt all communications. However, communication between a non-authenticating relay and its children is not encrypted unless it is an encrypted report or a mailboxed action or file.

This level of security is useful for many purposes. Your company may have security policies that require authenticating relays on your internet-facing nodes, in your DMZ, or any network connection that you do not totally trust. With authentication, you can prevent clients that have not yet joined your deployment from getting any information about the deployment.

Authenticating relays

BigFix deployments with internet-facing relays that are not configured as authenticating are prone to security threats.

Security threats, in this context, might mean unauthorized access to the relays and any content or actions, and download packages associated with them or to the **Relay Diagnostics** page that might contain sensitive information (for example, software, vulnerability information, and passwords).

You can configure relays as "authenticating" to authenticate the agents. This way, only trusted agents can gather site content or post reports. Use an authenticating relay configuration for an internet-facing relays in the DMZ. A relay configured to authenticate agents only performs TLS communication with child agents or relays that present a TLS certificate issued and signed by the server during a key exchange.

When a relay is configured as *authenticating*, only the BigFix clients in your environment can connect to it and all the communication between them happens through TLS (HTTPS). This configuration also prevents any unauthorized access to the Relay and Server diagnostics page.



Note: If you need to install new clients and you can only reach an authenticating relay, then you must perform a manual key exchange. For details, see Manual key exchange *(on page 1212)*.

How to enable relay authentication

To upgrade the relays to authenticating relays, do the following steps:

1. On the BES Support website, find the **BES Client Settings: Enable Relay authentication** Fixlet.
2. Run the Fixlet and wait for the action to finish.

You can configure relays for authentication by manually updating the `_BESRelay_Comm_Authenticating` configuration setting also. The default value of the setting is `0` which indicates that the relay authentication is disabled; to enable the authentication, set the value to `1`. For more details, see Authentication *(on page 757)*.

These settings must not be set on the BigFix root servers.

By default, every client re-registers with its parent relay once every six hours. Existing clients cannot send reports until they re-register themselves with the relay.

Related information

BigFix - Easily setup an internet facing relay

Handling the key exchange

When an agent tries to register and does not have a key and certificate, it automatically tries to perform a key exchange with its selected relay.

If the relay is a non-authenticating relay, it forwards the request up the relay chain to the server, which signs a certificate for the agent. This certificate can later be used by the agent when connecting to an authenticating relay.

Authenticating relays deny these automatic key exchange operations. The following is a typical scenario:

When you deploy a new BigFix environment or upgrade an existing BigFix environment to Version 11 all agents automatically perform the key exchange with their relays. If the administrator configures the internet facing relay as an authenticating relay, the existing agents already have the certificate and work correctly. No further action is required. When you connect new agents to the authenticating relay they do not work, until the manual key exchange (on page 1212) procedure is run on them.

Manual key exchange

If an agent does not have a certificate and can only reach an authenticating relay on the network, connected through the internet, you can manually run the following command on the agent so it can perform the key exchange with an authenticating relay:

```
BESClient -register [<password>] http://<relay>:52311
```

The client includes the password in its key exchange with the authenticating relay, which verifies it before forwarding the key exchange to its parent.

If you execute the command omitting the password, the password is requested interactively. On Windows sytems, run the command using the cmd /c prefix.

Another way to perform a manual registration to an authenticating relay is by setting a value to the client setting `_BESClient_SecureRegistration`. The value specifies the password needed to perform a manual registration to the authenticating relay. This setting is read only at client startup time. You can specify the relay in the `clientsettings.cfg` configuration file. For more information about this configuration file, see Windows Clients.

You can configure the password on the relay as:

- A single password in the client setting `_BESRelay_Comm_KeyExchangePassword` on the relay.
- A newline-delimited list of one-time passwords stored in a file named `KeyExchangePasswords` in the relay storage directory (value **StoragePath** of `HKEY\SOFTWARE\WOW6432Node\BigFix\Enterprise Server\GlobalOptions`).

Note: You can use only passwords that have ASCII characters and not passwords containing non-ASCII characters.

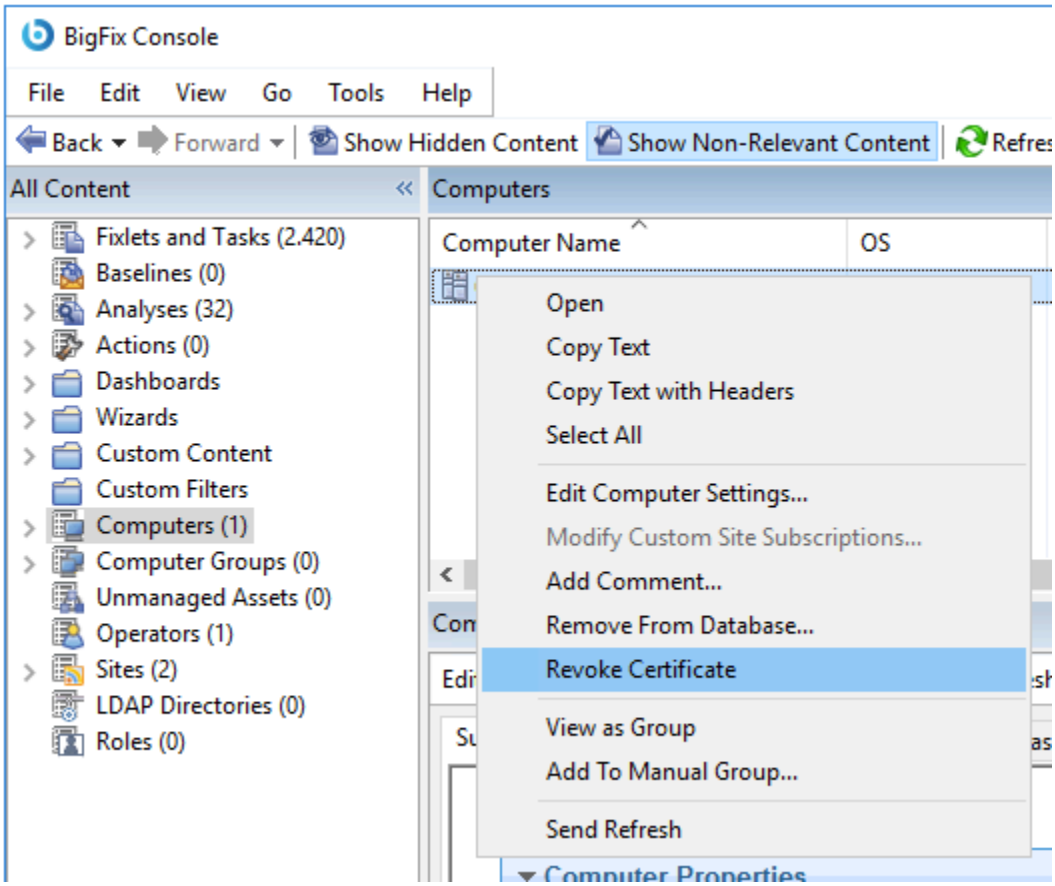
Revoking Client Certificates

After a client authenticates, you can revoke its certificate if you have any reason to doubt its validity.

When you do, that client is no longer authenticated for trusted communication. It is removed from the console and a revocation list is updated and collected by all relays, so that the client's key can no longer be used to communicate with authenticating relays.

To revoke a computer:

1. Right-click a computer in any list of computers.



2. From the pop-up menu, click **Revoke Certificate**.
3. From the confirmation dialog click **OK** if you are sure you want to remove the computer certificate.

This sends revocations down to the relays. After revoked, that client can no longer use its private key to gather content from the authenticating relays. The revoked client disappears from the computer list in the console.

Re-registering a revoked client

The client revoke procedure removes a client from the console and updates a client certificate revocation list.

Clients can automatically get a new certificate if they can connect to any non authenticating relay.

If such a relay is unavailable you must complete the following manual cleanup to register the client again:

1. Stop the client.
2. Delete the KeyStorage client directory and the client computer ID.
3. Complete the manual key exchange procedure.
4. Start the client.

At the end of this procedure the client gets a fresh certificate and a new client computer ID.

Note: If you must revoke the certificate of a SuSe client, connected to an authenticating relay but blocked from the root server, ensure that, before running the manual registration with the password, you copy the following entries in the `besclient.conf` file:

```
Settings\Client\__Relay_Control_Server
Settings\Client\__RelayServer1
```

The manual registration, in fact, deletes automatically these entries in the configuration file and does not create them again, so you must add them manually, after the registration completes, to let the client communicate again with its authenticating relay.

Mailboxing

With Client Mailboxing you can send an encrypted action to any given client, instead of broadcasting it to all clients.

This improves efficiency, since the client doesn't need to qualify every action, and it minimizes network traffic. As a consequence,

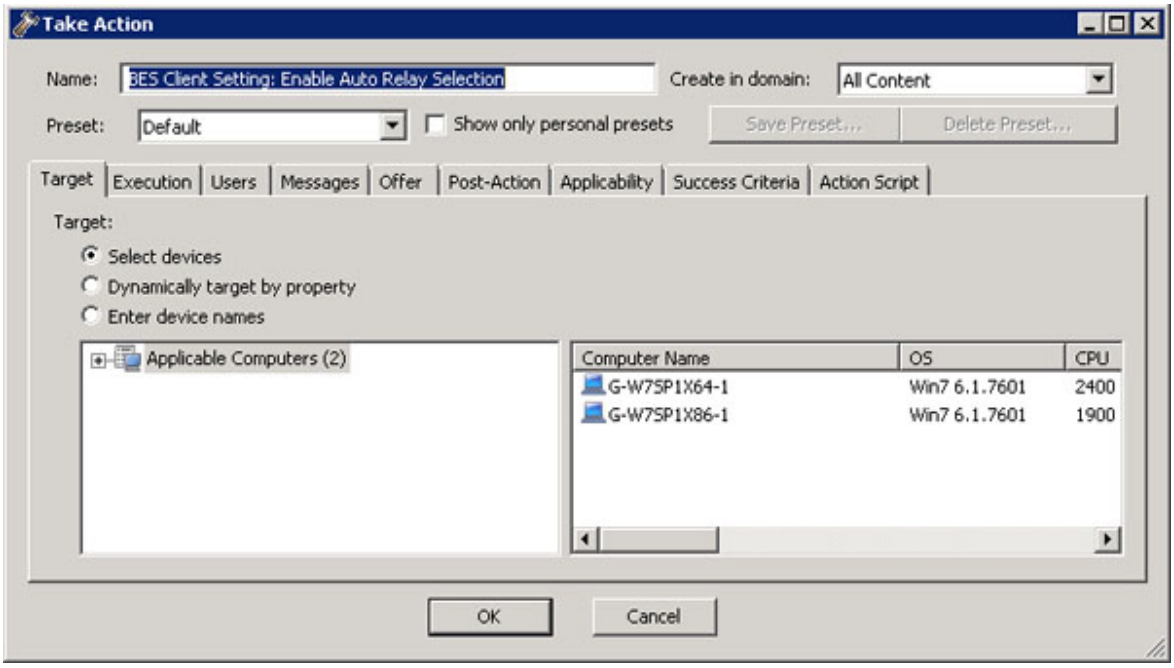
- Clients are only interrupted when they are targeted.
- Clients don't have to process actions that are not relevant to them for reporting, evaluating, gathering, and action processing.

Privacy is assured because the message is encrypted specifically for each recipient; only the targeted client can decrypt it.

A client's mailbox is implemented as a specialized action site, and each client is automatically subscribed to it. The client knows to scan for actions in this site as well as the master site and operator sites.

To send an encrypted action directly to a client mailbox, follow these steps:

1. Open the **Take Action** dialog (available from the Tools menu and various other dialogs).



2. Click the **Target** tab.

3. Click **Select devices** or **Enter device names**. Mail-boxing is only available when you specify a static list of clients. Dynamically targeted computers will not be encrypted and will instead be sent in the open to the master site or a specific operator site. If you select target clients with versions prior to 9.0, the action will also go into the master or operator site.
4. Click **OK**. Actions targeted by computer ID or name will now be encrypted and sent to the client mailbox.

The identifier of the operator who deploys the action is included with the action. Before a client takes the action, it first determines if it is currently administered by that operator. If not, it refuses to run the action.

Chapter 34. Four Eyes Approval Capability

Enabling Four Eyes Approval Capability on the BigFix Server.

Description

The Four Eyes Approval feature is used to prevent console operators from unilaterally taking actions on the endpoints within their control. After this feature is enabled, operators taking console actions will require the approval of a console operator who is also a member of a specified "approvers" Role.

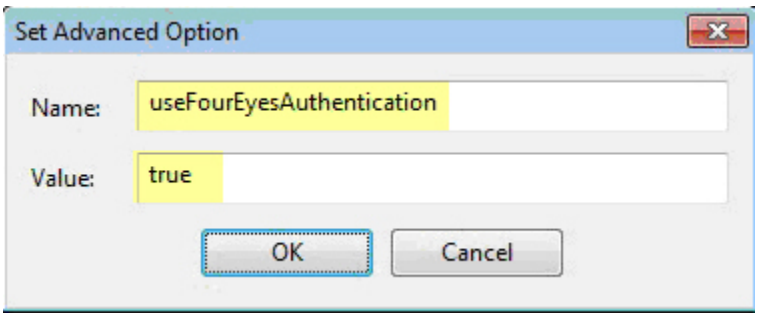
Access

Only System Administrators and Master Operators with access to the Site Administrator Private Key Password can access and configure this feature.

How to use it

Follow the steps below to enable this feature in your deployment:

1. To open the BigFix Administration Tool, click the Start menu on your computer and select **BigFix >BigFix Administration Tool**.
2. When the tool dashboard opens, click the **Advanced Options** tab and Click **Add**.
3. In the **Name** field, type “useFourEyesAuthentication”; in the **Value** field, type “true”; and then click **OK**.

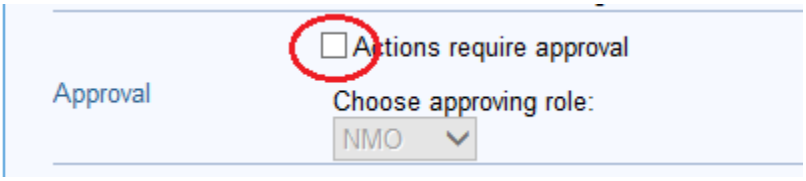


4. Click OK to exit the BigFix Administration Tool.
5. Restart the BES Root Server service on your BigFix server and restart the BigFix Console.
6. From the Console, click the **Tools>Create Role**.

7. Enter a name for the role (e.g., Approvers) and click **OK**. Note: The Role does not require any additional/special permissions.
8. Make one or more BigFix operators members of the role. Note: The BigFix Operators can be Local or AD/LDAP.
9. Click the *Operators* node in the Console navigation tree. This will open an Action window. In the **Details** tab, select the **Actions Require Approval** check box and select the role created in the previous step from the drop-down list.
10. Click **Save Changes**at the top of the window.

NOTE: Do not require approval from your own console actions. As a best practice, ensure that another Master Operator sets this option for you.

LIMITATION: WebUI does not support the Four Eyes Authentication Capability. Operators that want to deploy actions from the WebUI must have Approval disabled in their configuration.



Chapter 35. Representative Computers Selection Wizard

Representative Computers Selection Wizard selects certain BigFix Clients in each subnet as representative computers based on custom criteria. Designated representative computers can be used for targeting tasks that require only a few reliable or high end machines within a subnet or group, such as for selecting BigFix Relays or NMAP scan points.

This wizard is designed to give maximum flexibly to BigFix operators to search for desired computers to run a Fixlet or perform an action.

After successfully running this wizard, the operator can issue actions or policy actions targeting selected representative computers.

Step-by-Step Instructions:

Before You Start: If you have not done so, please complete a one time process in the grey box to add “Representative Computer" property to your BigFix deployment.

This property can be used in the Take Action dialog to easily target machines based on their representative status.

Step 1: Choose subnet(s)

Select interested subnets and hit the “Select Representative Computers” button. Note that the subnet information is gathered by Analysis 840 of BES Support site, and must be activated to use this wizard.

Step 2: Choose Criteria

Choose criteria by which representative computers are selected. This wizard will use information reported by these computers and display the results of all computers in selected subnet, indicating whether each computers meets the criteria or not.

You must subscribe to BigFix Power Management and have power tracking enabled in order to retrieve the “Average Daily On-time” data.

Step 3: Analyze the Results and Deploy the Action

After specifying the criteria, the wizard will automatically select the best/most reliable computers that meet the minimum selection criteria, and will set to remove any previously selected Representatives that no longer the selection criteria.

You can also view all the computers in each subnet by their quality as a Representative computer and if they meet the specified selection criteria, and make manual selection changes if necessary.

Click the “Deploy Action” button to deploy selected actions.

Add it as a Representative – It will set the value of the Client Setting "Representative Computer" to “Selected”.

Remove it as a Representative – It will set the value of the Client Setting "Representative Computer" to “Removed”.

The actions issued by the Wizard only sets the Client Setting on selected endpoints. You have to run your desired actions in Step 4 to complete the process.

Step 4: Run Fixlets or Take Actions

After successfully running the wizard, you can now target the "Representative Computer" property when targeting an action.

The "Representative Computer" property can have three possible values:

Selected - You have selected as a representative computer in the Wizard

Removed - It was “Selected” as a representative computer but it has been removed

Unspecified - The Wizard has never specified any value to this computer.

In the “Take Action” dialog, go to the “Target” tab, check “All computers with the property values selected in the tree below” -> “All Computers” -> "By Retrieved Properties" -> "By Representative Computer" to target selected computers in an action.

Chapter 36. Maintenance and Troubleshooting

If you are subscribed to the Patches for Windows site, you can ensure that you have the latest upgrades and patches to your SQL server database servers.

This means that you must install the client on all your computers, including the server and console computers. In addition, you might want to take advantage of these other tools and procedures:

- If you have the SQL Server installed, you should become familiar with the **MS SQL Server Tools**, which can help you keep the database running smoothly.
- It is standard practice to back up your database on a regular schedule, and the BigFix database is no exception. It is also wise to run the occasional error-check to validate the data.
- If you start to notice any performance degradation, check for fragmentation. BigFix writes out many temporary files, which might create a lot of disk fragmentation, so defragment your drive when necessary. Regular maintenance also involves running the occasional error-check on your disk drives.
- The BigFix **Diagnostics Tool** performs a complete test on the server components and can be run any time you experience problems. For additional information, see Running the BigFix Diagnostics tool.
- Check the **BigFix Management** domain often. There are a number of Fixlets available that can detect problems with any of your BigFix components. This can often prevent problems before they ever affect your network.
- Add relays to improve the overall system performance and pay close attention to them. Healthy relays are important for a healthy deployment.
- Review the **Deployment Health Checks** dashboard in the **BigFix Management** domain for optimizations and failures.
- Set up monitoring activities on the servers to notify you in the event of a software or hardware failure, including:
    - Server powered off or unavailable
    - Disk failure

- Event log errors about server applications
- Server services states
- FillDB buffer directory data back-up situations

Monitoring relays health

BigFix allows you to monitor your client and relay setups to ensure they are working optimally.

Before deploying a large patch, you might want to check the status of your relays to guarantee a smooth rollout.

Here are some suggestions for monitoring your relay deployment:

- Click the **BigFix Management** domain and the **Analyses** node and activate the relay Status analysis. This Analysis contains a number of properties that give you a detailed view of the relay health.
- Click the **Results** tab for the analysis to monitor the Distance to relay property in the relay status analysis to see what is normal in your network. If your topology suddenly changes, or you notice that some of your clients are using extra hops to get to the server, it could indicate the failure of a relay.
- Try to minimize the number of clients reporting directly to the server because it is generally less efficient than using relays. You can see which computers are reporting to which relays by studying this analysis.

Relay and Server diagnostics

To monitor your BigFix environment setup and status and to complete actions on your clients.

You can use the following diagnostics functions to get information about your server and your relay settings and to complete actions on your clients. Starting from V9.5.6, the relay diagnostics page is disabled by default and can be protected by a password when enabled; for more information, see: Relay diagnostics (on page 844).

To access the diagnostics, open a browser and type in the address field:

```
http://<computer_name>:52311/rd
```

or

```
http://<computer_name>:52311/RelayDiagnostics
```

where:

<computer_name>

Is the address of the workstation where the server or the relay that you want to check is installed.

The diagnostics page is divided in the following sections:

Relay or Server Diagnostics

In this section, you can gather information about your environment settings. Click the + sign to expand the different types of setting and see their values.

Note: The entry **Query Settings** refers to BigFix Query processing. For more information about this function, see Getting client information by using BigFix Query *(on page 203)*.

Relay Status Information

In this section, you can view information about the cache used on the relay in the queues dedicated to FillDB and toBigFix Query requests and results.

- **FillDB File Size Limit**
- **FillDB File Counter Limit**
- **Timeout for queries in queue** displays how long the BigFix Query requests can stay in the queue before being removed.
- **Size of queries in queue** shows the size of the cache that is used on the relay to store the BigFix Query requests.
- **Size of results in queue** shows the size of the cache that is used on the relay to store the BigFix Query results.

If you click the **Empty Query Queues** buttons, the queues that store the BigFix Query requests and results in the relay cache are cleaned up.

Console user information

In this section, you can check whether an user is authorized to access BigFix. This section is available only when you access the server diagnostic.

Click **Check User Authorization** and type the user's credentials to verify whether that user is granted access to the BigFix console without the need to actually log in with those credentials.

Site gathering information

In this section, you can collect information related to your environment sites.

- Click **Gather Status Page** to get information about site gathering status.
- Click **Gather All Sites** button to gather the latest version of site contents.
- In **Fixlet Site Requests**, you can collect information about different types of requests related to a site. Select the type of request, the URL of the site in the list provided, whether you want to use CRC or not and then click **Submit**.

Client register

In this section, you can perform requests either for a single computer or for all the computers in your environment.

- Click **Get Computer ID** button to know the computer ID of your relay.
- In **Single Computer Requests**, you can choose different types of requests related to a single computer by selecting one of the requests in the list and by clicking the **Submit** button. Depending on the request type, you might need to fill one or more text fields. The needed fields are automatically enabled.

- In **All Computer Requests**, you can select different types of requests related to all the computers in your environment by selecting one of the requests in the provided list and clicking the **Submit** button. Depending on the request type, you might need to specify the **Action ID**, if enabled.

Download information

In this section, you can collect information about the downloads that are run on the system.

- Click **Download Status Page** to get information about downloads active on your server or relay.
- Click **Download Status Text Page** to get information, in xml language, about downloads active on your server or on your relay.
- In **Download Requests**, you can collect information about a specific action for a specific site by providing the **Action ID** and the **Site URL** in the related fields. Click **Gather Download Request** button to run your request.

Virtualized environments and virtual machines

To run your operating system in multiple virtual machines.

In BigFix you can run your operating system in multiple images to benefit from the ability to share hardware and software resources. This is true especially on HCL z Systems where, within the z/VM environment, Linux images benefit from the reliability, availability, and serviceability of IBM z Systems servers and from internal high-speed communications. z/VM offers an ideal platform for consolidating Linux workloads on a single physical server where you can run hundreds to thousands of Linux images.

In BigFix design, the BESClient agent works in a loop checking the activity to run based on the contents of its directory `<BESClient_installation_path>/__BESData`. These activities, together with a large number of concurrent virtual machines as it is common in z/VM environments, might result in a 100% CPU usage. To avoid this problem and control the CPU

assignment to processes, use the configuration settings described in CPU Usage (on page 619).

Some useful parameters are `_BESClient_Resource_WorkIdle` and `_BESClient_Resource_SleepIdle`, that have default values of 10 and 480 milliseconds respectively, to control the CPU consumption by balancing the amount of work with the amount of idle time; with the default values, this means about 2% of work for each virtual machine. You can change these values if you need to have a lower percentage; the negative side in this instance is that the BigFix client becomes slower when a new activity must be processed. By setting new values, you can take account of the number of virtual machines and avoid the overall CPU being 100% busy.

With other parameters you can set your agents to remain quiet during a part of the day and become active for the remainder of the day; during the quiet period the CPU consumption is almost 0%. The parameters that control this behavior are `_BESClient_Resource_QuietEnable`, `_BESClient_Resource_QuietStartTime`, and `_BESClient_Resource_QuietSeconds`. For example, by setting the following values:

```
_BESClient_Resource_QuietEnable=1
_BESClient_Resource_QuietSeconds=43200
_BESClient_Resource_QuietStartTime=07:00
```

the agent enters quiet mode at 07:00 AM each day, remains in this state for 43,200 seconds, that is for 12 hours, and wakes up at 07:00 PM. During quiet mode, the agent uses almost 0% of CPU time and does not process activities.

Other useful parameters to control the amount of time a client stays in sleep mode, especially suitable when there are battery low power problems or the need to reduce CPU utilization, are `_BESClient_Resource_PowerSaveEnable` and `_BESClient_Resource_PowerSaveTimeout`*X* (X ranging from 0 to 5).

For a full description of all of these parameters and many more, see the configuration settings in the link listed previously.

Related reference

List of settings and detailed descriptions *(on page 404)*

Related information

CPU Usage *(on page 619)*

BES Client Helper Service (Windows only)

The BES Client Helper is intended to be a watcher process for the BES Client and will attempt to restart the service if the BES Client is not running.

The BES Client Helper will also perform a number of troubleshooting steps in the event that the BES Client service does not start at the right time:

1. It attempts a restart.
2. If it fails, it will try to remove the revocation file (create a backup copy) and attempt a new restart.
3. If it fails, it will try to remove the BESData folder and attempt a last restart.

This tool is intended to be installed as a service. It can be installed and uninstalled with the following Fixlets:

- #591: Install BES Client Helper Service
- #592: Uninstall BES Client Helper Service

The service checks the BES Client process once a day by default and no log file is produced. During the installation, you can choose a different check frequency and you can enable the logging activity.

How to change the settings after the installation: Frequency

Set the desired frequency (specified in seconds) into the registry key `_[HKEY_LOCAL_MACHINE\SOFTWARE\BigFix\BESClientHelper\ServiceRunPeriod]_` and restart the service.

How to change the settings after the installation: Logging

Uninstall and install again the helper with different settings using the Fixlet.

Alternatively, you can reinstall the service as follows:

1. Change the registry key `_[HKEY_LOCAL_MACHINE\SOFTWARE\BigFix\BESClientHelper\ServiceInstallationParameters]_` to "-l" (without quotes) to enable logging, or empty to disable
2. Run `_<path of BESClient>\BESClientHelper.exe -remove_`
3. Run `_<path of BESClient>\BESClientHelper.exe -install auto_`

Enabling debug/verbose logging for the BES Root Server and BES Relay services

This procedure describes the steps to enable debug/verbose logging on the BigFix Server or Relays, to log the activity performed by the BigFix Server and by the Relays.

Perform the following steps to enable debug/verbose logging level on BigFix server or relay.

The logging can be enabled by different means; using a Fixlet, creating a BigFix client setting with the BigFix Console or enabling it manually on the machine.

Enabling logging through the Fixlets

Use the following BESSupport Fixlets to enable/disable verbose logging on the BigFix server or relay:

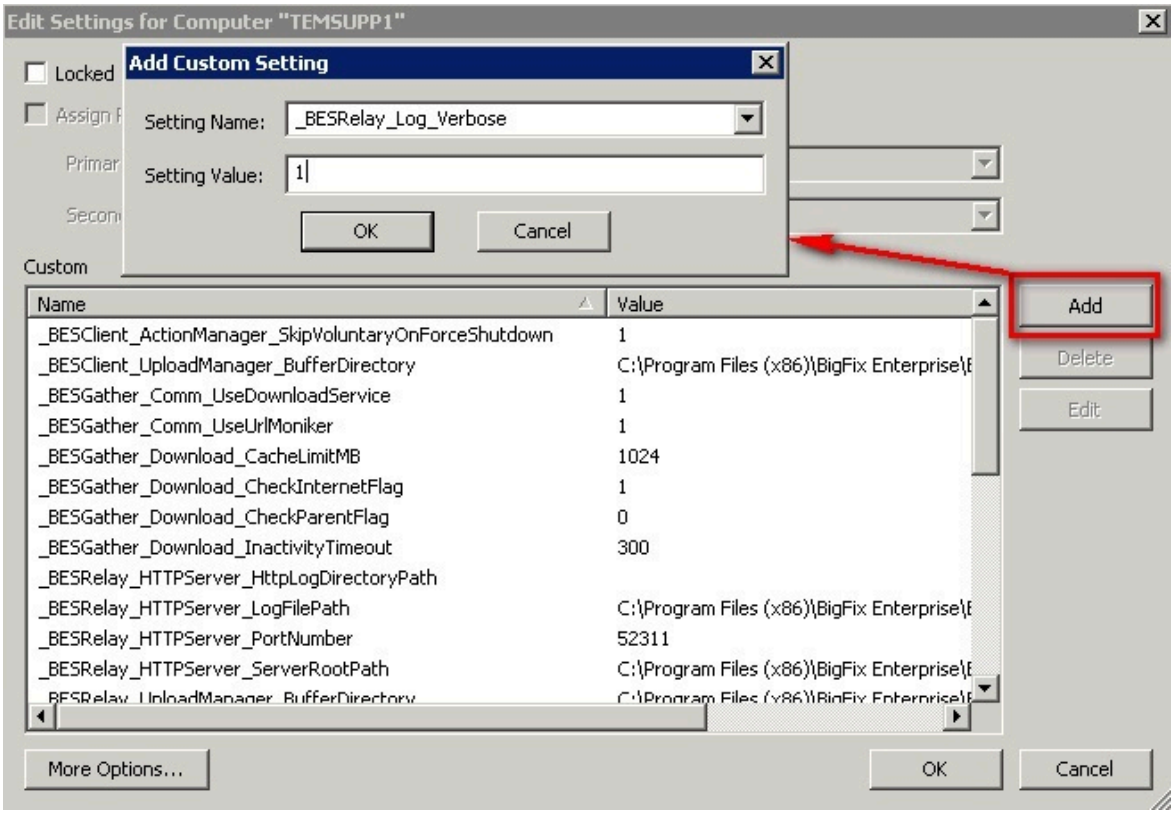
- Fixlet ID: 4595 - Enable Server verbose log
- Fixlet ID: 4596 - WARNING: Server verbose log is enabled

- Fixlet ID: 4776 - Enable Relay verbose log
- Fixlet ID: 4777 - WARNING: Relay verbose log is enabled

Enabling logging through the BigFix Console

1. Log in to the console as a master console operator.
2. Right click the BigFix Server or relay computer in the console.
3. Select Edit Computer Settings....
4. Check in the list to see if the _BESRelay_Log_Verbose setting has already been created. If it has, click the button Edit and change its value to 1 (to enable it).

5. If the setting has not been created, click the button Add to create it. Enter _BESRelay_Log_Verbose for the setting name and 1 for the setting value to enable the verbose logging.

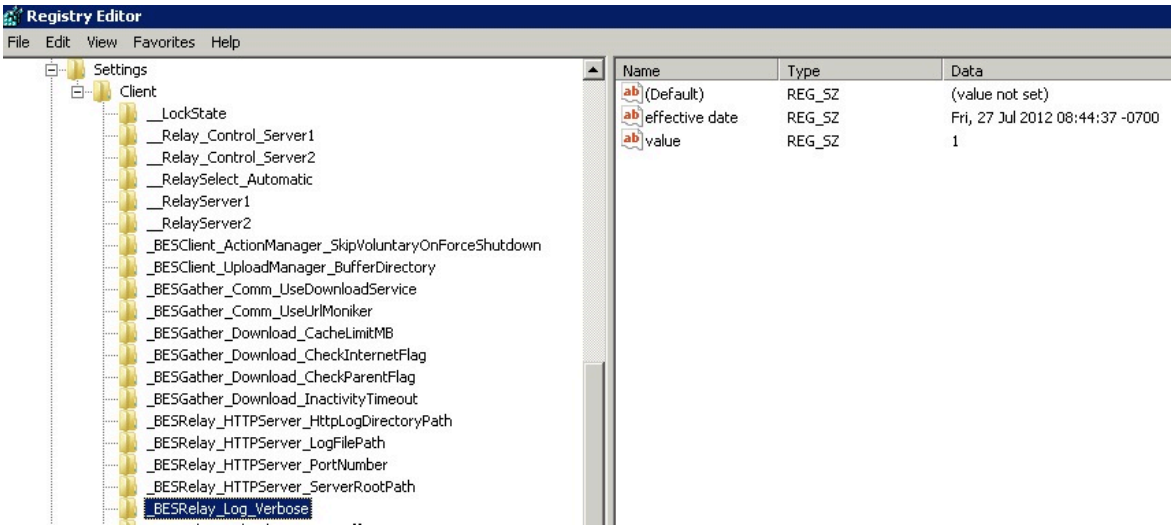


6. Click OK. An action named "Change '_BESRelay_Log_Verbose' Setting" is taken targeted at the BigFix server or relay machine.
7. After the action has completed successfully (and the setting has been applied), the new logging level is effective for the BES Root Server service, while the restart of the BES Relay service is needed for BigFix relay. You can take action on Task # 447: Restart Service in the BES Support site to do this.

Enabling logging manually through the registry (Windows)

1. Log in to the BigFix server or relay machine.
2. Open up the registry editor (regedit).

3. Add the registry key _BESRelay_Log_Verbose in HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\BigFix\EnterpriseClient\Settings\Client.
4. Create a REG_SZ value named "value".
5. Set the value to 1.



6. Restart the BES Relay service if on a relay, while it is not needed on the BigFix Server.

Verbose data is output to <BigFix_Server_Installation_Folder>\BESRelay.log file on the server and <BigFix_Relay_Installation_Folder>\logfile.txt on a relay.

An example of <BigFix_Server_Installation_Folder> on the server is `C:\Program Files (x86)\BigFix Enterprise\BES Server`.

Enabling logging manually through the settings file (Linux)

1. Log in to the BigFix relay machine.
2. Stop BESClient service, to prevent the changes to the configuration file are overwritten, with the command:

```
service besclient stop
```

3. Edit the configuration file /var/opt/BESClient/besclient.config and add or modify the following lines:

```
[Software\BigFix\EnterpriseClient\Settings\Client\_BESRelay_Log_Verbose]
effective date = [Enter Current Date Time In Standard Format]
value = 1
```

The effective current date time must be in a format similar to "Wed, 06 Jun 2012 11:00:00 -0700".

4. Start BESClient service with the command:

```
service besclient start
```

5. Restart the BESRelay service if on a relay:

```
service besrelay stop
service besrelay start
```

Note: The restart of BESRootServer service for the BigFix Server is not required to make the change effective.

Note: Verbose data is output to the `/var/log/BESRelay.log` file on both the server and a relay.

Note: To disable verbose logging, set the BigFix Client setting to 0.

Warning: Leave the verbose logging on just the time needed to troubleshoot the issue you are experiencing, in order to save disk space and processing resources. In large environments, leaving verbose logging on for extended periods of time may heavily lower the BES root service performances causing console timeouts and server activities deadlocks.

Note: A maximum of 10 rotated log files will be maintained in addition to the active log file with the names *logfile.txt*, *logfile.txt_0*, *logfile.txt_1*, ..., *logfile.txt_9*. The default value is 50*1024*1024 (52,428,800) bytes.

Monitoring expensive relevances on Web Reports

Monitor the relevances on your BigFix Web Reports that exceed a custom defined value to evaluate.

This feature is available starting from BigFix Version 10.0.8.

To enable this feature, perform the following steps:

1. Enable the Web Reports log and set it at least to "critical". You can use the Fixlet ID 4591 "Enable Web Reports Server log" from the BES Support site.
2. On the machine where the Web Reports is running, add the "WarnOnLongRelevanceEvaluationMinutes" setting to indicate the number of minutes beyond which a relevance is to be considered expensive (the default is "0", that means the feature is disabled):
   - On Windows:

     ```
     HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\BigFix\Enterprise
      Server\BESReports
     Value: WarnOnLongRelevanceEvaluationMinutes
     Type: REG_SZ
     ```

   - On Linux, modify the `/var/opt/BESWebReportsServer/beswebreports.config` file by adding the key "WarnOnLongRelevanceEvaluationMinutes" in the "Software\BigFix\Enterprise Server\BESReports" section.

   When this setting is defined and its value is bigger than "0", any relevance expression that is evaluated at a time beyond this value is tracked in the Web Reports log file with a message such as the following:

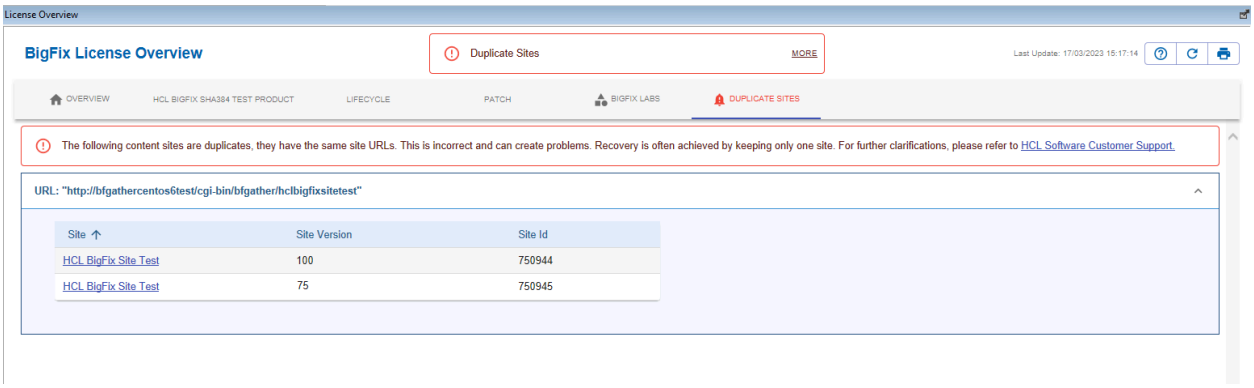
```
Wed, 07 Sep 2022 16:35:56 +0200 -- /relevance (8032) -- The following
 relevance expression
exceeded the "WarnOnLongRelevanceEvaluationMinutes" timeout ( 2
 minutes ) for the evaluation:
'exists ... whose ... ' (1234 ms)
```

Managing duplicate external sites

In BigFix the relationship between Sites and URL should be managed into a one to one correspondance.

If a URL is, by accident, associated to multiple sites, then the License Overview Dashboard warns you about the anomaly, presenting an error message. As of BigFix 11, the License Overview Dashboard handles the inconvenience and provides with an additional help to identify what the duplicated sites are, via a new tab named **Duplicate Sites**.

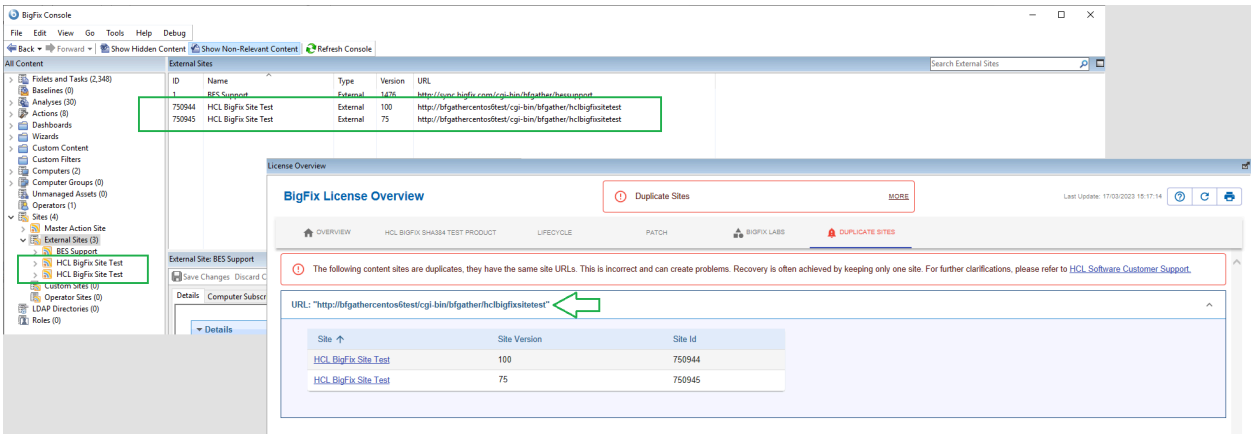
The new tab lists all the sites overlaps, allowing an easier troubleshooting.



A section dedicated to each affected URL groups all the duplicated sites, reporting the site name, site version and site ID.

When a site duplication occurs, the License Overview Dashboards presents two warnings. The first one on the top of the page, provides with an alternative way to land to the **Duplicate Sites** tab by clicking onto the "MORE" button.

The second warning links to the HCL Software Customer Support, where the Error in SiteData constructor : Error : Singular expression refers to non-unique object article describes how to remediate the Singular expression refers to non-unique object error.


BigFix Console
File Edit View Go Tools Help Debug
Back Forward Show Hidden Content Show Non-Relevant Content Refresh Console
All Content
External Sites
Search External Sites
ID Name Type Version URL
750944 HCL BigFix Site Test External 100 http://bfgathercentos6test/cgi-bin/bfgather/hclbigfixsitetest
750945 HCL BigFix Site Test External 75 http://bfgathercentos6test/cgi-bin/bfgather/hclbigfixsitetest
Fixlets and Tasks (2,348)
Baselines (0)
Analyses (30)
Actions (8)
Dashboards
Wizards
Custom Content
Custom Filters
Computers (2)
Computer Groups (0)
Unmanaged Assets (0)
Operators (1)
Sites (4)
Master Action Site
External Sites (3)
BES Support
HCL BigFix Site Test
HCL BigFix Site Test
Custom Sites (0)
Operator Sites (0)
LDAP Directories (0)
Roles (0)
External Site: BES Support
Save Changes Discard C
Details Computer Subscr
Details
License Overview
BigFix License Overview
Duplicate Sites
MORE
Last Update: 17/03/2023 15:17:14
OVERVIEW HCL BIGFIX SHA384 TEST PRODUCT LIFECYCLE PATCH BIGFIX LABS DUPLICATE SITES
The following content sites are duplicates, they have the same site URLs. This is incorrect and can create problems. Recovery is often achieved by keeping only one site. For further clarifications, please refer to HCL Software Customer Support.
URL: "http://bfgathercentos6test/cgi-bin/bfgather/hclbigfixsitetest"
Site Site Version Site Id
HCL BigFix Site Test 100 750944
HCL BigFix Site Test 75 750945

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.